

Филиал Государственного бюджетного профессионального
образовательного учреждения Иркутской области
«Иркутский колледж автомобильного транспорта
и дорожного строительства» в поселке Улькан

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ
РАБОТ ПО ДИСЦИПЛИНЕ
ОУД.14 Информатика**


по программе подготовки квалифицированных рабочих и служащих
15.01.31 Мастер по контрольно – измерительным приборам и автоматики

Улькан, 2021

Методические указания по выполнению практических работ по дисциплине разработаны на основе ФГОС СПО по профессии 15.01.31 Мастер контрольно-измерительных приборов и автоматики и рабочей программы дисциплины ОУД.14 «Информатика»

Организация-разработчик: Филиал Государственного бюджетного профессионального образовательного учреждения Иркутской области «Иркутский колледж автомобильного транспорта и дорожного строительства» в поселке Улькан

Составитель:
Оборина Н.С., преподаватель ВКК

Рассмотрены и одобрены на заседании методической комиссии
«Общеобразовательный, профессиональный цикл»
Методист  Баженова И.В.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данные Краткий теоретический материал по выполнению практических работ являются частью учебно-методического комплекса по дисциплине «Информатика» по профессии среднего профессионального образования.

Целью практических работ студентов является приобретение навыков работы на персональном компьютере в среде Windows и основных офисных программах MS Office – текстового редактора MS Word; табличного редактора MS Excel; системы управления базами данных MS Access; для создания презентаций MS Power Point. Работа на языке программирования Pascal, установка операционной системы Windows и принтера.

Выполнение практических работ направлено на закрепление полученных в ходе изучения тем знаний и реализацию выполнения требований. Практические занятия повышают качество знаний, их глубину, конкретность, оперативность, значительно усиливают интерес к изучению дисциплины, помогают студентам полнее осознать практическую значимость естественных наук.

Во всех практических работах учитывается полнота и качество выполнения практических заданий.

Практические работы выполняются согласно методическим рекомендациям на ПК и оформляются в тетради.

Практические занятия значительно повышают качественный уровень знаний, повышают мотивацию к изучению дисциплины, дают возможность учащимся более полно осознать необходимость практической значимости «Информатики».

Практические работы по дисциплине «Информатика» выполняются после изучения теоретического материала по темам: «Информационная деятельность человека», «Информация и информационные процессы», «Средства информационных и коммуникационных технологий», «Технологии создания и преобразования информационных объектов», «Телекоммуникационные технологии».

Выполнение практических работ согласно содержания учебной дисциплины, обеспечивает достижение студентами следующих результатов:

личностных:

чувство гордости и уважения к истории развития и достижениям отечественной информатики в мировой индустрии информационных технологий;

осознание своего места в информационном обществе;

готовность и способность к самостоятельной и ответственной творческой деятельности с использованием информационно-коммуникационных технологий;

умение использовать достижения современной информатики для повышения собственного интеллектуального развития в выбранной профессиональной деятельности, самостоятельно формировать новые для себя знания в профессиональной области, используя для этого доступные источники информации;

умение выстраивать конструктивные взаимоотношения в командной работе по решению общих задач, в том числе с использованием современных средств сетевых коммуникаций;

умение управлять своей познавательной деятельностью, проводить самооценку уровня собственного интеллектуального развития, в том числе с использованием современных электронных образовательных ресурсов;

умение выбирать грамотное поведение при использовании разнообразных средств информационно-коммуникационных технологий как в профессиональной деятельности, так и в быту;

готовность к продолжению образования и повышению квалификации в избранной профессиональной деятельности на основе развития личных информационно-коммуникационных компетенций;

метапредметных:

умение определять цели, составлять планы деятельности и определять средства, необходимые для их реализации;

использование различных видов познавательной деятельности для решения информационных задач, применение основных методов познания (наблюдения, описания,

измерения, эксперимента) для организации учебно-исследовательской и проектной деятельности с использованием информационно коммуникационных технологий;

использование различных информационных объектов, с которыми возникает необходимость сталкиваться в профессиональной сфере в изучении явлений и процессов;

использование различных источников информации, в том числе электронных библиотек, умение критически оценивать и интерпретировать информацию, получаемую из различных источников, в том числе из сети Интернет;

умение анализировать и представлять информацию, данную в электронных форматах на компьютере в различных видах;

умение использовать средства информационно-коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности;

умение публично представлять результаты собственного исследования, вести дискуссии, доступно и гармонично сочетая содержание и формы представляемой информации средствами информационных и коммуникационных технологий;

предметных:

сформированность представлений о роли информации и информационных процессов в окружающем мире;

владение навыками алгоритмического мышления и понимание методов формального описания алгоритмов, владение знанием основных алгоритмических конструкций, умение анализировать алгоритмы;

использование готовых прикладных компьютерных программ по профилю подготовки;

владение способами представления, хранения и обработки данных на компьютере;

владение компьютерными средствами представления и анализа данных в электронных таблицах;

сформированность представлений о базах данных и простейших средствах управления ими;

сформированность представлений о компьютерно-математических моделях и необходимости анализа соответствия модели и моделируемого объекта (процесса);

владение типовыми приемами написания программы на алгоритмическом языке для решения стандартной задачи с использованием основных конструкций языка программирования;

сформированность базовых навыков и умений по соблюдению требований техники безопасности, гигиены и ресурсосбережения при работе со средствами информатизации;

понимание основ правовых аспектов использования компьютерных программ и прав доступа к глобальным информационным сервисам;

применение на практике средств защиты информации от вредоносных программ, правил личной безопасности и этики работы с информацией и средствами коммуникаций в Интернете.

Практические работы, выполняемые студентами, позволяют им приобрести опыт познавательной и практической деятельности, а также способствуют освоению общих компетенций по Федеральному Государственному образовательному стандарту СПО:

- ОК 4. Осуществлять поиск и использования информации для эффективного выполнения профессиональных задач профессионального и личностного развития.
- ОК5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

Краткий теоретический материал включают в себя:

1. Перечень тем и заданий для практических работ.
2. Краткий теоретический материал и пояснения по выполнению данных работ.
3. Критерии оценки практических работ.
4. Литературу, необходимую для выполнения данных работ.

Перечень практических работ

Практическая работа	Объем часов
№ 1-3 Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеoinформации	3
№ 4-6 Скорость передачи информации.	3
№ 7-9 Использование описания (информационной модели) в процессе общения, практической деятельности, исследования.	3
№ 10-12 Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности	3
№ 13-15 Системы счисления.	3
№ 16-18 Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности	3
№ 19-21 Индуктивное определение объектов. Вычислимые функции, полнота формализации понятия вычислимости, универсальная вычислимая функция; диагональное доказательство не существования.	3
№ 22-24 Сложность вычисления; проблема перебора. Задание вычислимой функции системой уравнений.	3
№ 25-27 Выигрышные стратегии. Сложность описания. Кодирование с исправлением ошибок. Сортировка.	3
№ 28-30 Элементы теории алгоритмов. Вычислимость.	3
№ 31-33 Эквивалентность алгоритмических моделей.	
№ 34-36 Построение алгоритмов и практические вычисления.	3
№ 37-39 Основные конструкции языка программирования.	3
№ 40-42 Система программирования.	3
№ 43-45 Основные этапы разработки программ.	
№ 46-48 Разбиение задачи на подзадачи.	3
№ 49-51 Информационные ресурсы и каналы государства, общества, организации, их структура. Работа на портале государственных услуг.	3
№ 52-54 Роль информации в современном обществе и его структурах: экономической, социальной, культурной, образовательной.	3
№ 55-57 Образовательные информационные ресурсы.	3
№ 58-60 Экономика информационной сферы. Стоимостные характеристики информационной деятельности.	
№ 61-63 Информационная этика и право, информационная безопасность.	3
№ 64-66 Правовые нормы, относящиеся к информации, правонарушения в информационной сфере, меры их предотвращения.	3
№ 67-69 Операционные системы.	3
№ 70-72 Понятие о системном администрировании.	3
№ 73-75 Безопасность, гигиена, эргономика, ресурсосбережение, технологические требования при эксплуатации компьютерного рабочего места.	3
№ 76-78 Типичные неисправности и трудности в использовании ИКТ.	3
Комплектация компьютерного рабочего места в соответствии с целями его использования	
№ 79-81 Оценка числовых параметров информационных объектов и процессов, характерных для выбранной области деятельности.	3
№ 82-84 Профилактика оборудования.	3
№ 85-87 Использование готовых и создание собственных шаблонов	3
№ 88-90 Использование систем проверки орфографии и грамматики. Тезаурусы.	3
№ 91-93 Использование систем двуязычного перевода и электронных словарей.	3
№ 94-96 Коллективная работа над текстом, в том числе в локальной компьютерной сети. Использование цифрового оборудования	3
№ 97-99 Использование специализированных средств редактирования математических текстов и графического представления математических объектов.	3

№ 100-102 Использование систем распознавания текстов.	3
№ 103-105 Ввод и обработка графических объектов.	3
№ 106-108 Ввод и обработка звуковых объектов.	3
№ 109-111 Использование инструментов специального программного обеспечения и цифрового оборудования	3
№ 112-114 Создание графических комплексных объектов для различных предметных областей: преобразования, эффекты, конструирование. Создание и преобразование звуковых и аудиовизуальных объектов.	
№ 115-117 Создание презентаций, выполнение учебных творческих и конструкторских работ.	3
№118-120 Опытные работы в области картографии, использование геоинформационных систем в исследовании экологических и климатических процессов, городского и сельского хозяйства.	3
№ 121-125 Использование динамических (электронных) таблиц для выполнения учебных заданий из различных предметных областей: обработка результатов естественнонаучного и математического эксперимента, экономических и экологических наблюдений, социальных опросов, учета индивидуальных показателей учебной деятельности.	3
№ 126-130 Примеры простейших задач бухгалтерского учета, планирования и учета средств.	3
№ 131-135 Использование инструментов решения статистических и расчетно-графических задач.	3
№ 136-140 Примеры простейших задач бухгалтерского учета, планирования и учета средств Обработка числовой информации на примерах задач по учету и планированию.	5
№ 141-145 Компьютерные архивы информации: электронные каталоги, базы данных	
№ 146-150 Организация баз данных	5
№ 151-155 Примеры баз данных: юридические, библиотечные, здравоохранения, налоговые, социальные, кадровые	5
№ 156-160 Использование инструментов системы управления базами данных для формирования примера базы данных учащихся в школе.	5
№161-165 Использование инструментов поисковых систем (формирование запросов) для работы с образовательными порталами и электронными каталогами библиотек, музеев, книгоиздания, СМИ в рамках учебных заданий из различных предметных областей. Правила цитирования источников информации.	5
№ 166-170 Технологии и средства защиты информации в глобальной и локальной компьютерных сетях от разрушения, несанкционированного доступа.	
№ 171-175 Правила подписки на антивирусные программы и их настройка на автоматическую проверку сообщений.	5
№ 176-191 Инструменты создания информационных объектов для Интернета.	5
№ 192-206 Методы и средства создания и сопровождения сайта.	5
№ 207-211 Системы автоматического тестирования и контроля знаний	5
№ 212-216 Использование тестирующих систем в учебной деятельности	5
№ 217-228 Инструменты создания простых тестов и учета результатов тестирования.	12

Критерии оценивания практических работ:

Оценка «5» - работа выполнена в заданное время, самостоятельно, с соблюдением технологической последовательности, качественно и творчески;

Оценка «4» - работа выполнена в заданное время, самостоятельно, с соблюдением

технологической последовательности, при выполнении отдельных операций допущены небольшие отклонения; общий вид аккуратный;

Оценка «3» - работа выполнена в заданное время, самостоятельно, с нарушением технологической последовательности, отдельные операции выполнены с отклонением от образца (если не было на то установки); оформлено небрежно или не закончено в срок;

Оценка «2» - обучающийся самостоятельно не справился с работой, технологическая последовательность нарушена, при выполнении операций допущены большие отклонения, оформлено небрежно и имеет незавершенный вид.

Практическая работа № 1-3

Дискретное (цифровое) представление текстовой, графической, звуковой информации и видеоинформации

Цели занятия: закрепить умение кодирования и декодирования информации, шифрования и дешифрования данных; изучить способы представления текстовой, графической, звуковой информации и видеоинформации.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Теоретические сведения

Шифр Цезаря, также известный как шифр сдвига, код Цезаря или сдвиг Цезаря — один из самых простых и наиболее широко известных методов шифрования.

Шифр Цезаря — это вид шифра подстановки, в котором каждый символ в открытом тексте заменяется символом, находящимся на некотором постоянном числе позиций левее или правее него в алфавите. Например, в шифре со сдвигом вправо на 3, А была бы заменена на Г, Б станет Д, и так далее.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки со своими генералами.

Шаг шифрования, выполняемый шифром Цезаря, часто включается как часть более сложных схем, таких как шифр Виженера, и всё ещё имеет современное приложение в системе ROT13. Как и все моноалфавитные шифры, шифр Цезаря легко взламывается и не имеет практически никакого применения на практике.

Шифр Виженера — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова. Этот метод является простой формой многоалфавитной замены. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа.

Кодирование информации. Кодирование информации – это процесс формирования определенного представления информации.

В более узком смысле под термином «кодирование» часто понимают переход от одной формы представления информации к другой, более удобной для хранения, передачи или обработки.

Содержание работы:

Задание 1. Шифрование данных

1. **Шифры замены.** Каждая буква алфавита может быть заменена любым числом из соответствующего столбика кодировочной таблицы.

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р
21	37	14	22	01	24	62	73	46	23	12	08	27	53	35	04
40	26	63	47	31	83	88	30	02	91	72	32	77	68	60	44
10	03	71	82	15	70	11	55	90	69	38	61	54	09	84	45

С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
20	13	59	25	75	43	19	29	06	65	74	48	36	28	16
52	39	07	49	33	85	58	80	50	34	17	56	78	64	41
89	67	93	76	18	51	87	66	81	92	42	79	86	05	57

Какие сообщения закодированы с помощью этой таблицы?

16	55	54	10	69	09	61	89	29	90	49	44	10	08	02	73	21	32	83	54	74
41	55	77	10	23	68	08	20	66	90	76	44	21	61	90	55	21	61	83	54	42
57	30	27	10	91	68	32	20	80	02	49	45	40	32	46	55	40	08	83	27	17

2. **Шифр Цезаря.** Этот шифр реализует следующие преобразования текста: каждая буква исходного текста заменяется следующей после нее буквой в алфавите, который считается написанным по кругу.

Задание. Используя шифр Цезаря, зашифровать следующую фразу: **Делу время – потехе час**

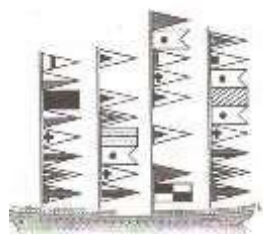
3. При помощи шифра Виженера зашифровать текст «Полиалфавитная замена». Ключ: «Шифр»

Задание №2.

С помощью флажковой азбуки расшифруйте следующее сообщение

Дана кодовая таблица флажковой азбуки

А	Б	В	Г	Д	Е	Ж
З	И	Й	К	Л	М	Н
О	П	Р	С	Т	У	Ф
Х	Ц	Ч	Ш	Щ	Ъ	Ы
Ь	Э	Ю	Я			



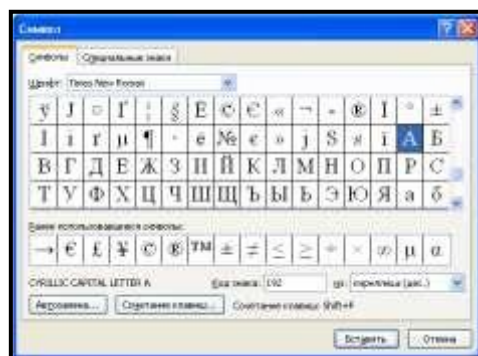
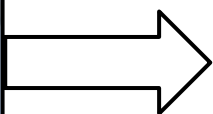
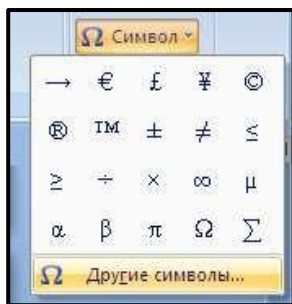
Задание №3

Расшифруйте следующие слова и определите правило кодирования:

ЕРАВШН, УМЫЗАК, АШНРРИ, РКДЕТИ.

Задание №4. Используя таблицу символов, записать

последовательность десятичных числовых кодов в кодировке Windows для своих ФИО. Таблица символов отображается в редакторе MSWord с помощью команды: вкладка **Вставка**→**Символ**→**Другие символы**



В поле **Шрифт** выбираете TimesNewRoman, в поле **из** выбираете кириллица. Например, для буквы «А» (русской заглавной) код знака– 192.

Пример:

И	В	А	Н	О	В	А	Р	Т	Е	М
200	194	192	205	206	194	192	208	210	197	204

П	Е	Т	Р	О	В	И	Ч
207	197	210	208	206	194	200	215

Задание №5. Используя стандартную программу **БЛОКНОТ**, определить, какая фраза в кодировке Windows задана последовательностью числовых кодов и продолжить код. Запустить **БЛОКНОТ**. С помощью дополнительной цифровой клавиатуры при нажатой клавише **ALT** ввести код, отпустить клавишу **ALT**. В документе появиться соответствующий символ.

0255	0243	0247	0243	0241	0252	0226	0195	0207	0202	0237	0224
------	------	------	------	------	------	------	------	------	------	------	------

0239	0229	0240	0226	0238	0236	0234	0243	0240	0241	0229
------	------	------	------	------	------	------	------	------	------	------

Задание №6. Ответить на вопросы:

1. Какие виды информации Вы знаете?
2. Приведите примеры аналогового представления графической информации.
3. Что такое пиксель?
4. Перечислите единицы измерения информации.
5. Каким кодом кодируется красный цвет? Зеленый? Красный? Черный?
6. Чем отличается непрерывный сигнал от дискретного?
7. Какие звуковые форматы вы знаете?
8. Какие этапы кодирования видеоинформации вам известны?
9. Какие форматы видео файлов вы знаете?

Задание №7. Сделать вывод о проделанной практической работе.

Практическая работа № 4-6
Скорость передачи информации

Цели занятия: закрепить знания о скорости передачи информации

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Задание. Заполните пропуски в таблице, находя значения скорости, времени, объем передаваемого файла. Указать единицы измерения.

I – объём передаваемого файла

V – скорость

T – время

	v	t	I
Формула	$v = \frac{I}{t}$	$t = \frac{I}{v}$	$I = v * t$
1.	10 бит/с	12с	
2.		2с	180байт
3.	20 Кбайт		320Кбит
4.	10 бит/с	3мин	
5.		2,5мин	300байт
6.	2048 Кбит/с		4Кбайт

Задание. Заполните пропуски в таблице, находя значения скорости, времени, объем передаваемого файла. Указать единицы измерения.

I – объём передаваемого файла

V – скорость

T – время

	v	t	I
Формула	$v = \frac{I}{t}$	$t = \frac{I}{v}$	$I = v * t$
1.	10 бит/с	12с	
2.		2с	180байт
3.	20 Кбайт		320Кбит
4.	10 бит/с	3мин	
5.		2,5мин	300байт
6.	2048 Кбит/с		4Кбайт

Решение задач.

Задача 1. Скорость передачи данных через ADSL-соединение равна 256000 бит/с. Передача файла через данное соединение заняла 16 секунд. Определите размер переданного файла в Кбайт.

Дано:

$$v = 256000 \text{ бит/с}$$

$$t = 16 \text{ с}$$

Найти: I (Кбайт) - ?

Решение:

$$v = 256000 \text{ бит/с} = 256 \cdot 1000 \text{ бит/с} = 256 \cdot 8 \cdot 125 \text{ бит/с} = 2^8 \cdot 2^3 \cdot 125 \text{ бит/с} = 2^{11} \cdot 125 \text{ бит/с}$$

$$t = 16 \text{ с} = 2^4 \text{ с}$$

$$I = v \cdot t; I = 2^{11} \cdot 125 \cdot 2^4 \text{ бит} = 2^{15} \cdot 125 \text{ бит} = 2^{15} \cdot 125 / (2^3 \cdot 2^{10}) \text{ Кбайт} = 2^2 \cdot 125 \text{ Кбайт} = 500 \text{ Кбайт}$$

Ответ: 500 Кбайт.

Задача 2. Файл размером 16 Кбайт передается через некоторое соединение со скоростью 512 байт в секунду. Определите размер файла (в Кбайт), который можно передать за то же время через другое соединение со скоростью 512 бит в секунду. В ответе укажите одно число — размер файла в байтах. Единицы измерения писать не нужно.

Дано:

$$I_1 = 16 \text{ Кбайт}$$

$$v_1 = 512 \text{ байт/с}$$

$$v_2 = 512 \text{ бит/с}$$

$$t_1 = t_2$$

Найти: I₂(байт) - ?

Решение:

$$I_1 = 16 \text{ Кбайт} = 2^4 \cdot 2^{10} \cdot 2^3 = 2^{17} \text{ бит}$$

$$v_1 = 512 \text{ байт/с} = 2^9 \cdot 2^3 \text{ бит/с} = 2^{12} \text{ бит/с}$$

$$v_2 = 512 \text{ бит/с} = 2^9 \text{ бит/с}$$

$$I = v \cdot t \Rightarrow t = I / v, t_1 = 2^{17} / 2^{12} \text{ с} = 2^5 \text{ с}, t_2 = t_1 = 2^5 \text{ с},$$

$$I_2 = 2^9 \cdot 2^5 = 2^{14} \text{ бит} = 2^{14} / 2^3 \text{ байт} = 2^{11} \text{ байт} = 2^{10} \cdot 2 \text{ байт} = 1024 \cdot 2 \text{ байт} = 2048 \text{ байт}$$

Ответ: 2048

Материалы для самостоятельной работы учащихся.

Материалы для самостоятельной работы учащихся представлены в виде задач в трех вариантах.

1 вариант.

1. Файл размером 64 Кбайт передаётся через некоторое соединение со скоростью 1024 бит в секунду. Определите размер файла (в Кбайт), который можно передать за то же время через другое соединение со скоростью 256 бит в секунду. В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно. Ответ: 16.
2. Файл размером 1000 Кбайт передаётся через некоторое соединение в течение 1 минуты. Определите размер файла (в Кбайт), который можно передать через это соединение за 36 секунд. В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно. Ответ: 600.
3. Модем передал 15Гбайт информации за 32 минуты. С какой скоростью модем передаст информацию. Значение скорости записать в Мбит/с. Ответ: 64
4. Файл размером 2 Мбайта передаётся через некоторое соединение за 16 секунд. Определите время в секундах, за которое можно передать через то же самое соединение файл размером 4096 Кбайт. В ответе укажите только число секунд. Единицы измерения писать не нужно. Ответ: 32.

2 вариант.

1. Файл размером 100 Кбайт передаётся через некоторое соединение со скоростью 1536 бит в секунду. Определите размер файла (в Кбайт), который можно передать за то же время через другое соединение со скоростью 768 бит в секунду. В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно. Ответ: 50.
2. Файл размером 4000 Кбайт передаётся через некоторое соединение в течение 1 минуты. Определите размер файла (в Кбайт), который можно передать через это соединение за 45 секунд. В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно. Ответ: 3000
3. Какова средняя скорость передачи данных (в бит/с), если файл размером 250 Кбайт был передан за 16 с. Ответ: 128000
4. Файл размером 8 Мбайт передаётся через некоторое соединение за 128 секунд. Определите время в секундах, за которое можно передать через то же самое соединение файл размером 4096 Кбайт. В ответе укажите только число секунд. Единицы измерения писать не нужно. Ответ: 64.

3 вариант.

1. Файл размером 160 Кбайт передаётся через некоторое соединение со скоростью 2048 бит в секунду. Определите размер файла (в Кбайт), который можно передать за то же время через другое соединение со скоростью 768 бит в секунду. В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно. Ответ: 60.
2. Файл размером 5000 Кбайт передаётся через некоторое соединение в течение 2 минут. Определите размер файла (в Кбайт), который можно передать через это соединение за 48 секунд. В ответе укажите одно число — размер файла в Кбайт. Единицы измерения писать не нужно. Ответ: 1310.
3. Какова скорость передачи информации через соединение (в бит/с), если для передачи файла, размером 5 Кбайт, на это было затрачено 40 с. Ответ: 1024
4. Файл размером 3 Мбайта передаётся через некоторое соединение за 2 минуты. Определите время (в секундах), за которое можно передать через это же соединение 4 Мбайта. В ответе укажите одно число — количество секунд. Единицы измерения писать не нужно. Ответ: 160.

Практическая работа № 7-9

Использование описания (информационной модели) в процессе общения, практической деятельности, исследования

Цели занятия: научиться пользоваться образовательными информационными ресурсами, искать нужную информацию с их помощью;

Оборудование, программное обеспечение: ПК, ОС Windows.

Методические рекомендации

Понятие «информационного ресурса общества» (ИРО) является одним из ключевых понятий социальной информатики. Широкое использование этого понятия началось после выхода в 1984 году книги Громова Г.Р. «Национальные информационные ресурсы: проблемы промышленной эксплуатации».

«Информационный ресурс – это знания, представленные в проектной форме», – такое краткое и недостаточно строгое определение было предложено профессором Ю.М. Каныгиным.

Таким образом, информационные ресурсы – это знания, подготовленные для целесообразного социального использования.

Понятие ИРО, накопленных в обществе знаний, может быть рассмотрено в узком и широком смысле слова.

ИРО в узком смысле слова – это знания, уже готовые для целесообразного социального использования, то есть отчужденные от носителей и материализованные знания.

ИРО в широком смысле слова включают в себя все отчужденные от носителей и включенные в информационный обмен знания, существующие как в устной, так и в материализованной форме.

Понятие *ресурс* определяется в Словаре русского языка С.И. Ожегова как запас, источник чего-нибудь.

Что же касается *информационных ресурсов*, то это понятие является сравнительно новым. Оно еще только начинает входить в жизнь современного общества, хотя в последние годы становится все более употребительным не только в научной литературе, но и в общественно-политической деятельности. Причиной этого, безусловно, является глобальная информатизация общества, в котором все больше начинает осознаваться особо важная роль информации и научных знаний.

Для *классификации информационных ресурсов* могут быть использованы следующие их наиболее важные параметры:

- тематика хранящейся в них информации;
- форма собственности – государственная (федеральная, субъекта федерации, муниципальная), общественных организаций, акционерная, частная;
- доступность информации – открытая, закрытая, конфиденциальная;
- принадлежность к определенной информационной системе – библиотечной, – архивной, научно-технической;
- источник информации – официальная информация, публикации в СМИ, статистическая отчетность, результаты социологических исследований;
- назначение и характер использования информации – массовое региональное, ведомственное;
- форма представления информации – текстовая, цифровая, графическая, мультимедийная;
- вид носителя информации – бумажный, электронный.

Под *образовательными информационными ресурсами* мы будем понимать текстовую, графическую и мультимедийную информацию, а также исполняемые программы (дистрибутивы), то есть электронные ресурсы, созданные специально для использования в процессе обучения на определенной ступени образования и для определенной предметной области.

При работе с образовательными ресурсами появляются такие понятия, как *субъект* и *объект* этих ресурсов. Классификацию субъектов информационной деятельности произведем следующим образом:

- субъект, создающий объекты (все пользователи образовательной системы – преподаватель, студент);

- субъект, использующий объекты (все пользователи образовательной системы);
- субъект, администрирующий объекты, то есть обеспечивающий среду работы с объектами других субъектов (администраторы сети);
- субъект, контролирующий использование объектов субъектами (инженеры).

К образовательным электронным ресурсам можно отнести:

- учебные материалы (электронные учебники, учебные пособия, рефераты, дипломы),
- учебно-методические материалы (электронные методики, учебные программы),
- научно-методические (диссертации, кандидатские работы),
- дополнительные текстовые и иллюстративные материалы (лабораторные работы, лекции),
- системы тестирования (тесты– электронная проверка знаний),
- электронные полнотекстовые библиотеки;
- электронные периодические издания сферы образования;
- электронные оглавления и аннотации статей периодических изданий сферы образования,
- электронные архивы выпусков.

Содержание работы:

ЗАДАНИЕ 1

1. Загрузите Интернет.
2. В строке поиска введите фразу «каталог образовательных ресурсов».
3. Перечислите, какие разделы включают в себя образовательные ресурсы сети Интернет. 1.
2.
3.
...
12.
4. Охарактеризуйте любые три

Название	Характеристика

ЗАДАНИЕ 2

С помощью Универсального справочника-энциклопедии найдите ответы на следующие вопросы:

Вопрос	Ответ
1) укажите время утверждения григорианского календаря	
2) каков диаметр пылинки	
3) укажите смертельный уровень звука	
4) какова температура кипения железа	
5) какова температура плавления йода	
6) укажите скорость обращения Земли вокруг Солнца	
7) какова масса Земли	
8) какая гора в Австралии является самой высокой	
9) дайте характеристику народа кампа	
10) укажите годы правления Ивана III	
11) укажите годы правления Екатерины II	
12) укажите годы правления Ивана IV	
13) укажите годы правления Хрущева Н.С.	
14) в каком году был изобретен первый деревянный велосипед	

ЗАДАНИЕ 3. Ответьте на вопросы

1. Что Вы понимаете под информационными ресурсами?
2. Перечислите параметры для классификации информационных ресурсов.
3. Что понимают под образовательными информационными ресурсами?
4. Что можно отнести к образовательным электронным ресурсам?

ЗАДАНИЕ 4. Сделайте вывод о проделанной работе

№ 5 Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности

Практическая работа № 10-12

Использование сред имитационного моделирования (виртуальных лабораторий) для проведения компьютерного эксперимента в учебной деятельности

Цели занятия: Ознакомить учеников с основными понятиями темы: вычислительные возможности компьютера, назначение математических моделей, компьютерная математическая модель, вычислительный эксперимент, управление на основе моделей, имитационное моделирование, объектно-информационные модели;

Формирование самостоятельности и ответственности при изучении нового материала, воспитание информационной культуры у учащихся;

Развить умение определять и анализировать объем информации

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Современным инструментом для информационного моделирования является компьютер. Конечно, на компьютере можно писать тексты (строить вербальные модели), рисовать карты и схемы (графические модели), строить таблицы (табличные модели). Но при таком использовании компьютера в моделировании его возможности проявляются не в полной мере.

Для моделирования на компьютере главной является его способность к быстрому счету. **Современные компьютеры считают со скоростями в сотни тысяч, миллионы и даже миллиарды операций в секунду.**

Учитывая, что расчеты производятся над многозначными числами (10-20 десятичных цифр), вычислительные возможности компьютера феноменальны. Эти возможности проявляются, прежде всего, при компьютерном математическом моделировании.

Для чего нужны математические модели

Многие процессы, происходящие в природе, технике, экономических и социальных системах, описываются сложными математическими соотношениями. Это могут быть уравнения, системы уравнений, системы неравенств и пр., которые **являются математическими моделями описываемых процессов.**



Математическая модель — это описание моделируемого процесса на языке математики.

В прежние времена, до появления ЭВМ, ученые стремились создавать такие математические модели, которые можно было бы просчитать вручную или с помощью несложных вычислительных механизмов. Поэтому математические модели были относительно простыми. Но простая модель не всегда хорошо описывает процесс. Ошибка расчетов по такой модели может быть слишком большой и полностью обесценить результат.

Еще в XVIII-XIX веках ученые-математики начали изобретать методы решения таких математических задач, которые не удавалось решить точно, аналитически. Например, вы знаете, что квадратное уравнение всегда можно решить точно, а вот

кубическое — уже не всегда. Такие методы называются численными методами. Они сводят решение любой задачи к последовательности арифметических операций. Но эта цепочка арифметических вычислений может быть очень длинной. И чем точнее мы хотим получить решение, тем она длиннее.

Может оказаться, что для решения сложной задачи численным методом ученому потребуется вся жизнь. А может и этого не хватить! И какой смысл, например, начинать расчет прогноза погоды на завтрашний день, если для этого потребуется несколько лет работы?

Компьютерная математическая модель

Появление компьютеров сняло эти проблемы. Стало возможным **проводить расчеты сложных математических моделей за приемлемое время**. Например, рассчитать погоду на завтрашний день до его наступления. Ученые перестали себя ограничивать в сложности создаваемых математических моделей, полагаясь на быстродействие компьютеров.



Компьютерная математическая модель — это программа, реализующая расчеты состояния моделируемой системы по ее математической модели.

Что такое вычислительный эксперимент



Использование компьютерной математической модели для исследования поведения объекта моделирования называется вычислительным экспериментом. Говорят также: численный эксперимент.



Вычислительный эксперимент в некоторых случаях может заменить реальный физический эксперимент.

Впечатляющий пример использования такой возможности — прекращение испытаний ядерного оружия, которые сопровождалось значительным экологическим ущербом. Благодаря очень точным математическим моделям и мощным компьютерам стало возможно просчитать все последствия, к которым приводит изменение в конструкции ядерной бомбы. Образно говоря, удалось «взорвать бомбу» внутри компьютера, ничего не разрушив.

Важным свойством компьютерных математических моделей является **возможность визуализации результатов расчетов**. Этим целям служит использование компьютерной графики.

Представление результатов в наглядном виде — важнейшее условие для их лучшего понимания. Например, результаты расчетов распределения температуры в некотором объекте можно представить в виде его разноцветного изображения: участки с самой высокой температурой окрасить в красный цвет, а с самой холодной — в синий. Участки с промежуточными значениями температуры окрашиваются в цвета спектра, равномерно переходящие от красного к синему (рис. 2.7).

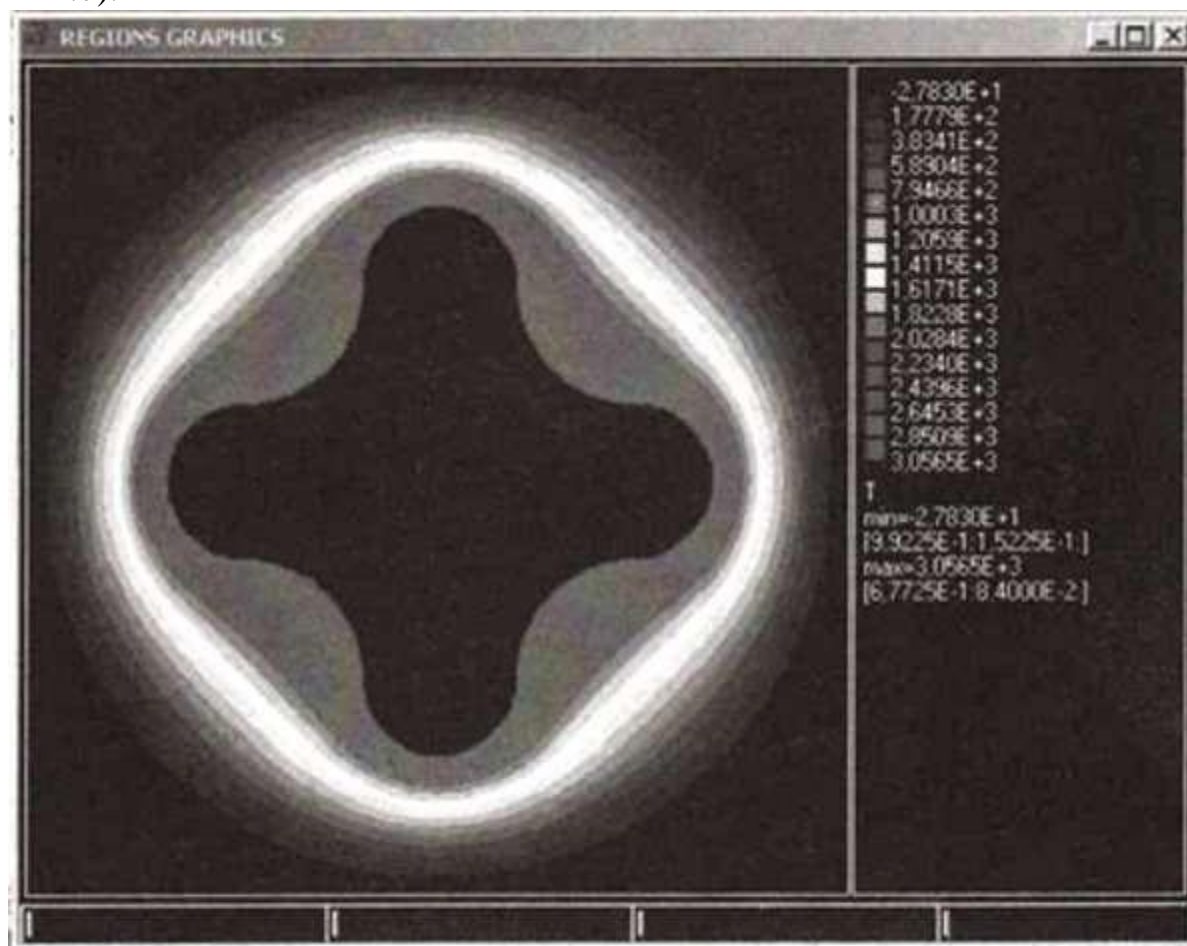


Рис. 2.7. Графическое представление результатов расчетов распределения температур по сечению твердотопливного ракетного двигателя

Для изображения изменяющихся со временем (динамических) результатов используют **графическую анимацию**.



Компьютерная графика позволяет человеку в процессе проведения численного эксперимента «заглянуть» в недоступные места исследуемого объекта. Можно получить изображение любого сечения объекта сложной формы с отображением рассчитываемых характеристик: температурных полей, давления и пр. В реальном физическом эксперименте такое можно сделать далеко не всегда. Например, невозможно выполнить измерения внутри работающей доменной печи или внутри звезды. А на модели это сделать можно.

Управление на основе моделей

Еще одно важное направление компьютерного математического моделирования связано с использованием компьютеров в управлении. **Компьютеры используют для управления работой химических реакторов** на заводах, атомных реакторов на электростанциях, ускорителей элементарных частиц в физических лабораториях, полета автоматических космических станций и т. д.

Управляя производственной или лабораторной установкой, компьютер должен просчитывать ее характеристики для того, чтобы вовремя снять показания с датчиков или оказать управляющее воздействие: включить реле, открыть клапан и т. п.

Все расчеты производятся по заложенным в программу управления математическим моделям. Важно, чтобы результаты этих расчетов получались в режиме реального времени управляемого процесса.

Имитационное моделирование



Имитационное моделирование — особая разновидность моделирования на компьютере.

Имитационная модель воспроизводит поведение сложной системы, элементы которой могут вести себя случайным образом. Иначе говоря, поведение которых заранее предсказать нельзя.

Такое поведение в математике называется стохастическим. Из курса физики вам знакомо явление броуновского движения: хаотического перемещения легких частиц на поверхности жидкости из-за неравномерных ударов молекул с разных сторон. Нельзя точно рассчитать траекторию броуновской частицы, но движение частицы можно симитировать на экране компьютера. Отсюда и происходит название — имитационная модель.



К имитационным моделям относятся модели систем массового обслуживания: например, системы торговли, автосервиса, скорой помощи, в которых появление заявок на обслуживание и длительность обслуживания одной заявки — события случайные.

Задачи, решаемые с помощью имитационных моделей систем массового обслуживания, заключаются в поиске режимов работы служб сервиса (магазинов, автозаправок и пр.), уменьшающих время ожидания клиентов.

Еще одним популярным объектом для имитационного моделирования являются транспортные системы, сеть городских дорог, перекрестки, светофоры, автомобили. Модели имитируют движение транспортных потоков по городским улицам (рис. 2.8).

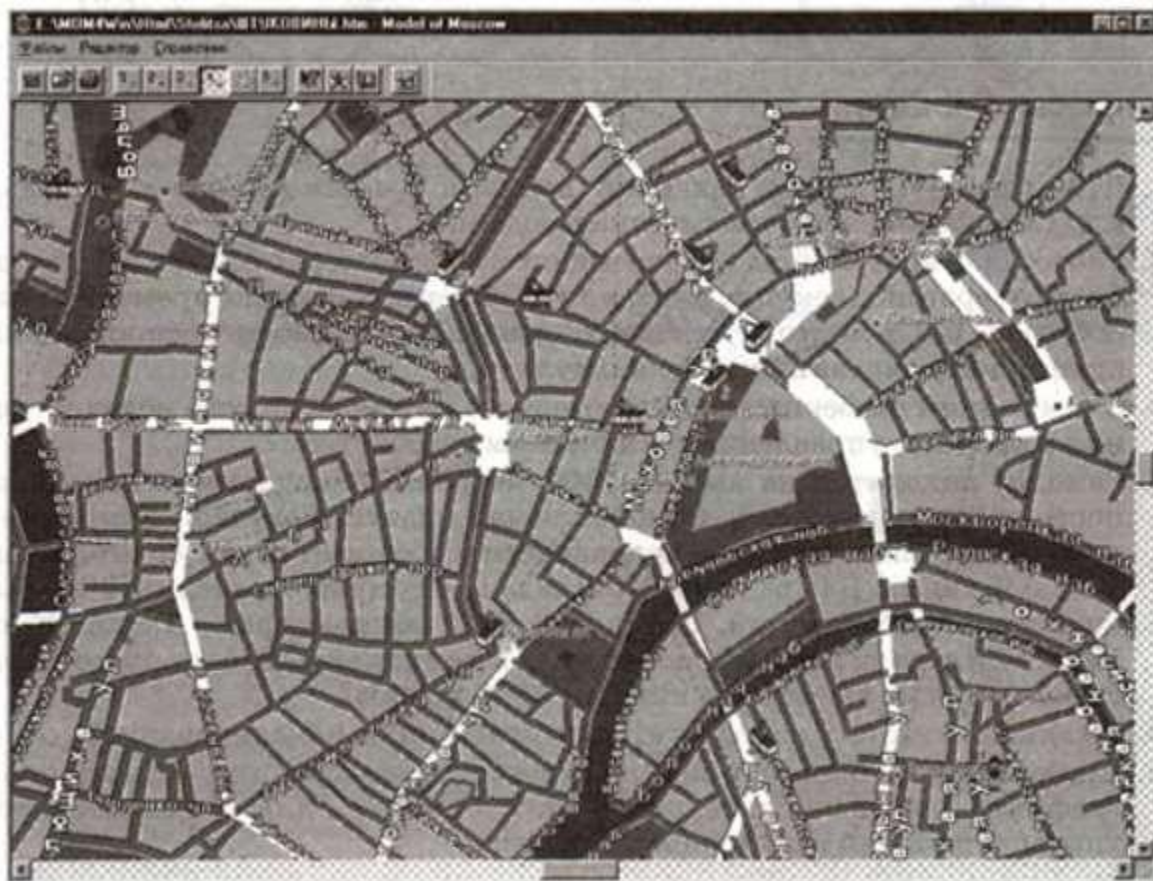


Рис. 2.8. Окно программы, имитирующей движение городского транспорта

Эксперименты на такой модели позволяют найти режимы управления движением (работа светофоров), уменьшающие возможность возникновения пробок. Работа имитационной модели всегда визуализируется на экране компьютера.

Практическая работа № 13-15

Системы счисления

Цели занятия: научиться переводить из одной системы счисления в другую.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

1. Восьмеричная система счисления. Используется восемь цифр: 0, 1, 2, 3, 4, 5, 6, 7. Употребляется в ЭВМ как вспомогательная - для записи информации в сокращенном виде. Для представления одной цифры восьмеричной системы используется три двоичных разряда (триада) (Таблица 1).

Шестнадцатеричная система счисления. Для изображения чисел употребляются 16 цифр. Первые десять цифр этой системы обозначаются цифрами от 0 до 9, а старшие шесть цифр - латинскими буквами: 10-A, 11-B, 12-C, 13-D, 14-E, 15-F. Шестнадцатеричная система используется для записи информации в сокращенном виде. Для представления одной цифры шестнадцатеричной системы счисления используется четыре двоичных разряда (тетрада) (Таблица 1).

Перевод чисел в десятичную систему осуществляется путем составления степенного ряда с основанием той системы, из которой число переводится. Затем подсчитывается значение суммы.

Пример.

а) Перевести $10101101.101_2 \rightarrow "10"$ с.с.

Здесь и в дальнейшем при одновременном использовании нескольких различных систем счисления основание системы, к которой относится число, будем указывать в виде нижнего индекса.

$$10101101.101_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 173.625_{10}$$

б) Перевести $703.04_8 \rightarrow "10"$ с.с.

$$703.04_8 = 7 \cdot 8^2 + 0 \cdot 8^1 + 3 \cdot 8^0 + 0 \cdot 8^{-1} + 4 \cdot 8^{-2} = 451.0625_{10}$$

в) Перевести $B2E.4_{16} \rightarrow "10"$ с.с.

$$B2E.4_{16} = 11 \cdot 16^2 + 2 \cdot 16^1 + 14 \cdot 16^0 + 4 \cdot 16^{-1} = 2862.25_{10}$$

Перевод целых десятичных чисел в недесятичную систему счисления осуществляется последовательным делением десятичного числа на основание той системы, в которую оно переводится, до тех пор, пока не получится частное меньше этого основания. Число в новой системе записывается в виде остатков деления, начиная с последнего.

Пример.

а) Перевести $181_{10} \rightarrow "8"$ с.с.

$$\begin{array}{r|l} 181 & 8 \\ \hline -176 & 22 \\ \hline 5 & 16 \\ \hline & 6 \end{array}$$

Результат: $181_{10} = 265_8$

б) Перевести $622_{10} \rightarrow "16"$ с.с.

$$\begin{array}{r|l} 622 & 16 \\ \hline -48 & 38 \\ \hline -142 & 32 \\ \hline -128 & 6 \\ \hline 14 & \end{array}$$

Результат: $622_{10} = 26E_{16}$

Для перевода восьмеричного или шестнадцатеричного числа в двоичную форму достаточно заменить каждую цифру этого числа соответствующим трехразрядным двоичным числом (триадой) (Таб. 1) или четырехразрядным двоичным числом (тетрадой) (Таб. 1), при этом отбрасывают ненужные нули в старших и младших разрядах.

Пример.

а) Перевести $305.4_8 \rightarrow "2"$ с.с.

$$\begin{array}{ccccccc} 3 & 0 & 5 & . & 4 & & \\ \hline 011 & 000 & 101 & & 100 & & \end{array} = 11000101.1_2$$

б) Перевести $7B2.E_{16} \rightarrow "2"$ с.с.

$$\begin{array}{ccccccc} 7 & B & 2 & . & E & & \\ \hline 0111 & 1011 & 0010 & & 1110 & & \end{array} = 11110110010.111_2$$

Для перехода от двоичной к восьмеричной (шестнадцатеричной) системе поступают следующим образом: двигаясь от точки влево и вправо, разбивают двоичное число на группы по три (четыре)

разряда, дополняя, при необходимости нулями крайние левую и правую группы. Затем триаду (тетраду) заменяют соответствующей восьмеричной (шестнадцатеричной) цифрой.

Пример.

а) Перевести $1101111001.1101_2 \rightarrow$ "8" с.с.

$$\underbrace{001}_1 \underbrace{101}_5 \underbrace{111}_7 \underbrace{001}_1 . \underbrace{110}_6 \underbrace{100}_4 = 1571.64_8$$

б) Перевести $11111111011.100111_2 \rightarrow$ "16" с.с.

$$\underbrace{0111}_7 \underbrace{1111}_F \underbrace{1011}_B . \underbrace{1001}_9 \underbrace{1100}_C = 7FB.9C_{16}$$

Перевод из восьмеричной в шестнадцатеричную систему и обратно, осуществляется через двоичную систему с помощью триад и тетрад.

Пример. Перевести $175.24_8 \rightarrow$ "16" с.с.

$$\underbrace{1}_7 \underbrace{7}_5 . \underbrace{2}_4 \underbrace{4}_8 = 1111101.0101_2 = \underbrace{0111}_7 \underbrace{1101}_D . \underbrace{0101}_5_2 = 7D.5_{16}$$

Результат: $175.24_8 = 7D.5_{16}$.

Содержание работы:

Задание 1. Перевести следующие числа в десятичную систему счисления:

в) 563.44_8 ; г) 721.35_8 ; д) $1C4.A_{16}$; е) $9A2F.B5_2$

Задание 2. Перевести следующие числа из "10" с.с в "8", "16" с.с

а) 463; б) 1209; в) 362; г) 3925; д) 11355.

Задание 3. Перевести следующие числа из "10" с.с в ", "8", "16" с.с (точность вычислений - 5 знаков после точки):

а) 0.0625; б) 0.345; в) 0.225; г) 0.725; д) 217.375; е) 31.2375

Задание 4. Перевести следующие числа в двоичную систему счисления:

а) 1725.326_8 ; б) 341.34_8 ; в) $7BF.52A_{16}$; г) $3D2.C_{16}$.

Задание 5. Сделайте вывод о проделанной работе

Практическая работа № 16-18

Цепочки (конечные последовательности), деревья, списки, графы, матрицы (массивы), псевдослучайные последовательности

Цели занятия: - демонстрация алгоритма работы с заданиями;

- использование методики на занятиях;

- повышение мотивации к изучению информатики и ИКТ.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Дано целое положительное число N , не превосходящее 1000. Необходимо определить, является ли это число степенью числа 3. То есть требуется определить, существует ли такое целое число K , что $3^K = N$, и вывести это число либо сообщение, что такого числа не существует.

Для решения этой задачи ученик написал программу, но, к сожалению, его программа оказалась неверной. Ниже эта написанная им программа для Вашего удобства приведена на пяти языках программирования.

Бейсик	Python
<pre>DIM N, K AS INTEGER INPUT N K = 0 WHILE K MOD 3 = 0 K = K + 1 N = N \ 3 WEND IF N > 0 THEN PRINT K ELSE PRINT "Не существует" END IF END</pre>	<pre>n = int(input()) k = 0 while k%3 == 0: k = k + 1 n = n // 3 if n > 0: print(k) else: print("Не существует")</pre>
Алгоритмический язык	Паскаль
<pre>алг нач цел n, k ввод n k := 0 нц пока mod(k, 3) = 0 k := k + 1 n := div(n, 3) кц если n > 0 то вывод k иначе вывод "Не существует" все кон</pre>	<pre>var n, k: integer; begin read(n); k := 0; while k mod 3 = 0 do begin k := k + 1; n := n div 3; end; if n > 0 then writeln(k) else writeln('Не существует') end.</pre>
Си	
<pre>#include <stdio.h> int main(){ int n, k; scanf("%d", &n); k = 0; while (k%3 == 0) { k = k + 1; n = n / 3; } if (n > 0) printf("%d", k); else printf("Не существует"); return 0; }</pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе числа 9.
2. Приведите пример числа, при вводе которого приведенная программа напечатает то, что требуется.

3. Найдите в программе все ошибки (их может быть одна или несколько).
Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде.

Достаточно указать ошибки и способ их исправления для одного языка программирования.

Обратите внимание: нужно исправить приведённую программу, а не написать свою. Можно только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

Разбор 24 задания

1. При вводе числа 9 программа выведет число 1.
2. Примеры чисел, при вводе которых программа выводит корректный ответ: 2, 3. Других чисел нет.
3. После выполнения программы при любом введённом n значение k будет равно 1 (тело цикла выполнится ровно 1 раз).

В результате программа напечатает либо 1 (если $n \geq 3$), либо «Не существует» (в противном случае). Таким образом, программа выводит корректный ответ, только если введено 2 или 3. Для правильного ответа достаточно указать любое из этих чисел.

4. Программа содержит две ошибки:
 - 1) неверное условие цикла;
 - 2) неверное условие при печати результата.

Пример исправления для языка Паскаль:

Первая
ошибка: `while k
mod 3 = 0 do begin`
Исправленная
строка:
`while n mod 3 = 0 do begin`

Вторая
ошибка: `if n>0
then`
Исправленная
строка:
`if n=1 then`

После исправления первой ошибки в результате выполнения цикла значение переменной n будет равно $n_0/(3k)$, где n_0 – введённое пользователем значение;

k – максимальный показатель степени, при котором $3k$ является делителем числа n_0 . Число n_0 является степенью числа 3, если $n_0 = 3k$, т.е. $n_0/(3k) = 1$.

В программах на других языках ошибочные строки и их исправления аналогичны.

Задание 25

Дан целочисленный массив из 40 элементов. Элементы массива могут принимать целые значения от 0 до 10 000 включительно. Опишите на естественном языке или на одном из языков программирования алгоритм, позволяющий найти и вывести количество пар элементов массива, в которых десятичная запись хотя бы одного числа оканчивается на 2. В данной задаче под парой подразумевается два подряд идущих элемента массива. Например, для массива из пяти элементов: 16 3 142 55 22 – ответ: 3.

Исходные данные объявлены так, как показано ниже на примерах для некоторых языков программирования и естественного языка. Запрещается использовать переменные, не описанные ниже, но разрешается не использовать некоторые из описанных переменных.

Бейсик CONST N = 40 DIM A (1 TO N) AS INTEGER DIM I, J, K, AS INTEGER FOR I = 1 TO N INPUT A(I) NEXT I ... END	Python //допускается также использовать //две целочисленные переменные j и k a = [] n = 40 for i in range(0, n): a.append(int(input())) ...
Алгоритмический язык <u>алг</u> <u>нач</u> цел N = 40 целтаб a[1:N] цел i, j, k нц для i от 1 до N ввод a[i] кц ... <u>кон</u>	Паскаль const N = 40; var a: array [1..N] of integer; i, j, k: integer; begin for i := 1 to N do readln(a[i]); ... end.
Си #include <stdio.h> #define N 40 int main() { int a[N]; int i, j, k; for (i = 0; i < N; i++) scanf("%d", &a[i]); ... return 0; }	Естественный язык Объявляем массив A из 40 элементов. Объявляем целочисленные переменные I, J, K. В цикле от 1 до 40 вводим элементы массива A с 1-го по 40-й. ...

В качестве ответа необходимо привести фрагмент программы (или описание алгоритма на естественном языке), который должен находиться на месте многоточия. Можно записать решение также на другом языке программирования (укажите название и используемую версию языка программирования, например, Free Pascal 2.6) или в виде блок-схемы. В этом случае необходимо использовать те же самые исходные данные и переменные, какие были предложены в условии (например, в образце, записанном на естественном языке).

Разбор 25 задания ЕГЭ 2017

На языке Паскаль

```

k := 0;
for i := 1 to N — 1 do
if (a[i] mod 10 = 2) or (a[i + 1] mod 10 =
2) then inc(k);
writeln(k);

```

(вместо строчки inc(k) можно использовать k:=k+1)

На алгоритмическом языке

```

k := 0;
нц для i от 1 до N-1 если mod(a[i],10)=2 или mod(a[i+1],10)=2 то
k

```

```

:
=

```

```

k
+
1

```

```

в
с
е
к
ц

```

```

в
ы
в
о
д

```

```

k

```

На языке Бейсик

```

K = 0
FOR I = 1 TO N — 1
IF (A(I) MOD 10 = 2) OR (A(I + 1) MOD 10 =
2) THEN K = K + 1
E
N
D
IF
N
E
X
T
I
P
R
I
N

```

T
K

На языке Си

```
k = 0;
for (i = 0; i < N - 1; i++)
if (a[i] % 10 == 2 || a[i + 1] %
10 == 2) k++;
printf(«%d», k);
```

На языке Python

```
k = 0
for i in range(0, n - 1):
if (a[i] % 10 == 2 or a[i + 1] % 10 == 2):
k += 1
print(k)
```

На естественном языке

Записываем в переменную K начальное значение, равное 0. В цикле от первого элемента до предпоследнего находим остаток от деления текущего и следующего элементов массива на 10. Если первый или второй из полученных остатков равен 2, увеличиваем переменную K на единицу. После завершения цикла выводим значение переменной K.

Задание 26

Два игрока, Паша и Валя, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Паша. За один ход игрок может добавить в кучу один камень или увеличить количество камней в куче в два раза. Например, имея кучу из 15 камней, за один ход можно получить кучу из 16 или 30 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней. Игра завершается в тот момент, когда количество камней в куче становится не менее 20. Если при этом в куче оказалось не более 30 камней, то победителем считается игрок, сделавший последний ход. В противном случае победителем становится его противник. Например, если в куче было 17 камней и Паша удвоит количество камней в куче, то игра закончится, и победителем будет Валя. В начальный момент в куче было S камней, $1 \leq S \leq 19$. Будем говорить, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника. Описать стратегию игрока – значит описать, какой ход он должен сделать в любой ситуации, которая ему может встретиться при различной игре противника.

Выполните следующие задания.

1. а) При каких значениях числа S Паша может выиграть в один ход? Укажите все такие значения и соответствующие ходы Паши. б) У кого из игроков есть выигрышная стратегия при $S = 18, 17, 16$? Опишите выигрышные стратегии для этих случаев.

2. У кого из игроков есть выигрышная стратегия при $S = 9, 8$? Опишите соответствующие выигрышные стратегии.

3. У кого из игроков есть выигрышная стратегия при $S = 7$? Постройте дерево всех партий, возможных при этой выигрышной стратегии (в виде рисунка или таблицы). На рёбрах дерева указывайте, кто делает ход; в узлах – количество камней в позиции.

Разбор 26 задания ЕГЭ 2017

1. а) Паша может выиграть, если $S = 19$ или $S = 10, 11, 12, 13, 14, 15$. При $S = 19$ первым ходом нужно добавить в кучу один камень, при остальных указанных значениях S нужно удвоить количество камней.

б) При $S = 16, 17$ или 18 удваивать количество камней не имеет смысла, так как после такого хода выигрывает противник. Поэтому можно считать, что единственный возможный ход – это добавление в кучу одного камня.

При $S = 18$ после такого хода Паши в куче станет 19 камней. В этой позиции ходящий (т.е. Валя) выигрывает (смотрите пункт 1а): при $S = 18$ Паша (игрок, который должен ходить первым) проигрывает.

Выигрышная стратегия есть у Вали.

При $S = 17$, после того как Паша своим первым ходом добавит один камень, в куче станет 18 камней. В этой позиции ходящий (т.е. Валя) проигрывает (смотрите выше): при $S = 17$ Паша (игрок, который должен ходить первым) выигрывает. Выигрышная стратегия есть у Паши.

При $S = 16$ выигрышная стратегия есть у Вали. Действительно, если Паша первым ходом удваивает количество камней, то в куче становится 32 камня, и игра сразу заканчивается выигрышем Вали. Если Паша добавляет один камень, то в куче становится 17 камней. Как мы уже знаем, в этой позиции игрок, который должен ходить (т.е. Валя), выигрывает.

Во всех случаях выигрыш достигается тем, что при своём ходе игрок, имеющий выигрышную стратегию, должен добавить в кучу один камень.

Можно нарисовать деревья всех возможных партий для указанных значений S . Другая возможность – (1) указать на то, что удваивать кучу не имеет смысла, и (2) последовательно сводить случай $S = 18$ к случаю $S = 19$, случай $S = 17$ – к случаю $S = 18$ и т.д.

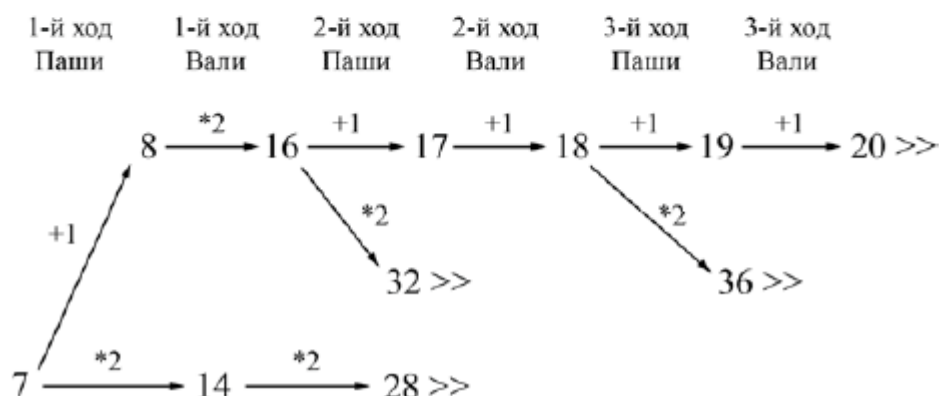
2. При $S = 9$ или 8 выигрышная стратегия есть у Паши. Она состоит в том, чтобы удвоить количество камней в куче и получить кучу, в которой будет соответственно 18 или 16 камней. В обоих случаях игрок, который будет делать ход (теперь это Валя), проигрывает (смотрите пункт 1б).

3. При $S = 7$ выигрышная стратегия есть у Вали. После первого хода Паши в куче может стать либо 8, либо 14 камней. В обеих этих позициях выигрывает игрок, который

будет делать ход (теперь это Валя). Случай $S = 8$ рассмотрен в пункте 2, случай $S = 14$ рассмотрен в пункте 1а.

В таблице изображено дерево возможных партий при описанной стратегии Вали. Заключительные позиции (в них выигрывает Валя) подчёркнуты. На рисунке это же дерево изображено в графическом виде (оба способа изображения дерева допустимы).

Положения после очередных ходов						
И.п.	1-й ход Паши (все ходы)	1-й ход Вали (только ход по стратегии)	2-й ход Паши (все ходы)	2-й ход Вали (только ход по стратегии)	3-й ход Паши (все ходы)	3-й ход Вали (только ход по стратегии)
7	$7+1=8$	$8*2=16$	$16+1=17$	$17+1=18$	$18+1=19$	$19+1=20$
			$16*2=32$		$18*2=36$	
	$7*2=14$	$14*2=28$				



Дерево всех партий, возможных при Валиной стратегии. Знаком >> обозначены позиции, в которых партия заканчивается.

Задание 27

Предлагается два задания с похожими условиями: задание А и задание Б. Можно решать оба задания или одно из них по своему выбору. Задание Б более сложное, его решение оценивается выше. Итоговая оценка выставляется как максимальная из оценок за задания А и Б.

Задание А. Имеется набор данных, состоящий из 6 пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0. Напишите программу для решения этой задачи. В этом варианте задания оценивается только правильность программы, время работы и размер использованной памяти не имеют значения.

Максимальная оценка за правильную программу— 2 балла.

Задание Б. Имеется набор данных, состоящий из пар положительных целых чисел. Необходимо выбрать из каждой пары ровно одно число так, чтобы сумма всех выбранных чисел не делилась на 3 и при этом была максимально возможной. Если получить требуемую сумму невозможно, в качестве ответа нужно выдать 0.

Напишите программу для решения этой задачи.

Постарайтесь сделать программу эффективной по времени и используемой памяти (или хотя бы по одной из этих характеристик).

Программа считается эффективной по времени, если время работы программы пропорционально количеству пар чисел N , т.е. при увеличении N в k раз время работы программы должно увеличиваться не более чем в k раз.

Программа считается эффективной по памяти, если размер памяти, использованной в программе для хранения данных, не зависит от числа N и не превышает 1 килобайта.

Максимальная оценка за правильную программу, эффективную по времени и памяти, – 4 балла.

Максимальная оценка за правильную программу, эффективную по времени, но неэффективную по памяти, – 3 балла.

Как в варианте А, так и в варианте Б программа должна напечатать одно число – максимально возможную сумму, соответствующую условиям задачи (или 0, если такую сумму получить нельзя).

НАПОМИНАЕМ! Не забудьте указать, к какому заданию относится каждая из представленных Вами программ.

Перед текстом программы кратко опишите Ваш алгоритм решения, укажите использованный язык программирования и его версию (например, Free Pascal 2.6.4).

Входные данные

Для варианта А на вход программе подаётся шесть строк, каждая из которых содержит два натуральных числа, не превышающих 10 000.

Пример входных данных для варианта А:

```
1 3
5 12
6 9
5 4
3 3
1 1
```

Для варианта Б на вход программе в первой строке подаётся количество пар N ($1 \leq N \leq 100\,000$). Каждая из следующих N строк содержит два натуральных числа, не превышающих 10 000.

Пример входных данных для варианта Б:

```
6
1 3
5 12
6 9
5 4
3 3
1 1
```

Пример выходных данных для приведённых выше примеров входных данных:
32

Разбор 27 задания ЕГЭ 2017

Задание Б.

Сначала рассмотрим решение для более общего задания (вариант Б).

Решение 1.

Чтобы получить максимально возможную сумму, будем брать из каждой пары самое большое число. Если полученная при этом сумма будет делиться на 3, её необходимо уменьшить. Для этого достаточно в одной из пар, где числа имеют разные остатки при делении на 3, заменить ранее выбранное число на другое число из той же пары. При этом разница между числами в паре должна быть минимально возможной. Если во всех парах оба числа имеют одинаковый остаток при делении на 3, получить нужную сумму невозможно.

Программа читает все данные один раз. В каждой паре определяется большее число Max и разность между большим и меньшим числами пары D . После обработки очередной пары программа хранит два числа: s – сумму всех максимальных элементов прочитанных пар и D_min – наименьшую возможную разность D , не кратную 3. Окончательным ответом будет значение s , если оно не делится на 3, и $s - D_min$ в противном случае. Если s делится на 3, а D_min не определено (разность между числами во всех парах кратна 3), ответ в соответствии с условиями задачи считается равным 0.

Программа 1. Пример правильной и эффективной программы для задания Б на языке Паскаль.

```
const
  aMax = 10000; {наибольшее возможное число в исходных
данных} var
  N: longint;
  {количество пар} a,
  b: longint; {пара
чисел}
  Max: longint; {максимум
в паре} Min: longint;
  {минимум в паре}
  s: longint; {сумма выбранных чисел}
  D_min: longint; {минимальная разность Max-Min не
кратная 3} i: longint;
begin
  s := 0;
  D_min :=
aMax + 1;
  readln(N);
  for i := 1 to N
  do begin
    readln(a, b);
    if a > b then begin Max:=a;
      Min:=b end else begin
      Max:=b; Min:=a end;
    s := s + Max;
```



```

    if ((Max — Min) mod 3 > 0) and (Max — Min <
      D_min) then D_min := Max — Min
end;
if s mod 3 = 0 then begin
  if D_min > aMax
  then s := 0 else s := s
  — D_min
end;
writeln(s)
end.

```

Решение 2.

Возможно и решение, основанное на другой идее, а именно будем хранить для каждого прочитанного набора пар три суммы (s_0 , s_1 , s_2) – максимальные суммы элементов пар, имеющие при делении на 3 соответственно остатки 0, 1 и 2. При обработке очередной пары (a_1 , a_2) эти суммы обновляются. Для этого достаточно рассмотреть суммы s_0+a_1 , s_1+a_1 , s_2+a_1 , s_0+a_2 , s_1+a_2 , s_2+a_2 и для каждого возможного остатка от деления на 3 выбрать в качестве нового значения s_0 , s_1 или s_2 значение наибольшей из указанных сумм, дающей данный остаток. Окончательным ответом будет большая из сумм s_1 и s_2 .

Эта идея приводит к более громоздкой реализации, но все основные требования по эффективности в ней выполнены, поэтому подобное решение при отсутствии ошибок можно оценить максимальным количеством баллов.

Ниже приводится пример основанной на этом принципе программы на языке Паскаль.

Программа 2. Пример правильной и эффективной программы для задания Б на языке Паскаль

```

var
  N: longint; { количество пар }
  a: array[1..2] of longint; { пара
чисел } s_old, s_new:
  array[0..2] of longint;
  { суммы с соответствующими остатками от
  деления на 3 } i, j, k, r: longint;
begin
  readln(N);
  for j := 0
    to 2 do
    s_old[j]
    := 0;
  for i := 1 to N
  do begin
    readln(a[1],
    a[2]); for j :=
    0 to 2 do
      s_new[j] := 0;

```

```

for k := 1 to 2 do
  begin for j := 0
    to 2 do begin
      if (s_old[j] > 0) or (i = 1)
        then begin r := (s_old[j] +
          a[k]) mod 3;
          if s_new[r] < s_old[j] +
            a[k] then s_new[r] :=
              s_old[j] + a[k]
        end;
      end;
    end;
  end;
  s_old
:= s_new
end;
if s_new[1] >
  s_new[2] then
  writeln(s_new[1])
else
  writeln(s_new[2]);
{если решения не существует, tos_new[1] и s_new[2] окажутся равными
нулю} end.

```

Задание А.

Это задание можно выполнить «в лоб»: сохранить в массиве все исходные данные, перебрать все возможные способы выбора одного элемента из каждой пары и найти максимальную сумму, соответствующую условиям задачи.

Ниже приводится пример такого решения Пример решения задачи А на языке Паскаль.

```

var
a: array[1..6, 1..2] of longint;
i1, i2, i3, i4, i5, i6:
longint; s, sMax:
longint;
begin
  for i1:= 1 to 6 do readln(a[i1,1],
    a[i1,2]); sMax := 0;
  for i1:=1
  to 2 do
  for i2:=1
  to 2 do
  for i3:=1
  to 2 do

```

```

for i4:=1
to 2 do
for i5:=1
to 2 do
for i6:=1 to 2 do begin
s:=a[1,i1]+a[2,i2]+a[3,i3]+a[4,i4]+a[5,i5]
+a[6,i6];
if (s mod 3 <> 0) and (s > sMax) then
sMax := s end;
writeln
n(sMax)
end.

```

Практическая работа № 19-21

Индуктивное определение объектов. Вычислимые функции, полнота формализации понятия вычислимости, универсальная вычислимая функция; диагональное доказательство не существования.

Цели занятия: познакомиться с формализацией понятия вычислимости, ввести понятие универсальной вычислимой функции.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Теория вычислимых функции разрабатывалась А. Чёрчем, и целью ее построения являлось получение точного (формального) определения понятия «алгоритм».

Для того, чтобы дать точное понятие алгоритма, необходимо определить, как задаются данные, с которыми будет работать Исполнитель, и как задаются элементарные шаги, из которых состоит алгоритм.

Универсальный способ представления различных данных – это представление их словами в некотором конечном алфавите. Действительно:

- если объект число, его можно представить в десятичной или двоичной форме, т. е. цепочкой символов алфавита $\{0,1\}$, или алфавита $\{0..9\}$.
- если объект – программа, то она является цепочкой символов в алфавите, содержащем буквы, цифры и специальные символы.
- если объект – изображение, то он представляется массивом пикселей, а каждый пиксель – тремя числами (интенсивностями красного, зеленого и синего цветов), т. е. изображение так же может быть закодировано строкой символов.
- современные высококачественные системы цифровой записи звука показывают, что и этот объект может быть адекватно описан строкой символов.

Таким образом, можно считать, что *алгоритм – это преобразование слов из заданного алфавита в результирующее слово.* Каким бы ни был конечный алфавит, любой его символ может быть закодирован с помощью двоичного алфавита. Иначе говоря, входные данные алгоритма могут быть представлены конечной цепочкой битов. То же самое можно сказать и о выходных данных. Каждая цепочка битов может интерпретироваться как целое неотрицательное число. Из этого можно сделать важный вывод: **любому алгоритму может быть поставлена в соответствие некоторая функция, отображающая множество неотрицательных целых чисел в себя.** Однако мы не утверждаем обратного. Существуют функции, не вычисляемые ни каким алгоритмом.

Вычислимой называют функцию, вычисляемую некоторым алгоритмом.

Вычислимая функция определяется на множестве целых неотрицательных чисел, результат функции так же должен принадлежать множеству целых неотрицательных чисел.

Итак, вычислимая функция – функция, для которой можно построить некоторый алгоритм. Алгоритм дает процедуру отыскания значений вычислимой функции. Но как отличить вычислимую функцию от невычислимой? Можно ли традиционным математическим языком описать общий вид вычислимых функций? А, написав, навести некоторый порядок среди алгоритмов, точнее ввести некий канонический вид написания алгоритмов, соответствующий строению вычислимых функций (мы помним, что для одной функции можно написать бесконечно много эквивалентных алгоритмов).

Для того чтобы внести ясность в эту проблему, поступим следующим образом.

1. Возьмем несколько простых базовых функций, для которых очевидно, что они могут быть вычислены, например, с помощью машины Тьюринга и составим из них набор **P**.

2. Введем операции над функциями. Операции получают сами функции (а не значения функций) в качестве операндов и дают в результате новые функции. Эти операции так же должны быть очевидны с математической точки зрения, т. е. не должно возникать сомнений в том, что из вычислимых функций с помощью этих операций получаются вновь вычислимые функции. Обозначим набор этих операций **O**.

3. Произведем (мысленно) все возможные операции набора **O** над базовыми функциями набора **P**, при этом будем по-разному комбинировать их в качестве аргументов. Все новые полученные функции включим в набор **P** и вновь произведем все возможные операции из набора **O**, подставляя в виде аргументов и по-разному комбинируя все возможные функции из уже расширенного набора **P**. Полученные при этом новые функции, так же включим в **P** и т. д.

При этом мы не ограничиваем сверху количество исполнений п.3., то таким образом будем получать все новые и новые функции. Иначе говоря, пп.1-3 описывают построение некоторого бесконечного множества функций. Однако, необходимо отметить, что каждая конкретная функция f из этого множества является результатом выполнения конечного числа операций, взятых из набора **O** над базовыми функциями.

Таким образом, процесс построения одной конкретной функции f :

- 1) начинается с исходных данных, выбираемых из базового набора;
- 2) выполняется пошагово (в дискретном времени);
- 3) на каждом шаге выполняется одна из элементарных операций (из набора **O**);
- 4) результат каждого шага строго определен;
- 5) процесс заканчивается через конечное число шагов.

Этот перечень дает нам возможность говорить об алгоритме af построения функции f .

Заметим, что мы не вносим саму функцию f в перечень исходных данных, т. е. не говорим об универсальном алгоритме построения функции. Напротив, для различных функций f_1 и f_2 алгоритмы будут различны. Более того, свойство массовости алгоритмов не выполняется, т. к. исходные данные всегда одни и те же – базовый набор функций. Это позволяет говорить о том, что текст алгоритма (последовательность применения операций из **O** и места подстановок в эти операции операндов из базового набора) задает единственную функцию, а не множество функций. Таким образом, *универсальный алгоритм, который на основе базовой функции построил бы любую функцию не существует.*

С другой стороны, легко преобразовать алгоритм af построения функции в алгоритм вычисления значений функции $f(x_1, x_2, \dots, x_n)$ введением и использованием в тексте исходных данных x_1, x_2, \dots, x_n .

Введем сейчас конкретные базовый набор функций **P** и систему операций **O** для формирования множества функций. В качестве области определения функций возьмем n -кратное декартово произведение множества неотрицательных целых чисел. Сами функции, как и их аргументы также принимают значения из множества неотрицательных целых чисел. Таким образом, мы обеспечиваем возможность, хотя бы потенциальную, вычислять значения функций, с помощью рассмотренных ранее машин Тьюринга.

Базовый набор функций.

$$P = \{ Z(x), S(x), I_n^A(x_1, x_2, \dots, x_n) \}$$

$$Z(x)=0, S(x)=x+1, I_m^n(x_1, x_2, \dots, x_n) = x_m$$

Для этих функций не представляет труда построить машины Тьюринга, вычисляющие их значения по значениям аргументов. Покажем это.

1. Вычисление функции $S(x)=x+1$, увеличение аргумента на единицу. Используем алфавит $A^*=\{0, 1, e\}$, причем x , будем кодировать последовательностью нулей и единиц, так как это принято в двоичном кодировании целых неотрицательных чисел. В момент старта головка машины Тьюринга находится напротив крайней левой ячейки с символом 1. В момент окончания работы головка машины Тьюринга так же должна находиться в крайнем левом положении (напротив крайней левой ячейки с символом 1).

	$q1$	$q2$	$q3$	$q4$
00	$q1$	$R1$	$q3$	$L0$
11	$q1$	$R0$	$q2$	$L1$
ee	$q2$	$L1$	$q4$	Se

2. Вычисление функции $Z(x)=0$, обнуление, функция должна превращать запись любого аргумента в ноль. Эта программа стирает с ленты код x , т. е. заполняет клетки символом e и перед остановкой записывает текущую клетку 0.

	$q1$	$q2$
0e	$q1$	R
1e	$q1$	R
e0	$q2$	S

3. Вычисление функций-проекторов $I_m^n(x_1, x_2, \dots, x_n)$, выбирающих m -й аргумент из последовательности n аргументов

$$I_m^n(x_1, x_2, \dots, x_n) = x_m, 1 \leq m \leq n.$$

Представление последовательности n аргументов зададим в виде записанных один за другим через разделитель – пустую клетку e – двоичных кодов x_i . Программа, написанная для конкретного значения m , действует следующим образом. Сначала стирается (заменяется на $e\dots e$) первый аргумент, затем стирается второй, ..., стирается $(m-1)$ -й аргумент; затем подтверждается m -й аргумент, и после этого стираются оставшиеся аргументы.

$q1$	$q2$...	$qm-1$	qm	$qm+1$...	qn	$qn+1$
0 e $q1$ R	e $q2$ R	...	e $qm-1$ R	0 qm R	e $qm+1$ R	...	e qn R	
1 e $q1$ R	e $q2$ R	...	e $qm-1$ R	1 qm R	e $qm+1$ R	...	e qn R	
e e $q2$ R	e $q3$ R	...	e qm R	e $qm+1$ R	e $qm+2$ R	...	e $qn+1$ R	

Таким образом, этот набор удовлетворяет выдвинутым выше требованиям.

Система операций. В систему операций O входят три операции: суперпозиции σ , примитивной рекурсии ρ и минимизации μ .

1. Операция суперпозиции σ , получая $n+1$ операнд – функции f_0, f_1, \dots, f_n производит результат $f = \sigma(f_0, f_1, \dots, f_n)$. Считая все функции зависящими от одного и того же набора аргументов (можно просто объединить наборы аргументов всех участвующих в операции функций в один набор и, если j -я функция ранее не зависела от i -го аргумента, то мы считаем этот аргумент несущественным для данной функции) мы можем задать формулу вычисления значений вновь образованной функции f :

$$f(x_1, x_2, \dots, x_n) = f_0(f_1(x_1, x_2, \dots, x_k), f_2(x_1, x_2, \dots, x_k), \dots, f_n(x_1, x_2, \dots, x_k)),$$

где k – количество переменных в объединенном наборе переменных функций f_1, f_2, \dots, f_n .

Первый операнд (функция f_0) операции суперпозиции играет отличающуюся от других операндов роль – он задает формирование сложной функции. Поэтому аргументы функции f_0 не входят в перечень аргументов результата операции суперпозиции, но их количество должно быть равно n . Это требование не ограничивает применимость операции суперпозиции, так как при большем ($>n$) количестве аргументов функции f_0 мы всегда можем расширить перечень операндов операции σ , добавив необходимое количество тождественных функций; тогда часть аргументов функции f_0 перейдет в состав аргументов функции результата.

Пример 1. Операция суперпозиции используется, когда основную задачу можно разбить на более простые подзадачи.

$$f(x) = \cos(x) + \sin(x)$$

$$\text{Здесь } f_1 = \cos(x), f_2 = \sin(x), f_0(x_1, x_2) = x_1 + x_2$$

Все рассматриваемые функции являются частичными, т. е. не всюду определенными. Это означает, что могут существовать такие комбинации значений аргументов, для которых значение функции не существует. Например, функция $f(x) = x/2$ определена только на подмножестве четных значений аргумента, а функция $f(x) = x-3$ не определена для значений аргумента 0, 1, 2. В таких случаях и функция-результат f операции суперпозиции не существует для некоторых комбинаций значений аргументов. Точнее $f(a_1, a_2, \dots, a_k)$ не существует, если не существует хотя бы одно из значений $f_1(a_1, a_2, \dots, a_k), \dots, f_n(a_1, a_2, \dots, a_k)$ или, если эти значения существуют и равны b_1, \dots, b_n , но не существует значение $f_0(b_1, \dots, b_n)$. Таким образом, функция f так же является частичной. Подобные замечания можно сделать и в отношении следующих двух операций.

Задача 1. Докажите, что функция $f(x) = x + 3$ является вычислимой, постройте для этого схему ее вычисления с использованием функций базового набора $Z(x), S(x), I_n^a(x_1, \dots, x_n)$.

Переменную x можно получить при помощи специального вида функции-проектора $I_1^1(x) = x$, называемого *тождественной функцией*. Применяя несколько раз функцию $S(x)$ увеличения на единицу получим требуемую функцию.

$$f(x) = S(S(S(I_1^1(x))))$$

2. Операция примитивной рекурсии ρ имеет два операнда $f = \rho(g, h)$. Первый операнд, функция g , зависит от n аргументов, $n \geq 0$, а второй операнд, функция h , имеет в общем случае два дополнительных аргумента, хотя в том и другом случае некоторые аргументы могут быть несущественными. Функция f определяется следующими *уравнениями примитивной рекурсии*:

$$\begin{cases} f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n) \\ f(x_1, \dots, x_n, y+1) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y)) \end{cases}$$

Эти уравнения еще называют *схемой примитивной рекурсии*.

Здесь рекурсия производится по последнему аргументу функции f : ее значение при аргументе $y+1$ вычисляется через ее же значение при аргументе равном y , которое в свою очередь вычисляется через значение функции f при аргументе равном $y-1$, и так далее до значения аргумента, равного 0, при котором процесс определения через себя (возвратный процесс) заканчивается.

Можно, наоборот, процесс вычисления значения $f(x_1, x_2, \dots, x_n, y)$ начинать всегда с 0, получая значение в соответствии с первым уравнением (начальным условием), а затем в соответствии со вторым уравнением вычислять значения функции при аргументах, равных 1, 2, 3 и т. д. до достижения значения y (итерационный процесс).

Приведем несколько примеров использования операций примитивной рекурсии и суперпозиции для пополнения базового набора P новыми функциями.

Пример 2. Сложение неотрицательных целых чисел $Add(x, y) = x + y$ задается следующими уравнениями

$$Add(x, y) = x + y = x + (y - 1) + 1 = Add(x, y - 1) + 1 = S(Add(x, y - 1))$$

$$\begin{cases} Add(x, 0) = I_1^1(x) \\ Add(x, y + 1) = S(Add(x, y)) \end{cases}$$

Здесь функция $g(x) = I_1^1(x) = x$ – тождественная функция, а функция $h = S$ существенным образом зависит только от последнего своего аргумента, а именно от $Add(x, y)$

Пример 3. Умножение $Mult(x, y) = x * y$ задается уравнениями

$$Mult(x, y) = x * y = x * (y - 1) + x = Mult(x, y - 1) + x = Add(x, Mult(x, y - 1))$$

$$\begin{cases} Mult(x, 0) = Z(x) \\ Mult(x, y + 1) = Add(x, Mult(x, y)) \end{cases}$$

Здесь функция $g(x) = Z(x)$ – функция, а функция $h = Add$ от y не зависит

Пример 4. Возведение в степень $Power(x, y) = x^y$ задается уравнениями

$$Power(x, y) = x^y = x^{y-1} * x = Power(x, y - 1) * x = Mult(x, Power(x, y - 1))$$

$$\begin{cases} Power(x, 0) = 1 \\ Power(x, y + 1) = Mult(x, Power(x, y)) \end{cases}$$

Задача 2. Пусть вычисление суммы n первых членов последовательности: 2, 4, 8, 16, 32, ... задается функцией $Sum(n) = 2 + 4 + 8 + \dots + 2n$. Постройте схему примитивной рекурсии для вычисления функции $Sum(n)$, используя следующий набор базовых функций $P = \{ Z(x), S(x), I nm(x_1, x_2, \dots, x_n), Add(x, y), Mult(x, y), Power(x, y) \}$. Покажите подробно вывод формулы и пример вычисления функции для 0 и 3, подставляя указанные значения в построенную формулу.

Поскольку функция Sum зависит только от одного аргумента, перепишем уравнения примитивной рекурсии применительно к функции одного аргумента:

$$\begin{cases} f(0) = \text{const} \\ f(y + 1) = h(y, f(y)) \end{cases}$$

Функция $g(x_1, \dots, x_n)$ не зависит от переменных, а, значит, равна константе (const).

Проведем несложные преобразования:

$$Sum(n) = Sum(n - 1) + 2n = Sum(n - 1) + Power(2, n) = Add(Power(2, n), Sum(n - 1))$$

$$Sum(n+1) = Add(Power(2,n+1),Sum(n)) = Add(Power(2,S(n)),Sum(n))$$

Итак, получаем следующую схему примитивной рекурсии.

$$\begin{cases} Sum(0) = 0 \\ Sum(n+1) = Add(Power(2, S(n)), Sum(n)) \end{cases}$$

Подставим значения 0 и 3 в полученную формулу для $Sum(n)$.

$Sum(0)=0$ Это видно из первого уравнения схемы.

$$Sum(3)=Add(Power(2,S(2)),Sum(2))= Add(Power(2,3),Sum(2)) = Sum(2)+23$$

$$Sum(2)=Add(Power(2,S(1)),Sum(1))= Add(Power(2,2),Sum(1)) = Sum(1)+22$$

$$Sum(1)=Add(Power(2,S(0)),Sum(0))= Add(Power(2,1),Sum(0)) = Sum(0)+21$$

$$Sum(0)=0$$

Выполняя обратный ход, т. е. подставляя значения, полученные ниже в формулу, полученную на предыдущем шаге получаем:

$$Sum(1) = Sum(0)+21 = 0+21 = 21$$

$$Sum(2) = Sum(1)+22 = 21+22 = 43$$

$$Sum(3) = Sum(2)+23 = 43+23 = 66$$

В результате подстановки конкретных значений, видно, что полученная схема примитивной рекурсии решает поставленную задачу.

Задача 3. Определите, какая функция вычисляется следующей схемой примитивной рекурсии

$$\begin{cases} f(x,0) = S(Z(x)) \\ f(x,y+1) = Mult(Add(x,S(y)), f(x,y)) \end{cases}$$

Произведем вычисления по формуле при $y=4$.

$$f(x,4) = Mult(Add(x,S(3)),f(x,3)) = Mult(Add(x,4),f(x,3)) = (x+4) \cdot f(x,3)$$

$$f(x,3) = Mult(Add(x,S(2)),f(x,2)) = Mult(Add(x,3),f(x,2)) = (x+3) \cdot f(x,2)$$

$$f(x,2) = Mult(Add(x,S(1)),f(x,1)) = Mult(Add(x,2),f(x,1)) = (x+2) \cdot f(x,1)$$

$$f(x,1) = Mult(Add(x,S(0)),f(x,0)) = Mult(Add(x,1),f(x,0)) = (x+1) \cdot f(x,0)$$

$$f(x,0) = S(Z(x)) = S(0) = 1$$

Производя обратный ход, получаем:

$$f(x,1) = (x+1) \cdot f(x,0) = (x+1)$$

$$f(x,2) = (x+2) \cdot f(x,1) = (x+2) \cdot (x+1)$$

$$f(x,3) = (x+3) \cdot f(x,2) = (x+3) \cdot (x+2) \cdot (x+1)$$

$$f(x,4) = (x+4) \cdot f(x,3) = (x+4) \cdot (x+3) \cdot (x+2) \cdot (x+1)$$

При произвольном y получаем $f(x,y) = (x+1) \cdot (x+2) \cdot (x+3) \cdot \dots \cdot (x+y)$

Класс функций, получаемых из базовых применением конечного числа операций суперпозиции и примитивной рекурсии называется классом *примитивно-рекурсивных функций* **Рпр**. **Примитивно-рекурсивные функции всюду определены**. Далеко не все функции, значения которых могут быть вычислены относятся к классу примитивно-рекурсивных функций. В операции ρ рекурсия производится по одной переменной. Если построить схему с рекурсией по двум переменным (двойная рекурсия, рекурсия 2-й степени), то с ее помощью будут получаться функции, не принадлежащие в общем случае, к классу примитивно-рекурсивных.

Пример функции, не являющейся примитивно-рекурсивной. Функции Аккермана.

$$\left\{ \begin{array}{l} B(0, y) = 2 + y \\ B(x+1, 0) = sg(x) \\ B(x+1, y+1) = B(x, B(x+1, y)) \\ A(x) = B(x, x) \end{array} \right. , \text{ здесь } sg(x) = \begin{cases} 0, & x = 0 \\ 1, & x > 0 \end{cases}$$

$A(x)$ – функция Аккермана, $B(x, x)$ –диагональная функция Аккермана.

В определении функции $B(x, y)$ рекурсия введена по двум переменным. В теории алгоритмов доказано, что функция Аккермана $A(x)$ растет быстрее, чем любая примитивно-рекурсивная функция.

$$\forall f(x) \in P_{\text{пр}}, \exists n \in \mathbb{N}: \forall x > n \quad A(x) > f(x)$$

Следовательно двух операций: суперпозиции и примитивной рекурсии недостаточно для описания вычислимых функций. Даже введение кратной рекурсии не решает проблему, т. к. для любого k найдется функция, которую можно определить с помощью рекурсии по k переменным, но нельзя определить с помощью рекурсии по $k-1$ переменной (k -рекурсивная, но не $(k-1)$ -рекурсивная).

2. Операция минимизации μ имеет один операнд $f = \mu(g)$. Значения функции f на заданном наборе аргументов x_1, x_2, \dots, x_n получаются следующим образом. Сначала с помощью функции g формируется уравнение $g(x_1, x_2, \dots, x_{n-1}, y) = x_n$, а затем отыскивается его решение относительно переменной y . Если таких решений несколько, то берется минимальное из них (отсюда и название операции), оно и считается значением функции f на данном наборе аргументов.

$$f(x_1, x_2, \dots, x_n) = \mu y (g(x_1, x_2, \dots, x_{n-1}, y) = x_n).$$

Может оказаться, что уравнение не имеет ни одного решения. В этом случае, считается, что функция f не определена на заданном наборе аргументов. Оператор минимизации для функции одной переменной является средством отыскания обратной функции. Для функции многих переменных он играет более сложную роль, но она так же связана с обращением функции.

$$Dec(x) = \begin{cases} x-1, & x > 0 \\ \text{не существует}, & x = 0 \end{cases}$$

Пример 1. Функция уменьшения на единицу

$Dec = \mu(S)$. Здесь $S(x)=x+1$ – функция из базового набора (увеличение на 1).

$Dec(x) = \mu y (S(y) = x)$. Уравнение $S(y) = x$ имеет решение $y=x-1$ при $x>0$ и не имеет решения при $x=0$.

$$Sub(x, y) = \begin{cases} x - y, & x \geq y \\ \text{не существует}, & x < y \end{cases}$$

Пример 2. Функция вычитания двух переменных

$Sub = \mu(Add)$. Здесь $Add(x, y) = x + y$

$Sub(x, y) = \mu y (Add(y, z) = x)$. Уравнение $Add(y, z) = x$ имеет решение $z = x - y$ при $x \geq y$ и не имеет решения при $x < y$.

Вычислим значение этой функции, используя операцию минимизации.

$Sub(7, 2) = ?$

Нужно положить $y = 2$, и придавая переменной z последовательно значения $0, 1, 2, \dots$ каждый раз вычислять сумму $y + z$. Как только она стане равной 7 , то соответствующее значение z принять за значение $Sub(7, 2)$.

$z = 0? 2 + 0 = 2 \neq 7$

$z = 1? 2 + 1 = 3 \neq 7$

$z = 2? 2 + 2 = 4 \neq 7$

$z = 3? 2 + 3 = 5 \neq 7$

$z = 4? 2 + 4 = 6 \neq 7$

$z = 5? 2 + 5 = 7$

Таким образом, $Sub(7, 2) = 5$.

$Sub(3, 4) = ?$

$z = 0? 4 + 0 = 4 \neq 3$

$z = 1? 4 + 1 = 5 \neq 3$

$z = 2? 4 + 2 = 6 \neq 3$

$z = 3? 4 + 3 = 7 \neq 3$

.....

Видим, что данный процесс будет продолжаться бесконечно, следовательно $Sub(3, 4)$ не определено.

Приведенный пример показывает, что оператор минимизации может превратить всюду определенную функцию в частичную.

Множество функций, получаемых применением к базовому набору P конечного числа операций суперпозиции, примитивной рекурсии и минимизации называется множеством *частично-рекурсивных функций* и обозначается $P_{чр}$.

Множество всюду определенных функций из $P_{чр}$ называется множеством *рекурсивных* (или *общерекурсивных функций*) и обозначается P_r . В частности, приведенные выше функции Аккермана являются общерекурсивными.

$P_{пр} \subset P_r \subset P_{чр} \subset P_{ч}$

Существует *теорема Клини о нормальной форме*: Любая вычислимая функция $f(x_1, x_2, \dots, x_n)$ представима в форме

$$f(x_1, x_2, \dots, x_n) = Ff(x_1, x_2, \dots, x_n, \mu_y(Gf(x_1, \dots, x_n, y)=0))$$

где Ff, Gf – примитивно-рекурсивные функции.

Обозначим $\mathbf{P}_{\text{выч}}$ множество вычислимых функций. Все базовые функции из \mathbf{P} являются вычислимыми. В теории алгоритмов доказываются следующие утверждения:

1. Класс $\mathbf{P}_{\text{выч}}$ замкнут относительно операции суперпозиции σ . Иными словами, если в качестве операндов операции суперпозиции взять вычислимые функции, то результат обязательно будет вычислимой функцией.
2. Класс $\mathbf{P}_{\text{выч}}$ замкнут относительно операции примитивной рекурсии ρ .
3. Класс $\mathbf{P}_{\text{выч}}$ замкнут относительно операции минимизации μ .

Следствием этих трех утверждений является вложение $\mathbf{P}_{\text{чр}} \subseteq \mathbf{P}_{\text{выч}}$

Практическая работа № 22-24

Сложность вычисления; проблема перебора. Задание вычислимой функции системой уравнений

Цели занятия: Усвоить понятия: алгоритм как фундаментальное понятие информатики, свойства алгоритмов, основные типы алгоритмов, изучить способы представления алгоритмов, научиться составлять алгоритмы в виде блок – схем, псевдокодов и на языке Паскаль.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Алгоритм — точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Решение любой задачи на ЭВМ можно разбить на следующие этапы: разработка алгоритма решения задачи, составление программы решения задачи на алгоритмическом языке, ввод программы в ЭВМ, отладка программы (исправление ошибок), выполнение программы на ПК, анализ полученных результатов.

Первый этап решения задачи состоит в разработке алгоритма.

Алгоритм может быть описан одним из трех способов:

- словесным (пример в начале раздела);
- графическим (виде специальной блок-схемы);
- с помощью специальных языков программирования.

Блок-схема – распространенный тип схем, описывающий алгоритмы или процессы, изображая шаги в виде блоков различной формы, соединенных между собой стрелками.

1. **Линейный алгоритм** – это такой алгоритм, в котором все операции выполняются последовательно одна за другой.

2. **Алгоритмы разветвленной структуры** применяются, когда в зависимости от некоторого условия необходимо выполнить либо одно, либо другое действие.

3. **Алгоритмы циклической структуры.**

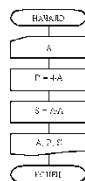
Циклом называют повторение одних и тех же действий (шагов). Последовательность действий, которые повторяются в цикле, называют **телом цикла**.

Циклические алгоритмы подразделяют на алгоритмы с предусловием, постусловием и алгоритмы с конечным числом повторов. В алгоритмах с предусловием сначала выполняется проверка условия окончания цикла и затем, в зависимости от результата проверки, выполняется (или не выполняется) так называемое тело цикла.

Линейные алгоритмы

Линейный алгоритм – это алгоритм, в котором блоки выполняются последовательно сверху вниз от начала до конца.

На **рисунке** приведен **пример** блок-схемы линейного алгоритма вычисления периметра P и площади S квадрата со стороной длины A .



Пример записи алгоритма в виде блок-схемы, псевдокодов и на языке Паскаль. Вычислить среднее геометрическое чисел a и b .

Блок-схема	Псевдокоды	Паскаль
	алг среднее геометрическое вещ a, b, g нач ввод a, b $g := (a * b)^{(1/2)}$ вывод g кон	program Srednee_geometr; var a, b, g : real; begin readln (a, b); $s := \text{sqrt}(a * b)$; writeln (g) end.

Содержание работы:

Задание №1. Построить линейный алгоритм вычисления площади прямоугольника $S=a \cdot b$. Записать алгоритмы в виде блок-схемы, в виде псевдокодов на алгоритмическом языке, а также на языке Паскаль.

Задание №2. Построить линейный алгоритм вычисления значения Y по формуле $Y=(7X+4)(2X-2)$ при $X=3$. Составьте алгоритм самостоятельно, выделяя каждое действие как отдельный шаг. Записать алгоритмы в виде блок-схемы, в виде псевдокодов на алгоритмическом языке, а также на языке Паскаль.

Сделайте вывод о проделанной работе.

Контрольные вопросы:

1. Поясните понятие «алгоритм».
2. В чем состоит особенность описания алгоритмов с помощью структурной схемы и конструкций алгоритмического языка?
3. Перечислите типовые алгоритмические конструкции и объясните их назначение.

Литература

4. Информатика: учебник для начального и среднего профессионального образования. Цветкова Н.С., Великович Л.С. – Академия, 2017 г.
5. Информатика. Практикум для профессий и специальностей технического и социально-экономического профилей. Н. Е. Астафьева, С. А. Гаврилова, под ред. М.С. Цветковой, Академия, 2017 г.

Практическая работа №8.

Разработка несложного алгоритма решения задачи.

Цели занятия:

- изучить примеры построения алгоритмов с использованием конструкций проверки условий, циклов и способов описания структур данных
- освоить принципы решения задач с использованием основных алгоритмических конструкций.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Пример 1.

Определить площадь трапеции по введенным значениям оснований (a и b) и высоты (h).

Запись решения задачи

1):

на алгоритмическом языке:

алг трапеция

вещ a, b, h, s

нач

ввода, b, h

$s := ((a+b)/2) * h$

вывод s

кон

Запись алгоритма в виде блок-схемы (рис.

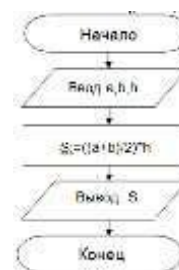


Рисунок 1. Блок-схема линейного алгоритма

Пример 2. Определить среднее арифметическое двух чисел, если a положительное и частное (a/b) в противном случае.

Запись решения задачи :

на алгоритмическом языке

Запись алгоритма в виде блок-схемы (рис. 2):

алг числа
вещ a,b,c
нач
ввод a,b
если $a > 0$
то $c := (a+b)/2$
иначе $c := a/b$
все
вывод c
кон

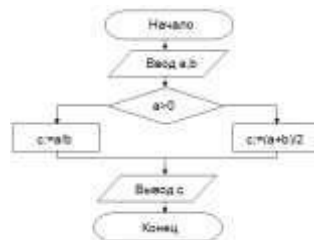


Рисунок 2. Блок-схема алгоритма с ветвлением

Пример 3. Составить алгоритм нахождения суммы целых чисел в диапазоне от 1 до 10.

Запись решения задачи
на алгоритмическом языке:

алг сумма
вещ a,s
нач
 $S := 0;$
 $a := 1;$
нц
пока $a \leq 10$
 $S := S + a;$
 $a := a + 1;$
кц
вывод S
кон

Запись алгоритма в виде блок-схемы (рис. 3):



Рисунок 3. Циклический алгоритм с предусловием

В алгоритме с **постусловием** сначала выполняется тело цикла, а затем проверяется условие окончания цикла. Решение задачи нахождения суммы первых десяти целых чисел в данном случае будет выглядеть следующим образом:

алг сумма
вещ a,s
нач
 $S := 0;$
 $a := 1;$
нц
 $S := S + a;$
 $a := a + 1;$
пока $a \leq 10$
кц
вывод S
кон

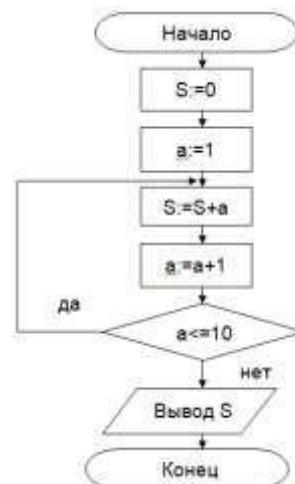


Рисунок 4. Циклический алгоритм с постусловием

Содержание работы:

Задание №1. Определить площадь прямоугольного треугольника со сторонами a и b.

Записать решение задачи на алгоритмическом языке и в виде блок-схемы.

Задание №2. Определить произведение двух чисел, если b положительное и частное (a/b) в противном случае.

Записать решение задачи на алгоритмическом языке и в виде блок-схемы.

Задание № 3. Сделайте вывод о проделанной работе

Контрольные вопросы:

1. Виды алгоритмов.
2. В чем состоит особенность описания алгоритмов с помощью структурной схемы и конструкций алгоритмического языка?
3. Отличительные особенности алгоритмов с предусловием и постусловием.

Литература

6. Информатика: учебник для начального и среднего профессионального образования. Цветкова Н.С., Великович Л.С. – Академия, 2017 г.
7. Информатика. Практикум для профессий и специальностей технического и социально-экономического профилей. Н. Е. Астафьева, С. А. Гаврилова, под ред. М.С. Цветковой, Академия, 2017 г.

Практическая работа №9

Среда программирования. Тестирование программы. Программная реализация несложного алгоритма

Цель занятия: изучить среду программирования и структуру программы Turbo Pascal, научиться составлять несложные программы в этой среде программирования, производить тестирование программы.

Оборудование, программное обеспечение: ПК, ОС Windows, среда программирования Turbo Pascal.

Методические рекомендации

Теоретическая часть

Программы на языке Паскаль имеют блочную структуру:

Блок типа PROGRAM - имеет имя, состоящее только из латинских букв и цифр. Его присутствие не обязательно, но рекомендуется записывать для быстрого распознавания нужной программы среди других листингов.

Программный блок, состоящий в общем случае из 7 разделов: раздел описания модулей (uses);

раздел описания меток (label); раздел описания констант (const);

раздел описания типов данных (type); раздел описания переменных (var); раздел описания процедур и функций; раздел описания операторов.

Общая структура программы на языке Паскаль: Program ИМЯ.; {заголовок программы}

Uses ...; {раздел описания модулей}

Var ...; {раздел объявления переменных}

...

Begin {начало исполнительной части программы}

... {последовательность операторов}

End. {конец программы}

Пример программы, которая осуществляет сложение двух чисел и выводит сумму на экран:

Program Summa; Uses

Crt; {Подключаем модуль Crt}

Var

number1, {переменная, в которой будет содержаться первое число} number2, {переменная, в которой будет содержаться второе число} rezult {переменная, в которой будет содержаться результат}

:integer; {указывает тип целых чисел} Begin

ClrScr; {Используем процедуру очистки экрана из модуля Crt} Write ('Введите первое число ');

{Выводим на экран символы, записанные между апострофами} Readln (number1);

{Введенное пользователем число считываем в переменную number1} Write ('Введите второе число ');

{Выводим на экран символы, записанные между апострофами} Readln (number2);

{Введенное пользователем число считываем в переменную number2} rezult := number1 + number2;

{Находим сумму введенных чисел и присваиваем переменной rezult} Write ('Сумма чисел ', number1, ' и ', number2, ' равно ', rezult);

{Выводим на экран строку, содержащую ответ задачи} Readln; {Процедура задержки экрана}

End.

Практическая часть

Задание 1. Изучите внешний вид системы программирования Турбо Паскаль.

Задание 2. Откройте файл, в который Вы запишите программу, выполняющую сложение двух чисел. Для этого нажмите клавишу F10, чтобы выйти в главное меню, затем клавишами перемещения курсора выберите опцию File, а в выпавшем меню команду New.

Найдите в этой программе заголовок, раздел описания переменных, признак начала программы, признак конца программы, тело программы, комментарий.

Ответьте на вопросы:

1. Какое назначение переменных number1, number2, result?
2. Что обозначает строка: number1, number2, result : integer; ?
3. Если присвоить переменным number1 и number2 соответственно значение 5 и 7, то какую строчку выдаст компьютер при исполнении последней процедуры Write? Запишите ее в отчет.
4. Переведите с английского языка слова: Write, Read. Как вы думаете, что должен делать операторы с таким названием?
5. Как вы понимаете запись: readln(number1); ?
6. Чему равно значение переменной result после выполнения оператора: result := number1 + number2; ?
7. Что делает оператор присваивания в этой программе?

Задание 3. Измените программу, выполненную в задании 2 так, чтобы она находила произведение двух чисел. Сохраните текстом программы в файле Proizv.pas. Результат покажите преподавателю.

Задание 4. Измените программу, выполненную в задании 3 так, чтобы она выполняла расчет площади прямоугольника по его длине и ширине. Заполните по результатам работы программы таблицу:

№	Параметры прямоугольника		
	длина	ширина	площадь
1			
2			

Контрольные вопросы

1. Что такое среда программирования?
2. Опишите среду программирования Turbo Pascal.
3. Как сохранить текст программы в Turbo Pascal?
4. Какова структура программы на языке Pascal?
5. Как осуществить запуск программы?

Практическая работа № 25-27

Выигрышные стратегии. Сложность описания. Кодирование с исправлением ошибок.

Сортировка

Цели занятия: Получить навыки кодирования и декодирования информации.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Теория кодирования – это раздел теории информации, изучающий способы отождествления сообщений с отражающими их сигналами. Задачей теории кодирования является согласование источника информации с каналом связи.

Объектом кодирования служит как дискретная, так и непрерывная информация, которая поступает к потребителю через источник информации. Понятие кодирования означает преобразование информации в форму, удобную для передачи по определенному каналу связи. Обратная операция – **декодирование** – заключается в восстановлении принятого сообщения из закодированного вида в общепринятый, доступный для потребителя.

В теории кодирования существует ряд направлений:

- статическое или эффективное кодирование;
- помехоустойчивое кодирование;
- корректирующие коды;
- циклические коды;
- арифметические коды.

С появлением управляющих систем, в частности ЭВМ, роль кодирования существенно возросла и изменилась, так как без кодирования невозможна передача информации. В последнее время в связи с развитием телекоммуникационных систем и широким использованием вычислительной техники для обработки и хранения информации возникла новая область знаний – информационная безопасность.

Кодированием называют универсальный способ отображения информации при ее хранении, обработке и передаче в виде системы соответствий между сигналами и элементами сообщений, при помощи которых эти элементы можно зафиксировать.

Код – это правило однозначного преобразования сообщения из одной символической формы представления сообщения в другую, обычно без каких-либо потерь информации.

Если все кодовые слова имеют одинаковую длину, то код называется равномерным, или блочным.

Под абстрактным алфавитом будем понимать упорядоченное дискретное множество символов.

Алфавитное кодирование. Алфавитное, т.е. побуквенное, кодирование можно задать таблицей кодов. Фактически кодом преобразования является некоторая подстановка.

$$\sigma = \begin{pmatrix} \alpha_1 & \dots & \alpha_n \\ \beta_1 & \dots & \beta_n \end{pmatrix}$$

Тогда $\sigma \in A$, где A – алфавиту A , $\beta_i \in B$ – множеству слов, составленных в алфавите B . Множество кодов букв называется множеством элементарных кодов. Алфавитное кодирование можно использовать для любого множества сообщений.

Компьютерная обработка данных основана на применении двоичного кода. Этот универсальный способ кодирования годится для любых данных, независимо от их происхождения и содержания.

Кодирование текста

Тексты – это последовательности символов, входящих некоторый алфавит. Кодирование текста сводится к двоичному кодированию алфавита, на основе которого он построен. Чаще всего применяется байтовое кодирование алфавита. В этом случае максимальная мощность алфавита составляет 256 символов. Такой алфавит может содержать два набора буквенных символов (например, русский и латинский), цифры, знаки препинания и математические знаки, пробел и небольшое число дополнительных символов. Примером такого алфавита является код **ASCII**.

Однако, ограниченный набор из 256 кодов символов сегодня уже не удовлетворяет возросшие потребности международного общения. Все большее распространение получает универсальная система 16-разрядного кодирования символов **UNICODE**.

Мощность алфавита в системе кодирования UNICODE составляет $2^{16}=65\,536$ разных кодов, из которых 63 484 кода соответствуют символам большинства алфавитов, а оставшиеся 2048 кодов разделены пополам и образуют таблицу размером 1024 столбцов x 1024 строк. В этой таблице более миллиона ячеек, в которых можно разместить еще более миллиона различных символов. Это символы «мертвых» языков, а также символы, не имеющие лексического содержания, указатели, знаки и т.п. Для записи этих дополнительных символов необходима пара 16-разрядных слов (16 разрядов для номера строки и 16 разрядов для номера столбца).

Таким образом, система UNICODE является универсальной системой кодирования всех символов национальных письменных систем и обладает возможностью существенного расширения.

Кодирование изображений

Рисунки, картинки, фотографии кодируются **в растровом формате**. В этом виде каждое изображение представляет собой прямоугольную таблицу, состоящую из цветowych точек. Цвет и яркость каждой отдельной точки выражаются в числовой форме, что позволяет использовать двоичный код для представления графических данных.

Черно-белые изображения принято представлять в градациях серого цвета, для этого используется модель **GreyScale**. Если яркость точки кодируется одним байтом, можно использовать 256 различных серых тонов. Такая точность согласуется с восприимчивостью человеческого глаза и возможностями полиграфической техники.

При кодировании цветных изображений применяют принцип декомпозиции цвета на составляющие, для этого используют модель **RGB**. Цветное изображение на экране получается путем смешивания трех базовых цветов: красного (Red, R), синего (Blue, B) и зеленого (Green, G).

Каждый пиксель на экране состоит из трех близко расположенных элементов, светящихся этими цветами.

Цветные дисплеи, использующие такой принцип называются RGB -мониторами.

Код цвета пикселя содержит информацию о доле каждого базового цвета.

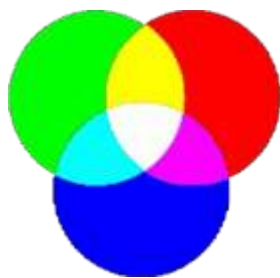


Схема цветообразования

Если все три составляющих имеют одинаковую интенсивность (яркость), то из их сочетаний можно получить 8 различных цветов:

R	G	B	цвет
0	0	0	черный
0	0	1	синий
0	1	0	зеленый
0	1	1	голубой
1	0	0	красный
1	0	1	розовый
1	1	0	коричневый
1	1	1	белый

Формирование цветов при глубине цвета 24 бита:

цвет	интенсивность R	интенсивность G	интенсивность B
черный	00000000	00000000	00000000
красный	11111111	00000000	00000000
зеленый	00000000	11111111	00000000
синий	00000000	00000000	11111111
голубой	00000000	11111111	11111111
желтый	11111111	11111111	00000000
белый	11111111	11111111	11111111

Чем больше глубина цвета, тем шире диапазон доступных цветов и тем точнее их представление в оцифрованном изображении. Пиксель с битовой глубиной, равной единице, имеет лишь 2 (в первой степени) возможных состояния — два цвета: черный или белый. Пиксель с битовой глубиной в 8 единиц имеет 28 или 256 возможных цветовых значений. Пиксель же с битовой глубиной в 24 единицы имеет 224 степени) или 16,7 миллионов возможных значений. Считается, что 24-битные изображения, содержащие 16,7 миллионов цветов, достаточно точно передают краски окружающего нас мира. Как правило, битовое разрешение задается в диапазоне от 1 до 48 бит/пиксель.

При печати на бумаге используется несколько иная цветовая модель: если монитор испускал свет, оттенок получался в результате сложения цветов, то краски - поглощают свет, цвета вычитаются. Поэтому в качестве основных используют голубую (Cyan, C), пурпурную (Magenta, M) и желтую (Yellow, Y) краски. Кроме того, из-за не идеальности красителей, к ним обычно добавляют четвертую -- черную (black, K). Для хранения информации о каждой краске и в этом случае чаще всего используется 1 байт. Такая система кодирования носит название **СМУК**.

Более грубое представление цвета использует меньшее число разрядов. Например, кодирование цветной графики 16-разрядными числами носит название **High Color**. В этом случае каждому цвету отводят пять разрядов.

Кодирование звука и видео

Приемы работы со звуковой информацией пришли в компьютерную технику позже всего. Аналитический метод кодирования, применимый к любым звуковым сигналам основан на аналогово-цифровом преобразовании. Исходный аналоговый сигнал представляют как последовательность цифровых сигналов, записанных в двоичном коде. Разрядность преобразования определяет объем данных, соответствующих отдельному цифровому сигналу. При воспроизведении звука выполняют обратное цифро-аналоговое преобразование. Этот метод кодирования содержит погрешность, так что воспроизводимый сигнал несколько отличается от оригинала.

Метод кодирования на основе табличного синтеза применим только к музыкальным произведениям. В заранее подготовленных таблицах хранятся образцы (**сэмплы**) звуков различных музыкальных инструментов. Числовые коды определяют инструмент, ноту и продолжительность звучания.

При кодировании видеосигнала требуется записать последовательность изображений (кадров) и звук (звуковая дорожка). Формат видеозаписи позволяет включить оба потока данных в одну цифровую последовательность.

ПРИМЕР ВЫПОЛНЕНИЯ ЗАДАНИЙ

1. Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "Я УМЕЮ КОДИРОВАТЬ ИНФОРМАЦИЮ".

Решение: Для того, чтобы легко закодировать фразу, заменяя каждую букву ее порядковым номером в алфавите необходимо составить таблицу соответствия "Буква - её порядковый номер":

№	Буква	№	Буква	№	Буква	№	Буква	№	Буква
1	А	8	Ж	15	Н	22	Ф	29	Ы

2 Б	9 З	16 О	23 Х	30 Ъ
3 В	10 И	17 П	24 Ц	31 Э
4 Г	11 Й	18 Р	25 Ч	32 Ю
5 Д	12 К	19 С	26 Ш	33 Я
6 Е	13 Л	20 Т	27 Щ	- расстояние между буквами
7 Ё	14 М	21 У	28 Ь	- пробел

Теперь можно легко зашифровать фразу. Она будет выглядеть так: "33 21-14-6-32 12-16-5-10-18-16-3-1-20-30 10-15-22-16-18-14-1-24-10-32"

2. Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца):

0	1	2	3	4	5	6	7	8
0 А	Б	В	Г	Д	Е	Ж	З	И
1 Й	К	Л	М	Н	О	П	Р	С
2 Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
3 Ы	Ь	Й	Э	Ю	Я	_	.	,
4 :	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: ТЕЛЕГРАММА. Решение: 20 05 12 05 03 17 00 13 13 00

ВАРИАНТ 1

Дана кодовая таблица азбуки Морзе

А	• —	Л	• — • •	Ц	— • — •
Б	— • • •	М	— —	Ч	— — — •
В	• — —	Н	— •	Ш	— — — —
Г	— — •	О	— — —	Щ	— — • —
Д	— • •	П	• — — •	Ъ	• — — • — •
Е	•	Р	• — •	Ы	— • — —
Ж	• • • —	С	• • •	Ь	— • • —
З	— — • •	Т	—	Э	• • — • •
И	• •	У	• • —	Ю	• • — —
Й	• — — —	Ф	• • — •	Я	• — • —
К	— • —	Х	• • • •		

Расшифруйте (декодируйте), что здесь написано (буквы отделены друг от друга пробелами): — — — — • — • • — — — — • • — — • • — —

Закодируйте с помощью азбуки Морзе слова: информатика, данные, алгоритм.

Закодируйте с помощью азбуки Морзе свое имя и фамилию.

Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца)

	0	1	2	3	4	5	6	7	8
0	А	Б	В	Г	Д	Е	Ж	З	И
1	Й	К	Л	М	Н	О	П	Р	С
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
3	Ы	Ь	Э	Ю	Я	_	.	,	?
4	:	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: Я УМЕЮ РАБОТАТЬ С ИНФОРМАЦИЕЙ! А ТЫ?

Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "Я УМЕЮ КОДИРОВАТЬ ИНФОРМАЦИЮ". Что необходимо предусмотреть, чтобы зашифрованный текст был записан без пропусков?

ВАРИАНТ 2

Дана кодовая таблица азбуки Морзе

А	• —	Л	• — • •	Ц	— • — •
Б	— • • •	М	— —	Ч	— — — •
В	• — —	Н	— •	Ш	— — — —
Г	— — •	О	— — —	Щ	— — • —
Д	— • •	П	• — — •	Ъ	• — — • — •
Е	•	Р	• — •	Ы	— • — —
Ж	• • • —	С	• • •	Ь	— • • —
З	— — • •	Т	—	Э	• • — • •

И	• •	У	• • –	Ю	• • – –
Й	• – – –	Ф	• • – •	Я	• – • –
К	– • –	Х	• • • •		

Расшифруйте (декодируйте), что здесь написано (буквы отделены друг от друга пробелами): - • - • • • - - - - -

2

Закодируйте с помощью азбуки Морзе слова: информатика, данные, алгоритм.

3

Закодируйте с помощью азбуки Морзе свое имя и фамилию.

Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца)

	0	1	2	3	4	5	6	7	8
0	А	Б	В	Г	Д	Е	Ж	З	И
1	Й	К	Л	М	Н	О	П	Р	С
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
3	Ы	Ь	Э	Ю	Я	–	•	,	?
4	:	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: "Таблицы маршрутизации - электронные базы данных".

Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "Скрипт - набор команд какого-либо языка". Что необходимо предусмотреть, чтобы зашифрованный текст был записан без пропусков?

ВАРИАНТ 3

1

Дана кодовая таблица азбуки Морзе

А	• –	Л	• – • •	Ц	– • – •
Б	– • • •	М	– –	Ч	– – – •
В	• – –	Н	– •	Ш	– – – –
Г	– – •	О	– – –	Щ	– – • –
Д	– • •	П	• – – •	Ъ	• – – • – •
Е	•	Р	• • •	Ы	– • – –
Ж	• • • –	С	• • •	Ь	– • • –
З	– – • •	Т	–	Э	• • – • •
И	• •	У	• • –	Ю	• • – –
Й	• – – –	Ф	• • • •	Я	• – • –
К	– • –	Х	• • • •		

Расшифруйте (декодируйте), что здесь написано (буквы отделены друг от друга пробелами): • - - • • • - - - • • -

2

Закодируйте с помощью азбуки Морзе слова: флаг, имя, кодирование.

3

Закодируйте с помощью азбуки Морзе свое имя и фамилию.

Дана кодировочная таблица (первая цифра кода - номер строки, вторая - номер столбца)

	0	1	2	3	4	5	6	7	8
0	А	Б	В	Г	Д	Е	Ж	З	И

1	Й	К	Л	М	Н	О	П	Р	С
2	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
3	Ы	Ь	Э	Ю	Я	_	.	,	?
4	:	;	-	!	"				

С помощью этой кодировочной таблицы зашифруйте фразу: "УЧЕНИЕ ДА ТРУД К СЛАВЕ ВЕДУТ"

Заменяя каждую букву ее порядковым номером в алфавите, зашифруйте фразу: "ЮЗЕР - ПОЛЬЗОВАТЕЛЬ КОМПЬЮТЕРА". Что необходимо предусмотреть, чтобы зашифрованный текст был записан без пропусков?

Практическая работа № 28-30

Элементы теории алгоритмов. Вычислимость

Цели занятия: Закрепление общего понятия «алгоритм». Решение простейших алгоритмических задач.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Практическая работа №1

Тема: ПР 1-2 Словесная форма представления алгоритма.

Цель: *Закрепление общего понятия «алгоритм». Решение простейших алгоритмических задач*

Задание 1. Определить площадь трапеции по введенным значениям оснований (a и b) и высоты (h).

Задание 2. Определить среднее арифметическое двух чисел, если a положительное и частное (a/b) в противном случае.

Задание 3. Составить алгоритм нахождения суммы целых чисел в диапазоне от 1 до 10.

Практическая работа №2

Тема: ПР 3-4 Графическая форма представления алгоритма

Цель: Получение практических навыков в работе с графической формой представления алгоритма

1. Линейный алгоритм, выполненный в практической работе №1, переделать на циклический с заданным количеством циклов по примеру 4.
2. Найти сумму значений переменной P , полагая, что начальное значение этой переменной равно нулю, т.е. $P=0$. В каждом цикле переменная изменяется на 2, т.е. $P=P+2$. Количество циклов равно 5. В результате данного алгоритма значение переменной будет равно $P=10$.
3. Пусть заданы начальные значения переменных:
 $x:=1; y:=5$.
Начало цикла;
пока $y>x$
 $y:=y-x$;
конец цикла.
Определить количество циклов и значения переменных x, y после выхода из цикла.

Практическая работа №3

Тема: 5-6 Представление алгоритма на псевдокоде.

Цель: Получение практических навыков в работе с алгоритмом, написанном на псевдокоде.

Составить алгоритм решения задачи с помощью алгоритмического языка псевдокод и с помощью блок-схем, используя конструкцию линейного алгоритма.

1. Вычислить площадь поверхности и объем усеченного конуса по следующим формулам

$$S = \pi (R + r) l + \pi R^2 + \pi r^2 ;$$

$$V = (1/3) \pi (R^2 + r^2 + Rr) h .$$

2. Вычислить координаты центра тяжести трех материальных точек с массами m_1, m_2, m_3 и координатами $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ по формулам:

$$x_c = (m_1 x_1 + m_2 x_2 + m_3 x_3) / (m_1 + m_2 + m_3) ;$$

$$y_c = (m_1 y_1 + m_2 y_2 + m_3 y_3) / (m_1 + m_2 + m_3) .$$

3. Вычислить площадь треугольника со сторонами a, b, c по формуле

Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)} ,$$

где p – полупериметр, вычисляемый по формуле

$$\frac{a+b+c}{2} ,$$

4. Вычислить координаты точки, делящей отрезок $a_1 a_2$ в отношении $n_1:n_2$ по формулам:

$$x = \frac{x_1 + \gamma x_2}{1 + \gamma} ;$$

$$y = \frac{y_1 + \gamma y_2}{1 + \gamma} ,$$

где $\gamma = \frac{n_1}{n_2}$.

5. Вычислить медианы треугольника со сторонами a, b, c по формулам:

$$m_a = 0.5 \sqrt{2b^2 + 2c^2 - a^2} ;$$

$$m_b = 0.5 \sqrt{2a^2 + 2c^2 - b^2} ;$$

$$m_c = 0.5 \sqrt{2b^2 + 2a^2 - c^2} ;$$

6. Вычислить площадь круга и длину окружности по введенному значению радиуса.

7. Вычислить площадь S и периметр L эллипса по введенным значениям полуосей a и b :

$$S = \pi \cdot a \cdot b ;$$

$$\frac{1}{2(a^2 + b^2)}$$

Практическая работа №4

Тема: ПР7-8 Составление программ для машины Тьюринга.

Цель: Получение практических навыков в написании программ для «Машины Тьюринга»

Задача 1.1. $A = \{a, b\}$. Заменить во входном слове все символы “a” на “b”.

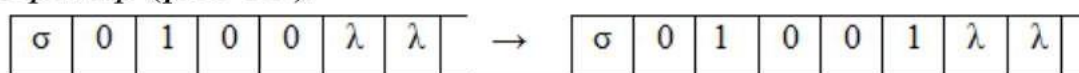
Пример (рис. 1.2):



Рис. 1.2. Пример замены символов

Задача 1.2. $A = \{0, 1\}$. Записать в конце слова 1. Если слово пустое – вывести сообщение об ошибке (отметив заключительное состояние как Ω^1).

Пример (рис. 1.3):



Задача 1.3. $A = \{0, 1\}$. После слова записать первую букву слова. Если слово пустое – вывести сообщение об ошибке (отметив заключительное состояние как Ω^1).

Пример (рис. 1.4):

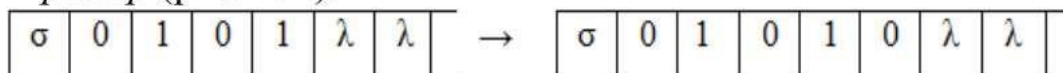


Рис. 1.4. Пример записи после слова первого символа

Задача 1.5. $A = \{0, 1, 2\}$. Во входном слове заменить все комбинации “012” на звездочки.

Пример (рис. 1.6):

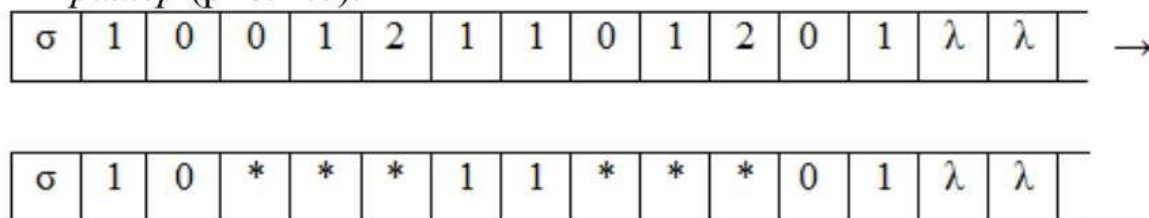


Рис. 1.6. Пример замены комбинации символов

Практическая работа №5

Тема: ПР9-10 Составление программ для машины Поста.

Цель: Получение практических навыков в написании программ для «Машины Поста»

Задача 1.15. $A = \{0, 1\}$. Машина двухленточная. Дано слово, оно записано на первой ленте. Скопировать это слово на вторую ленту в обратном порядке.

Задача 1.16. $A = \{0, 1\}$. Машина трехленточная. Даны два числа в двоичном коде, они записаны на первой и второй ленте соответственно. Считается, что слова записаны нормально: т.е. оба слова всегда есть и начинаются они с единицы (исключение только для числа ноль). Определить, чему равна сумма этих двух чисел, и результат записать на третью ленту.

Задача 1.17. $A = \{1\}$. Машина трехленточная. Даны два числа в унарном коде, они записаны на первой и второй ленте соответственно. Определить, чему равно произведение этих двух чисел и результат записать на третью ленту.

Практическая работа №6

Тема: ПР 11-12 Составление нормальных алгоритмов Маркова.

Цель: Получение практических навыков в составлении алгоритмов Маркова.

Задача 1.18. $A = \{0, 1\}$. Заменить произвольное слово на символ «0».

Задача 1.19. $A = \{0, 1\}$. Удвоить все символы в слове.

Решение. Предлагается следующий алгоритм: найти начало слова, затем удваивать все символы.

Задача 1.20. $A = \{0, 1\}$. Удалить каждый третий символ.

Решение. Предлагается следующий алгоритм: найти начало слова, затем отсчитывать каждый третий символ, меняя вид челнока.

Задача 1.21. $A = \{0, 1\}$. Поставить в конце слова «+», если в нем есть хотя бы один ноль, поставить «-», если это не так.

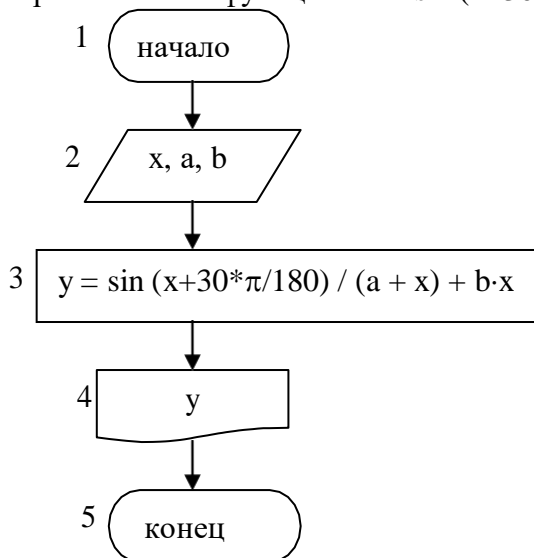
Задача 1.22. $A = \{0, 1\}$. Переработать слово в «+», если в нем есть хотя бы два нуля (неважно расположены они рядом или нет), и в «-», если это не так (т.е. исходное слово удаляется).

Практическая работа №7

Тема: ПР 13-14 Решение задач по составлению линейных алгоритмов.

Цель: Получение практических навыков при решении задач для составления линейных алгоритмов

1. Вычислить и вывести на экран значение функции: $Y = \sin(x+30^\circ)/(a+x)+b \cdot a$.



2. Вычислить медианы треугольника со сторонами a, b, c по формулам:

$$m_a = 0.5\sqrt{2b^2 + 2c^2 - a^2} ;$$

$$m_b = 0.5\sqrt{2a^2 + 2c^2 - b^2} ;$$

$$m_c = 0.5\sqrt{2b^2 + 2a^2 - c^2} ;$$

3. Вычислить площадь круга и длину окружности по введенному значению радиуса.

4. Вычислить площадь S и периметр L эллипса по введенным значениям полуосей a и b :

$$S = \pi \cdot a \cdot b ;$$

$$L = \frac{1}{2} \pi (a^2 + b^2) .$$

5. Вычислить объем V и площадь боковой поверхности цилиндра S по введенным значениям радиуса основания R и высоты цилиндра H .

$$V = \pi \cdot R^2 \cdot H ;$$

$$S = 2 \cdot \pi \cdot R \cdot H .$$

6. Вычислить объем V и площадь боковой поверхности конуса S по введенным значениям радиуса основания r , высоты h и образующей l :

$$V = \frac{1}{3} \pi \cdot r^2 \cdot h ;$$

$$S = \pi \cdot r \cdot l .$$

7. Вычислить объем V и площадь поверхности S сферы по введенному значению радиуса r :

$$V = \frac{4}{3} \cdot \pi \cdot r^3 ;$$

$$S = 4 \cdot \pi \cdot r^2 .$$

Практическая работа №8

Тема: ПР 15-16 Решение задач по составлению ветвящихся алгоритмов.

Цель: Получение практических навыков при решении задач для составления ветвящихся алгоритмов

1. Составить программу для решения квадратного уравнения $ax^2 + bx + c = 0$.
2. Определить максимальное четное число из двух введенных.
3. Определить, можно ли из отрезков с длинами x , y и z построить треугольник.
4. Ввести два числа a и b . Большее число заменить утроенным произведением, меньшее – полусуммой.
5. Если среди трех чисел a , b , c имеется хотя бы одно четное, то найти максимальное число, иначе – минимальное.
6. Определить, в каком квадранте находится точка с координатами x и y и вывести номер квадранта на экран.
7. Найти квадрат наибольшего из двух чисел a и b . Вывести на экран число 1, если наибольшим является число a , число 2 – если наибольшим числом является b .
8. Определить, попадает ли точка с координатами x и y в круг радиусом R . Если точка попадает в круг, вывести на экран единицу, в противном случае – ноль.
9. Написать алгоритм решения задачи, которая решает уравнение $ax + b = 0$ относительно x для любых чисел a и b , введенных с клавиатуры. Все числа считаются действительными.
10. Написать алгоритм решения задачи, которая определяет, лежит ли точка $A(x,y)$ внутри некоторого кольца («внутри» понимается в строгом смысле, т.е. случай, когда точка A лежит на границе кольца, недопустим). Центр кольца находится в начале координат. Для кольца заданы внутренний и внешний радиусы $r1$, $r2$. Координаты x и y вводятся с клавиатуры.

Практическая работа №9

Тема: ПР 17-18 Решение задач по составлению циклических алгоритмов.

Цель: Получение практических навыков при решении задач для составления циклических алгоритмов

1. Найти сумму чисел, кратных трем, в диапазоне от 0 до 50.
2. Найти сумму первых десяти чисел, кратных пяти.
3. Найти произведение четных чисел в диапазоне от 2 до 30.
4. Вводятся положительные числа. Прекратить ввод, когда сумма введенных чисел превысит 100.
5. Требуется найти сумму чисел, кратных 7, в диапазоне от 0 до 100. Вывести на экран сумму чисел и их количество.
6. Определить количество целых чисел, кратных 3 (от 3 и далее), дающих в сумме число, превышающее 200.
7. Вводятся 10 чисел. Вывести на экран суммы положительных и отрицательных чисел и их количество.
8. Вывести на экран значения функции $y=\sin(x)$ для $0 \leq x \leq 180$ с шагом в 1° .
9. Подсчитать площади десяти кругов с радиусами от 1 см с шагом 2 см и вывести значения площадей на экран.
10. Вводятся положительные числа. Прекратить ввод чисел, когда их сумма превысит 100. Результат вывести на экран.

Практическая работа №10

Тема: ПР 19-20 Составление алгоритма поиска в неупорядоченном массиве.

Цель: *Получение практических навыков в работе с алгоритмами поиска в неупорядоченном массиве*

1. Определить порядковые номера нулей в массиве.
2. Все отрицательные элементы массива заменить нулями. Выдать сообщение о количестве таких замен.
3. Найти порядковый номер минимального числа в массиве.
4. Найти значение и координаты первого по порядку положительного элемента в массиве.
5. Найти количество единиц в массиве.
6. Каких чисел в массиве больше: положительных или отрицательных? (массив может содержать нулевые значения).
7. Исходные данные: В таблице 10 строк. Первый столбец – фамилия работника, второй – стаж работника.
8. По стажу работника определить, к какой группе он относится, если 1 группа – стаж до 3-х лет, 2 группа – до 5 лет, 3 группа – до 10 лет, 4 группа – свыше 10 лет.
9. Найти сумму диагональных элементов блока ячеек, состоящего из 5 строк и 5 столбцов.
10. Двадцати переменным массива присвоить одно и то же значение, равное наибольшему из них.

Практическая работа № 31-33

Эквивалентность алгоритмических моделей

Цели занятия: - сформировать представление об алгоритме и его свойствах;

- сформировать представление о способах их описания алгоритмов;
- сформировать представление о типах алгоритмов;
- сформировать представление об основных алгоритмических конструкциях.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Слово *алгоритм* происходит от латинской формы написания имени великого математика IX века Аль-Хорезми, который сформулировал правила выполнения арифметических действий.

Первоначально под алгоритмами понимали только правила выполнения четырёх арифметических действий над многозначными числами.

Алгоритм – организованная последовательность действий.

Алгоритмизация -

Свойства алгоритма:

- Дискретность
- Определённость
- Результативность
- Массовость
- Конечность

Способы записи алгоритмов (блок-схема)

Алгоритмы могут быть заданы:

словесно

таблично

графически

Словесное задание описывает алгоритм с помощью слов и предложений естественного языка.

Табличное задание служит для представления алгоритма в форме таблиц и расчётных формул.

Графическое задание или *блок-схема* – способ представления алгоритма с помощью геометрических фигур, называемых *блоками*.

Правила изображения блок-схем



Типы алгоритмов

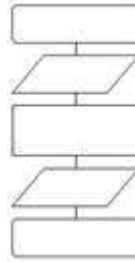
Алгоритмы бывают:

линейные

разветвляющиеся

циклические

Алгоритм, в котором команды выполняются последовательно одна за другой, называется *линейным алгоритмом*.



В *разветвляющиеся алгоритмы* входит условие, в зависимости от выполнения или невыполнения которого выполняется та или иная последовательность команд (серий). В алгоритмической структуре «*ветвление*» та или иная серия команд выполняется в зависимости от истинности *условия*. *Условие* может быть либо истинным, либо ложным.

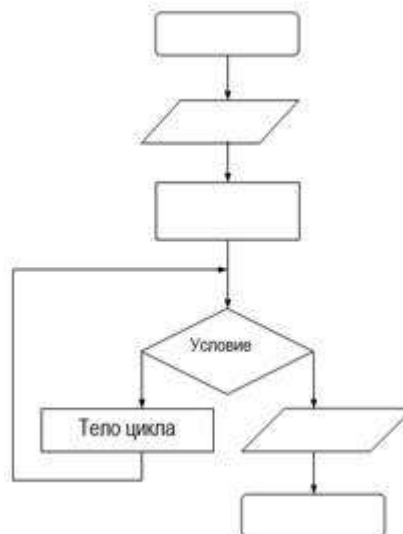


В *циклические алгоритмы* входит последовательность команд, выполняемая многократно. Такая последовательность команд называется *телом цикла*. В алгоритмической структуре «*цикл*» серия команд (тело цикла) выполняется многократно. Циклические алгоритмические структуры бывают двух типов:

- *циклы со счётчиком*, в которых тело цикла выполняется определённое количество раз;



- *циклы с условием*, в которых тело цикла выполняется, пока условие истинно.



Практическое занятие

Задание 1. составить в виде блок-схемы алгоритм нахождения середины отрезка при помощи циркуля и линейки опираясь на пример алгоритма естественного языка

Дан отрезок АВ.

Пример: «Алгоритм деления отрезка АВ пополам».

1. поставить ножку циркуля в т.А;
2. установить раствор циркуля равным длине отрезка АВ;
3. провести окружность;
4. поставить ножку циркуля в т.В;
5. провести окружность;
6. через точки пересечения окружностей провести прямую;
7. отметить точку пересечения этой прямой с отрезком АВ.

Задание 2. Используйте ресурсы Интернета для нахождения определения свойств алгоритма и запишите их в тетрадь. Приведите примеры.

Задание 3. Допишите в тетради в основные алгоритмические конструкции недостающие правила блок-схем.

Задание 4. Сделать вывод о проделанной практической работе

Практическая работа № 34-36

Построение алгоритмов и практические вычисления

Цели занятия: Повторить и закрепить с учащимися понятия алгоритма, исполнителя, повторить свойства и способы описания алгоритмов; алгоритмическим способом решения задач, с основными алгоритмическими структурами; закрепить полученные знания с помощью электронного теста.

Формировать навыки реализации теоретических знаний в практической деятельности; формировать такие мыслительные операции как анализ, синтез, сравнение, обобщение; формировать умение работать в группе; паре.

Воспитывать чувство ответственности за качество и результат выполняемой работы; воспитывать стремление к совершенствованию знаний.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

1. Новая модель робота-собаки «Айбо»

Японская компания Sony создала новую модель (второго поколения) робота-собаки «Айбо». Эта игрушка теперь умеет не только бегать, прыгать и вилять хвостом (что было реализовано в первой модели), но и выполнять 50 различных поданных голосом команд хозяина, а изображение с видеокамер, заменяющих ей глаза, при желании владельца транслируется на экран монитора компьютера. Цена такой игрушки – около 1,5 тысячи долларов.

2. В японском театре играют роботы

На сцене университетского театра в Осаке поставлен первый экспериментальный спектакль, в котором на сцену выходят люди и роботы. Действие пьесы под названием "Я, работник" происходит в недалеком будущем. Драматург Оризо Хирата говорит, что его пьеса посвящена проблемам будущих взаимоотношений между людьми и искусственным разумом. Спектакль длится всего 20 минут. Робот, известный под именем Wakamaru, дебютировал на сцене театра. Программу, необходимую для выступлений на сцене, в течение двух месяцев написали программисты университета.

3. Самый маленький робот

Японской компанией Тому Со создан уникальный гуманоид — самый маленький на сегодняшний день человекообразный робот, чей рост достигает всего трех сантиметров. Микроробот, названный «Robo-Q» оснащен искусственным интеллектом, который позволяет ему ориентироваться в пространстве и играть в различные игры. Так, гуманоид может играть в футбол, находить путь в лабиринтах и ловко обходить внезапно появляющиеся на его пути препятствия, используя встроенные инфракрасные сенсоры.

- Скажите мне, ребята, а кем является эти роботы?

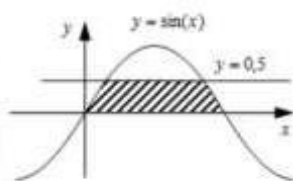
Ответ: *исполнителями. А исполняют они алгоритмы, записанные на языках программирования.*

А вот сейчас вы постараетесь проверить себя, как вы усвоили тему и кому нужна дополнительная работа.

1. **Компьютерное тестирование (Тест-теория).** Записать номера неверных ответов для самокоррекции.
2. **Анализ выполнения теоретической части.**
3. **Физкультминутка: упражнения под музыкально-звуковое сопровождение.**
(Обратить внимание на задание у доски).
4. **Компьютерное тестирование (Тест-выполнение алгоритмов).**
5. **Анализ выполнения практической части.**
6. **Работа в паре.**

Задайте с помощью предиката $P(x)$ множество точек, соответствующее заштрихованной области на плоскости.

1. $(y \leq \sin(x)) \text{ or } (y \geq 0) \text{ and } (y \leq 0.5)$
2. $(y \leq \sin(x)) \text{ and } (y \geq 0) \text{ and } (y \leq 0.5) \text{ and } (x \geq 0) \text{ and } (x \leq 4)$
3. $(y \leq \sin(x)) \text{ and } (y \geq 0) \text{ and } (y \leq 0.5) \text{ and } (x \geq 0)$
4. $(y \leq \sin(x)) \text{ and } (y \geq 0 \text{ and } (y \leq 0.5))$



7. **Отчет работы группы (мини - проект).**
8. **Подведение итогов урока.**
9. **Домашнее задание.**
10. **Рефлексия.**

Задание для работы в паре:

Составить алгоритм определения подходящего возраста кандидатуры для вступления в брак, используя следующее соображение: возраст девушки равен половине возраста мужчины +7, возраст мужчины определяется соответственно как удвоенный возраст минус 14. Данные для проверки выбирать самостоятельно.

Домашнее задание:

1. **Составить блок-схемы к пословицам:** Сделал дело – гуляй смело; Что посеешь, то и пожнешь; Где песня поется, там весело живет. Вертится, как белка в колесе.
2. **Вычислить значение** гипотенузы прямоугольного треугольника, если известны значения его катетов.
3. **Работа в группе:**

А) Задание первой группе.

Для производства вакцины на заводе планируется выращивать культуру бактерий. Известно, что если масса бактерий – x г, то через день она увеличится на $(a-bx)x$ г, где коэффициенты a и b зависят от вида бактерий. Завод ежедневно будет забирать для нужд производства m г бактерий. Для составления плана важно знать ответ на вопрос: как изменяется масса бактерий через 1, 2, 3, ..., 365 дней. Ответьте, на этот вопрос. (Составьте алгоритм).

Б) Задание второй группе.

Расположенный на берегу реки металлургический завод осуществил сброс сточных вод, в результате чего концентрация вредных веществ в реке резко увеличилась. С течением

времени эта концентрация, естественно, уменьшается. Требуется сообщить, каков будет уровень загрязнения реки через сутки, двое суток и т.д., до тех пор, пока концентрация не станет меньше предельно допустимой.

В) Задание 3 группе.

Задача о дорожно - транспортном происшествии.

На узкой улице внезапно заклинило тормоза у «Лады Калины». В результате немедленно образовалась «пробка». Шофер стоящего сзади грузовика, у которого лопнуло терпение, предложил помочь убрать «Ладу Калину» с проезжей части дороги, оттащив ее на обочину с помощью троса. Удается ли оттащить «Ладу Калину»?

Задание № ? (Придумать интересное практическое задание, которое можно решить, составив программу с циклом).

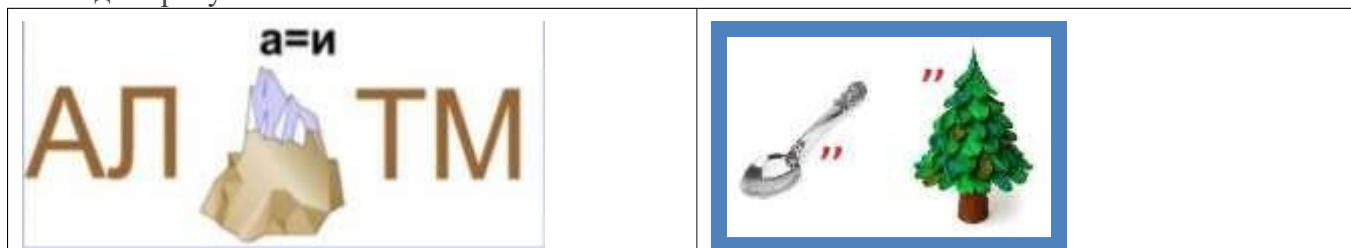
Дополнительное задание.

1.

Мощность радиоактивного излучения пропорциональна концентрации активного вещества. Период полураспада одного из изотопов углерода составляет 8 дней. В начальный момент времени мощность излучения составляет 2 рентгена в час. Определить, через сколько дне мощность излучения снизится до безопасной для человека величины - 0.15 рентгена в час. Изменение концентрации, а следовательно, мощности излучения описывается формулой

$Q = Q_0 e^{-\lambda t}$, где Q_0 -начальная мощность $\lambda = \ln 2 / p$, где p -период полураспада. (Для решения задачи изменяйте t с заданным шагом (например 5 дней). Вычислить Q и сравнить его с допустимой величиной. Число повторений цикла заранее не известно.

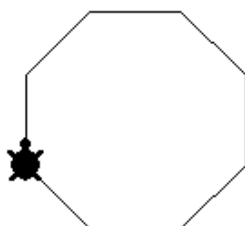
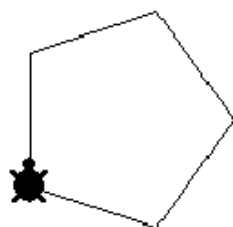
Разгадать ребус

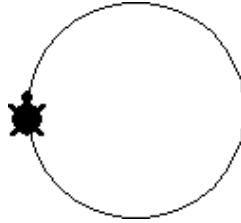


Дополнительно

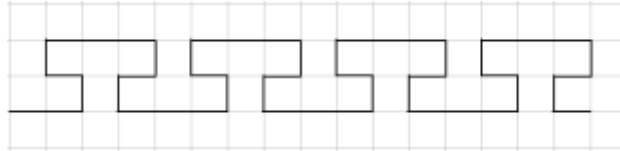
Построение циклического алгоритма для управления учебным исполнителем

1. Напишите программу на языке ЛОГО для изображения правильных пятиугольника, восьмиугольника, 36-угольника:

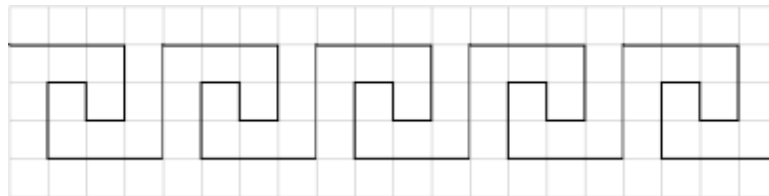




2. Напишите программу на языке ЛОГО для изображения рисунка:



3. Напишите программу на языке ЛОГО для изображения рисунка:



Практическая работа № 37-39

Основные конструкции языка программирования

Цели занятия: ввести понятия алгоритма, его свойств; изучить основные алгоритмические конструкции и их описание средствами языков программирования.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Понятие алгоритма

Алгоритм является фундаментальным понятием информатики. Представление о нем необходимо для эффективного применения вычислительной техники к решению практических задач.

Алгоритм - это последовательность действий, которая должна быть выполнена для достижения желаемого результата.

Алгоритм решения некоторой задачи - это алгоритм, приводящий к решению этой задачи за конечное число действий

Свойства алгоритма и его исполнители

1. **Дискретность.**

Разделение алгоритма на последовательность законченных действий – шагов. Каждое действие должно быть закончено прежде, чем исполнитель приступит к выполнению следующего шага.

2. **Результативность.**

Получение из исходных данных результата за конечное число шагов.

3. **Массовость.**

Возможность применения алгоритма к большому количеству различных исходных данных.

4. **Детерминированность.**

Выполнение команд алгоритма в строго определенной последовательности.

5. **Выполнимость и понятность.**

Алгоритм не должен содержать предписаний, смысл которых может восприниматься неоднозначно.

6. **Точность.**

Запись алгоритма должна быть такой, чтобы на каждом шаге его выполнения было известно, какую команду нужно выполнять следующей.

7. **Конечность.**

Завершение работы алгоритма за конечное число шагов.

Способы описания алгоритма

Найти сумму двух чисел 2 и 3

1. **Словесный способ**

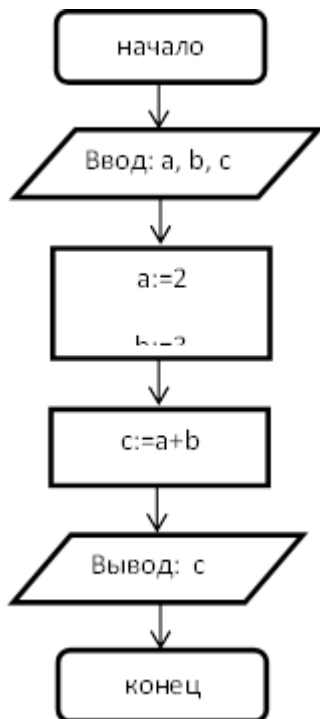
Алгоритм представляет собой описание на естественном языке последовательных этапов обработки данных.

К двум прибавляем три получаем пять.

2. **Графический способ**

Изображение алгоритма в виде последовательности связанных между собой функциональных блоков.

Блок-схема позволяет сделать алгоритм более наглядным и выделяет в алгоритме основные алгоритмические структуры (линейная, ветвление, выбор и цикл).



Элемент блок-схемы	Назначение элемента блок-схемы
	Прямоугольник с закругленными углами, применяется для обозначения начала или конца алгоритма
	Параллелограмм, предназначен для описания ввода или вывода данных, имеет один вход сверху и один выход внизу
	Прямоугольник, применяется для описания линейной последовательности команд, имеет один вход сверху и один выход внизу
	Ромб, служит для обозначения условий в алгоритмических структурах «ветвление» и «выбор», имеет один вход сверху и два выхода (налево, если условие выполняется, и направо, если условие не выполняется)

3. Псевдокод

Система обозначений и правил, предназначенная для единообразной записи алгоритмов.

Алг Сумма

дано a, b, c;

надо c=a+b;

нач a:=2, b:=3;

c:= a+b;

кон.

4. Программный способ (алгоритмический)

Алгоритм, предназначенный для записи на компьютере, должен быть записан на понятном ему языке. Такой язык называется **языком программирования**, а запись алгоритма на этом языке – **программа**. Языки программирования предназначены для создания программ, которые могут быть исполнены ЭВМ или другими автоматическими устройствами, например, станками с числовым программным управлением. Система Pascal ABC предназначена для обучения программированию на языке Паскаль. Как и любой алгоритм, являющийся последовательностью инструкций, программа на языке Паскаль состоит из команд (операторов), записанных в определенном порядке и формате. Команды позволяют получать, сохранять и обрабатывать данные различных типов (например, целые числа, символы, строки символов, т.д.).

Кроме команд в записи программы участвуют еще так называемые "служебные слова", организующие структуру программы. Правила языка Паскаль предусматривают единую для всех программ форму основной структуры

Program <Имя программы>;

<Раздел описаний>;

Begin

<Тело программы>;

End.

Здесь слова *Program*, *Begin* и *End* являются служебными. Правильное и уместное употребление этих слов является обязательным.

Пример. Ввести в компьютер два целых числа, найти их сумму, результат вывести на эк-ран с поясняющим текстом.

Внимание! Две косые черты (//) отделяют комментарии, их набирать не нужно.

program raschet;// название программы

uses crt;// подключаемые модули

```
var x, y, s:integer;// объявление имен переменных и их типа
begin// начало исполнительской части
writeln('Введите два целых числа');//написать на экране текст
readln(x,y);//прочитать данные с клавиатуры и запомнить их в переменных
s:=x+y; // выполнить расчет и запомнить его в переменной
writeln('Сумма чисел =',s); //написать на экране текст и значение переменной
end. //конец программы
```

Задания к работе.

Задание 1. Некий злоумышленник выдал следующий алгоритм за алгоритм получения кипятка:

1. Налить в чайник воду.
2. Открыть кран газовой горелки.
3. Поставить чайник на плиту.
4. Ждать, пока не закипит вода.
5. Поднести спичку к горелке.
6. Зажечь спичку.
7. Выключить газ.

Исправьте алгоритм, чтобы предотвратить несчастный случай.

Задание 2.

Имеются два кувшина емкостью 3 л и 8 л. Напишите алгоритм на естественном языке, выполняя который можно набрать из реки 7 л воды. (Разрешается пользоваться только этими кувшинами.)

Задание 3.

Перед выходным днем папа сказал своему сыну: «Давай спланируем свой завтрашний день. Если будет хорошая погода, то проведем день в лесу. Если же погода будет плохая, то сначала займемся уборкой квартиры, а во второй половине дня сходим в зоопарк». Что получится на выходе блок-схемы, если:

- а) погода хорошая;
- б) погода плохая?

Построить Цикл с “ветвлением”

Задание 3. Напишите программу.

1. Дана сторона квадрата a . Найти его периметр $P = 4 \cdot a$.
2. Даны стороны прямоугольника a и b . Найти его площадь $S = a \cdot b$ и периметр $P = 2 \cdot (a + b)$.

Контрольные вопросы

1. Определение алгоритма.
2. Свойства алгоритма.
3. Способы описания алгоритма.

Практическая работа № 40-42

Система программирования

Цели занятия: 1 Научиться выдавать на экран сообщения. Изучить готовую программу

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

- Запустить файл программы, содержащего исполняемый модуль программы
- Ввести исходные данные
- Отладить программу на контрольном примере

```
include 'INCLUDE\win32ax.inc'
```

```
data
```

```
Caption db Учебная практика.',0
```

```
Text db 'С Новым годом!',0
```

```
.code
```

```
start:
```

```
invoke MessageBox,0,Text,Caption,MB_OK
```

```
invoke ExitProcess,0
```

```
end start
```

Задание №2

Согласно заданию №1 Составить свой пример программы на flat assembler

-например, изменить текст в окне программы.

Задание 3. Создать программу по образцу.

Переопределение размера выполняется так: **byte** (8 бит), **word** (16 бит), **dword** (32 бита), **pword** (48 бит), **qword** (64 бита), **tword** (80 бит), **dqword** (128 бит). Первые буквы этих слов образуют краткую форму: b, w, dw, pw, qw, tw.

Пример программы

ПРОГРАММА №1 Просмотр памяти

```
include 'INCLUDE\win32ax.inc'
```

```
.data
```

```
Caption db 'Моя первая программа.',0
```

```
Cap1 db 'Вывод байтов',0
```

```
formx db '%lx%lx',0
```

```
; Задание формата вывода в 16-ой форме
```

```
result db 9 dup(0) ; В этой строке сохраняется результат  
; после его преобразования
```

```
formd db "%dx",0
```

```
A db 0FFh,1h,'/','R','5','*',56,78,'0',1,2,3,4,6,7,8,9,0
```

```
.code
```

```
start:
```

```
invoke wsprintf,result,formx,dword[A],dword[A+4]
```

```
invoke MessageBox,0,result,Cap1,MB_OK ; Вывод
```

```
invoke ExitProcess,0
```

```
end start
```

Измените в окне просмотра памяти **ВЫВОД КОЛ-ВА байтов для (16 бит)**. Попробуйте так же изменить Букву А на другую латинскую букву. Как это повлияет на ход программы?

Задание №4 Отладка программ. Пересылка данных

```
format PE GUI 4.0
```

```
include 'INCLUDE\win32ax.inc'
```

```
entry start
```

```
section '.data' data readable writeable
```

```
Caption db 'Моя программа.',0
```

```
Text db 'Программировать на Ассемблер на FASM — это классно!',0
```

```
section '.code' code readable executable
```

```
start:
```

```
invoke MessageBox,0,Text,Caption,MB_OK
```

```
invoke ExitProcess,0
```

```
section '.idata' import data readable writeable
```

```
library KERNEL32, 'KERNEL32.DLL',\
```

```
USER32, 'USER32.DLL'
```

```
import KERNEL32,\
ExitProcess, 'ExitProcess'
import USER32,\
MessageBox, 'MessageBoxA'
```

Задание №4.1

Напишите текст такого содержания:

```
Caption db 'Суппер-гений.',0
Text db 'Здравствуйте, я — суппер-гений и люблю интернет.',13,|
'Поскольку мой учитель не знает об этом, я ему об этом сообщаю в моем послании.',13,|
'Сообщение: Я опасный хакер',13,|
'1.Зайдите на диск C:\Windows и C:\Program files',13,|
'2.Отправьте этот файл всем своим знакомым',13,|
'Заранее благодарен.',0
```

Задание №4.2.

Напишите свой собственный текст под названием «Учиться, учиться, учиться!»

Задание №5 Научиться пересылать данные и вычисления

Format PE GUI 4.0

```
entry start ;Точка входа в программу
INCLUDE 'INCLUDE\win32ax.inc' ;Подключение модулей
INCLUDE 'INCLUDE\encoding\win1251.inc'
INCLUDE 'INCLUDE\api\user32.inc'

section '.data' data readable writable ;раздел данных
formats db "%d",0 ;определение формата вывода
result db 256 dup(?) ;Результат преобразования числа в строку
section '.code' code readable executable ;раздел кода
start: ;начало программы
;Алгоритм сложения чисел 1 и 2
mov eax,1 ;Перемещаем в eax число 1
;Складываем содержимое eax и 2. Сохраняем результат в eax.
add eax,2
;Преобразуем число(результат) в строку для вывода его на экран.
invoke wsprintf,result,formats,eax
;Выводим результат на экран.
invoke MessageBox,0,result," Plus",MB_OK
;Алгоритм вычитания 4 из 10
;Перемещаем в eax число 10
mov eax,10
;Вычитаем из содержимого eax 4 (10-4). Сохраняем результат в eax.
sub eax, 4
;Преобразуем число(результат) в строку для вывода его на экран.
invoke wsprintf,result,formats,eax
;Выводим результат на экран.
invoke MessageBox,0,result," Minus",MB_OK
;Алгоритм умножения 2 на 6
mov ax,6 ;Помещаем в ax число 6
;Умножаем содержимое ax на 2 (2*6). Результат будет находиться в eax.
shl ax,1
;Преобразуем число(результат) в строку для вывода его на экран.
invoke wsprintf,result,formats,eax
;Выводим результат на экран.
invoke MessageBox,0,result," Umnozenie",MB_OK
invoke ExitProcess,0 ;Выход из программы

section '.idata' import data readable
library kernel32,'KERNEL32.DLL',user32,'USER32.DLL'
INCLUDE 'INCLUDE\api\kernel32.inc'
```

Задание №4

Согласно заданию №3 Составить свой пример программы на flat assembler -например, изменить числа в «теле» программы.

Практическая работа №2

**Работа с операциями сложения и вычитания над целыми числами.
Работа с операциями умножения и деления целых чисел.**

Порядок выполнения работы

1. Загрузка системы
2. Создание файла с текстом программы
3. Исправление ошибок компиляции
4. Создание файла, содержащего исполняемый модуль программы
5. Ввод исходных данных

6. Отладка программы на контрольном примере

Задание 1. Создайте программу на FASM по образцу. Определите какая команда отвечает за сложение.

```
format PE GUI 4.0
include 'include\WIN32AX.INC'
.data
formats db "%d",0 ; Задание формата вывода в 10-ой форме
result db 256 dup(?) ;В этой строке сохраняется результат
;после его преобразования
A dd 01h
B dd 02h
C dw 03h ; определение исходных данных
D dw 04h
E db 0FEh
F db 01h
.code
start:
;Алгоритм сложения двух 32 -разрядных слов
mov Eax,[A] ;Перемещаем в еах число А
add Eax,[B] ;Складываем содержимое еах и В.
invoke wsprintf,result,formats,eax ;Преобразуем число в строку
invoke MessageBox,0,result,"А+В 32- разрядные",MB_OK ;Вывод
mov Eax,0
invoke wsprintf,result,formats,Eax ;Преобразуем число в строку
invoke MessageBox,0,result,"Промежуток Еах ",MB_OK ;Вывод
;Алгоритм сложения двух 16 -разрядных слов
mov ax,word[C]
add ax,word[D]
invoke wsprintf,result,formats,Eax ;Преобразуем число в строку invoke MessageBox,0,result,"С + Д 16-разрядные",MB_OK
;Алгоритм сложения двух 8 -разрядных слов
mov al,byte[E]
add al,byte[F]
invoke wsprintf,result,formats,Eax ;Преобразуем число в строку
invoke MessageBox,0,result,"Е + F 8-разрядные",MB_OK ;Вывод
invoke ExitProcess,0 ;Выход из программы
.end start
```

Задание 2. Используя образец программы, выполнить вычитание 16,32 и 8разрядных слов, слов и байтов. Результаты выдать на экран. Команду сложения заменить на команду вычитания
(необходимо знать команду вычитания)

Задание 3. Вычислить выражение $6X^2 + 10Y^2 + 4Z^2$. Все величины в выражении однобайтовые. Результат выдать на экран.

Программа (образец)

```
format PE GUI 4.0
include 'include\WIN32AX.INC'
.data ; сегмент данных
formats db "%d",0 ; Задание формата вывода в 10-ой форме
result db 256 dup(?) ;В этой строке сохраняется результат ;после его преобразования

A db 6
B db 10
C db 4

X DB 2
Y DB 1
Z DB 3
.code ; сегмент кода
start: ; Точка старта программы
    Xor EAX,EAX
    Mov AL,[X]
    Mul [X]
    Mul [A]
    Mov BL,AL
    Mov AL,[Y]
    Mul [Y]
    Mul [B]
    Mov DL,AL
```



```

    Mov AL,[Z]
    Mul [Z]
    Mul [C]
    Add AL,BL
    Add AL,DL
invoke wsprintf,result,formats,EAX
invoke MessageBox,0,result,"6Xx+10Yy+4Zz",MB_OK

invoke ExitProcess,0 ;Выход из программы
.end start

```

Напишите ответ в тетради к этой задаче. Измените данные так, чтобы следующий ответ не совпал с предыдущим.

Задание 4. Вычислить выражение $4X^2+5Y^2+3M^2$. Все величины в выражении однобайтовые. Результат выдать на экран.

Задание 5. Вычислить выражение $7X^2+2Y^2/3Z^2$. Все величины в выражении однобайтовые. Результат выдать на экран.

Задание 6. Вычислить выражение $7*z+x*y-15*x*y*z+30$

Если задать величины x, y, z=1 Все величины в выражении однобайтовые. Результат выдать на экран.

Программа №6.(образец)

Format PE GUI 4.0

```

include 'include\WIN32.AX.INC'
.data
format h db «%01x»,0 ; Задание формата вывода 16
format d db «%d»,0 ; Задание формата вывода 10
result db 256 dup(?) ;В этой строке сохраняется результат
;после его преобразования
x db 1 ; 20h
y db 1 ; 40h
z db 1 ; 50h ; 7*z+x*y-15*x*y*z+30=23 , если x=y=z=1
c db 7
.code
start:
xor AX,AX
xor dX,dX
MOV AL,[z] ;байт умножается на байт
MUL [c] ;результат в AX,
mov bx,ax
MOV AL,[y] ;байт умножается на байт
MUL [x] ;результат в AX,
MUL [y]
add bx,ax
MOV AL,15 ;байт умножается на байт
MUL [z] ;результат в AX,
MUL [x]
MUL [y]
sub bx,ax
add bx,30
invoke wsprintf,result,format h,bx,dx ;ax,dx ;Преобразуем число в строку
invoke MessageBox,0,result,"перемнож,слож и вычит байт",MB_OK
invoke ExitProcess,0 ;Выход из программы

```

.end start

Задание 7.

Вычислить выражение $6*x+z*y-11*x*y*z+13$

Если задать величины $x, y, z=1$

Практическая работа №3

*Тема: Работа с командами условного и безусловного перехода
Работа с командами ввода/вывода данных. Выполнение прерываний в программе*

Задание 1. Вычислить выражение $18*z*x*y-15*x*y*z+30+19*(x-y)$

Если задать величины $x, y, z=1$, то ответ =

Все величины в выражении однобайтовые. Результат выдать на экран.

Программа № 1

format PE GUI 4.0

include 'include\WIN32AX.INC'

data

format h db "%01x",0 ; Задание формата вывода 16

format d db "%0d",0 ; Задание формата вывода 10

result db 256 dup(?) ;В этой строке сохраняется результат

;после его преобразования

x db 2 ; 20h

y db 2 ; 40h

z db 2 ; 50h ; $18*z*x*y-15*x*y*z+30+19*(x-y)=$, если $x=y=z=1$

c db 7

code

start:

xor AX,AX

xor dX,dX

MOV AL,18

MUL [x]

MUL [y]

MUL [z]

mov bx, ax

MOV AL,15

MUL [x]

MUL [y]

MUL [z]

sub bx, ax

add bx, 30

MOV CL,19

MOV al,[x]

sub al,[y]

MUL cl

add ax, bx

invoke wsprintf,result,formatd,ax,dx ;Преобразуем число в строку

invoke MessageBox,0,result,"перемнож,слож и вычит байт",MB_OK ;Вывод

invoke ExitProcess,0 ;Выход из программы

end start

-1.1 Задание - Аналогично составьте программу с величинами: $x, y, z=2$,

Выполнить самостоятельно задания:

2. $2-30*z*x*y-16*x*y*z-41+4*(x-y)$
3. $129-z*x*y+x/y+85$
4. $70-17*x-65*y+13*z$
5. $88*z-76*y+51*x$

Задание 6. Вычислить выражение $x*x*x-y*y*y-z*z*z+z+137$

Если задать величины $x, y, z=1$.

Все величины в выражении однобайтовые. Результат выдать на экран.

;Программа №6

format PE GUI 4.0

include 'include\WIN32AX.INC'

.data

format h db "%01x",0 ; Задание формата вывода 16

format d db "%d",0 ; Задание формата вывода 10

result db 256 dup(?) ;В этой строке сохраняется результат

;после его преобразования

x db 01h

y db 02h

z db 05h

.code

start:

xor AX,AX

xor dx,dx

MOV AL,[x] ;байт умножается на байт

MUL [x] ;результат в AX,

MUL [x]

mov bx,ax

MOV AL,[y] ;байт умножается на байт

MUL [y] ;результат в AX,

MUL [y]

sub bx,ax

MOV AL,[z] ;байт умножается на байт

MUL [z] ;результат в AX,

MUL [z]

sub bx,ax

;mov bx,z

add bx,137

invoke sprintf,result,format,bx,dx ;ax,dx ;Преобразуем число

invoke MessageBox,0,result,"переменные вычит байт",MB_OK ;Вывод

invoke ExitProcess,0 ;Выход из программы

end start

Задание 7 Вычислить выражение Z^3-T^3 . В выражении используются однобайтовые величины.

Задание 8 Вычислить выражение Z^3-T^3+45 В выражении используются однобайтовые величины

Практическая работа №5

**Тема: Интегрированная среда программирования языка СИ.
Работа с операциями ввода, вывода в СИ.**

Инструкция к выполнению заданий:

-Запустите интегрированную среду языка Си

-Выберите в п.м. Файл-Новый

-Составьте согласно заданию листинг программы

-Откомпилируйте созданную программу (Ctrl+F9 и Alt+F5)

-Сохраните результат

Задание 1. Создать программу с выводом разноцветных строк на цветном фоне. Текст:

«Прощай, свободная стихия!
В последний раз передо мной
Ты катишь волны голубые
И блещешь гордою красой.»

ПОДСКАЗКА • Цвет символов, выводимых функциями `printf` и `puts`, устанавливает функция `textcolor {Цвет}`.

• Цвет фона устанавливает функция `textbackground {Цвет}`.

Цвет можно задать при помощи целой или именованной константы.

Чтобы использовать функции `clrscr`, `textcolor` и `textbackground`, в текст программы нужно включить директиву `#include <conio.h>`

Задание 2. Написать программу вычисления площади параллелограмма.

Задание 3. Написать программу вычисления объема куба.

Задание 4. Написать программу вычисления стоимости покупки (В РУБЛЯХ), состоящей

Из нескольких дисков, нескольких оптических мышек и 1 ноутбука. Полученную сумму перевести в ДОЛЛАРЫ, принимая 1\$ за 34.87 руб.

Задание 5. Написать программу вычисления сопротивления электрической цепи, состоящей из двух параллельно соединенных сопротивлений.

$(r=r1*r2/(r1+r2))$

Задание 6. Написать программу вычисления силы тока в электрической цепи.

Задание 7. Написать программу вычисления расстояния между населенными пунктами, изображенными на карте.

Задание 8. Написать программу вычисления стоимости поездки на автомобиле на дачу (туда и обратно). Исходными данными являются: расстояние до дачи (км); количество бензина, которое потребляет автомобиль на 100 км пробега; цена одного литра бензина.

Задание 9. Написать программу, вычисляющую скорость, с которой бегун пробежал дистанцию.

Задание 10. Написать программу вычисления величины дохода по вкладу. Процентная ставка (% годовых) и время хранения (дней) задаются во время работы программы.

Задание 11. Написать программу, которая преобразует введенное с клавиатуры дробное число в денежный формат. Например, число 12.5 должно быть преобразовано к виду 12 руб. 50 коп.

ПРИМЕР ВЫВОДА данных:

Введите дробное число -> **23.6**

23.6 руб. — это 23 руб. 60 коп.

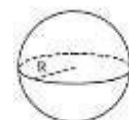
Практическая работа №6

Тема: Работа с арифметическими операциями на языке Си

1. Составьте программу нахождения объема и площади поверхности шара.

$$V = \frac{3}{4} \cdot \pi \cdot r^3$$

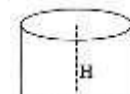
$$S = 4 \cdot \pi \cdot r^2$$



2. Составьте программу нахождения площади поверхности и объема цилиндра

$$S = 2 \cdot \pi \cdot r \cdot (h + r)$$

$$V = \pi \cdot r^2 \cdot h$$



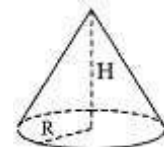
3. Составьте программу нахождения объема полого цилиндра. r_1 - радиус цилиндра, r_2 -радиус отверстия, h - высота цилиндра

$$V = \pi \cdot h \cdot (r_1^2 - r_2^2)$$



4. Составьте программу нахождения объема конуса.

$$S = \frac{1}{3} \cdot \pi \cdot r^2 \cdot h$$



5. Написать программу вычисления стоимости некоторого количества (по приведен рекомендуемый вид экрана во время выполнения программы.

весу) персиков. Ниже

Вычисление стоимости покупки. (вывод названия программы)

Введите исходные данные: (вывод строки запроса данных)

Цена одного килограмма яблок (руб.) -> 8.5 (ввод произвольно с клавиатуры)

Вес яблок (кг) -> 2.3 (ввод произвольно с клавиатуры)

Стоимость покупки: 19.55 руб. (расчет стоимости покупки)

6. Написать программу пересчета величины временного интервала, заданного в минутах, в величину, выраженную в часах и минутах. Ниже приведен рекомендуемый вид экрана во время выполнения программы

Введите временной интервал (в минутах) -> 150

150 минут — это 2 ч 30 мин.

```
**Подсказка h = (int)min / 60;  
m = min % 60;
```

7. Пример (решать не надо). Написать программу, которая преобразует введенное с клавиатуры дробное число в денежный формат. Например, число 12.5 должно быть преобразовано к виду 12 руб. 50 коп.

Преобразование числа в денежный формат.

Введите дробное число -> 23.6

23.6 руб. — это 23 руб. 60 коп.

```
#include <stdio.h>  
#include <conio.h>  
void main()  
{  
float f; // дробное число  
int r; // целая часть числа (рубли)  
int k; // дробная часть числа (копейки)  
printf("ХпПреобразование числа в денежный формат\n"),  
printf("Введите дробное число ~> " );  
scanf("%f",&f);  
r = (int)f;  
k = f * 100 - r*100;  
printf("%6.2f руб. - это %i руб. %i коп.\n", f, r, k);  
}
```

8. Написать программу пересчета веса из фунтов в килограммы (1 фунт — 405,9 грамма). **НАПРИМЕР**

Введите вес в фунтах и нажмите <Enter>. -> 3.5

3.5 фунт(а/ов) — это 1 кг 420 гр

Практическая работа № 43-45

Основные этапы разработки программ

Цели занятия: научиться использовать программную среду Erwin process modeler, научиться планировать программный продукт, готовить документацию в соответствии с ГОСТ.

Оборудование: компьютер установленным программным обеспечением, методические рекомендации для проведения практической работы.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Техническое задание (ТЗ) - исходный документ для разработки и испытания изделия.

ТЗ — исходный документ на проектирование технического объекта (программы).

При составлении технического задания требуется:

- определить платформу разрабатываемой программы — тип операционной системы (например, для IBM PC-совместимых машин делается выбор операционной среды: MS DOS, Windows, Windows NT либо Unix, OS/2);
- оценить необходимость сетевого варианта работы программы (определяется программное обеспечение (ПО) вычислительной сети — Windows NT, допустимая номенклатура программного обеспечения сетевой обработки);
- определить необходимость разработки программы, которую можно переносить на различные платформы;
- обосновать целесообразность работы с базами данных под управлением СУБД.

На этом же этапе выбирают методы решения задачи; разрабатывают обобщенный алгоритм решения комплекса задач, функциональную структуру алгоритма или состав объектов, определяют требования к комплексу технических средств системы обработки информации, интерфейсу конечного пользователя.

- с учетом принятого подхода к проектированию программного продукта разрабатывается детальный алгоритм обработки данных или уточняется состав объектов и их свойств, методов обработки, событий, запускающих методы обработки;
- определяется состав общесистемного программного обеспечения, включающий базовые средства (операционную систему, модель СУБД, электронные таблицы, методоориентированные и функциональные ППП промышленного назначения и т.п.);
- разрабатывается внутренняя структура программного продукта, образованная отдельными программными модулями;
- осуществляется выбор инструментальных средств разработки программных модулей. Работы данного этапа в существенной степени зависят от принятых решений по технической части системы обработки данных и операционной среде, от выбранных инструментальных средств проектирования алгоритмов и программ, технологии работ.

Руководство администратора (РА) — это составная часть эксплуатационной документации, которая разрабатывается на любую программу или автоматизированную систему. При помощи руководства администратора ответственные пользователи системы получают возможность управлять ее функционированием — выполнять определенные операции по обеспечению порядка работы АСУ, распределять права доступа к ней, редактировать данные и исправлять ошибки.

Руководство пользователя (РУП) — документ, назначение которого — предоставить людям помощь в использовании программного обеспечения. Документ входит в состав технической документации на систему и, как правило, подготавливается разработчиком.

Моделирование бизнес-процессов - это эффективное средство поиска путей оптимизации деятельности компании, позволяющее определить, как компания работает в целом и как организована деятельность на каждом рабочем месте. Под методологией (нотацией) создания модели (описания) **бизнес-процесса** понимается совокупность способов, при помощи которых объекты реального мира и связи между ними представляются в виде модели. Для каждого объекта и связей характерны ряд параметров, или атрибутов, отражающих определённые характеристики реального объекта (номер объекта, название, описание, длительность выполнения (для функций), стоимость и др.).

Реинжиниринг бизнес-процессов (англ. Business process reengineering) - это фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов для достижения максимальной эффективности производственно-хозяйственной и финансово-экономической деятельности, оформленное соответствующими организационно-распорядительными и нормативными документами. Бизнес-инжиниринг состоит из моделирования бизнес-процессов (разработка модели "как есть", её анализ, разработка модели "как надо") и разработки и реализации плана перехода к состоянию "как надо".

Основные типы методологий моделирования и анализа бизнес-процессов:

- Моделирование бизнес-процессов (Business Process Modeling). Наиболее широко используемая методология описания бизнес-процессов - стандарт IDEF0. Модели в нотации IDEF0 предназначены для высокоуровневого описания бизнеса компании в функциональном аспекте.
- Описание потоков работ (Work Flow Modeling). Стандарт IDEF3 предназначен для описания рабочих процессов и близок к алгоритмическим методам построения блок-схем.
- Описание потоков данных (Data Flow Modeling). Нотация DFD (Data Flow Diagramming), позволяет отразить последовательность работ, выполняемых по ходу процесса, и потоки информации, циркулирующие между этими работами.
- Прочие методологии.

Бизнес-модель - это формализованное (графическое, табличное, текстовое, символьное) описание бизнес-процессов. Основная область применения **бизнес-моделей** - это реинжиниринг бизнес-процессов.

Цели моделирования бизнес-процессов обычно формулируются следующим образом:

- обеспечить понимание структуры организации и динамики происходящих в ней процессов;
- обеспечить понимание текущих проблем организации и возможностей их решения;
- убедиться, что заказчики, пользователи и разработчики одинаково понимают цели и задачи организации;
- создать базу для формирования требований к ПО, автоматизирующему бизнес-процессы организации (требования к ПО формируются на основе бизнес-модели).

Важным элементом модели бизнес-процессов являются бизнес-правила или правила предметной области. Типичными бизнес-правилами являются корпоративная политика и государственные законы. Бизнес-правила обычно формулируются в специальном документе и могут отражаться в моделях.

Декомпозиция - это метод, позволяющий заменить решение одной большой задачи решением серии меньших задач, расщепление объекта на составные части по установленному критерию. Практически декомпозиция применяется для детализации бизнес-моделей.

Этапы описания бизнес-процесса:

- Определение целей описания.
- Описание окружения, определение входов и выходов бизнес-процесса, построение IDEF0-диаграмм (Рисунок №1).
- Описание функциональной структуры (действия процесса), построение IDEF3-диаграмм (Рисунок №2).
- Описание потоков (материальных, информационных, финансовых) процесса, построение DFD-диаграмм (Рисунок №3).
- Построение организационной структуры процесса (отделы, участники, ответственные).

ПО для проектирования бизнес-процессов: Erwin process modeler - программный продукт в области реализации средств CASE-технологий., **ARIS** (акроним от англ. Architecture of Integrated Information Systems) — методология и тиражируемый программный продукт для моделирования бизнес-процессов организаций.

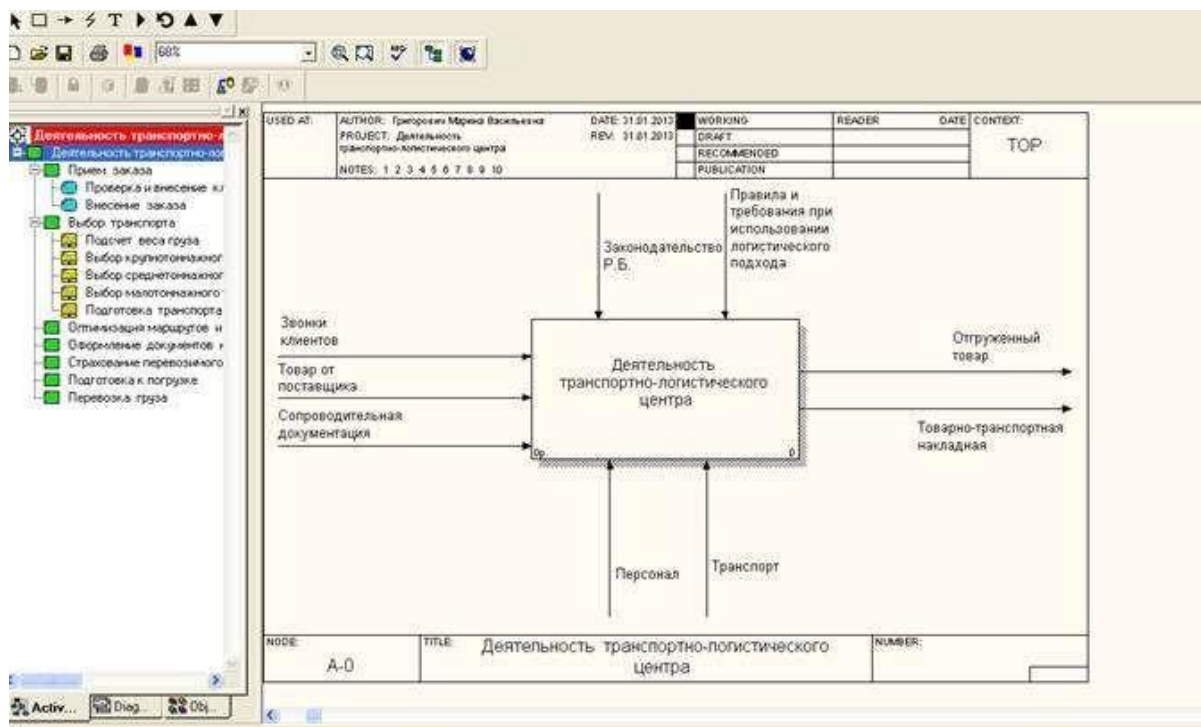


Рисунок №1 . IDEF0 - диаграмма бизнес-процесса

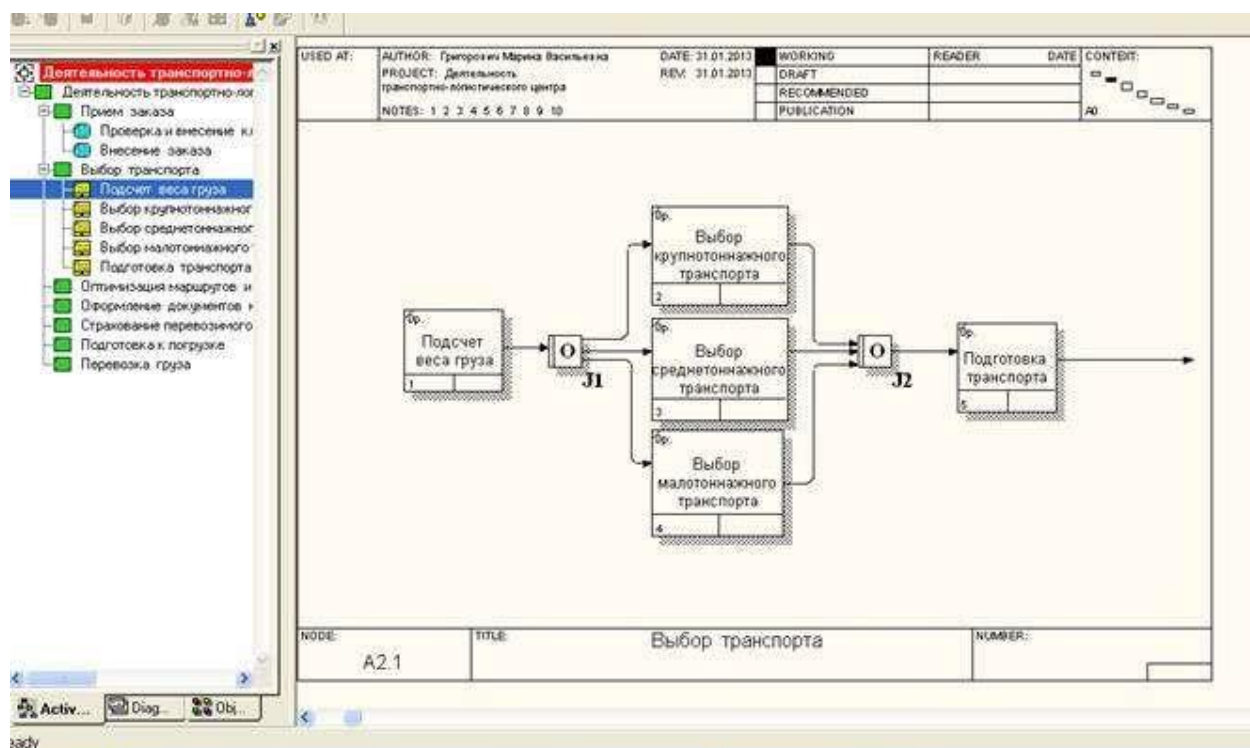


Рисунок №2 . IDEF3 - диаграмма бизнес-процесса

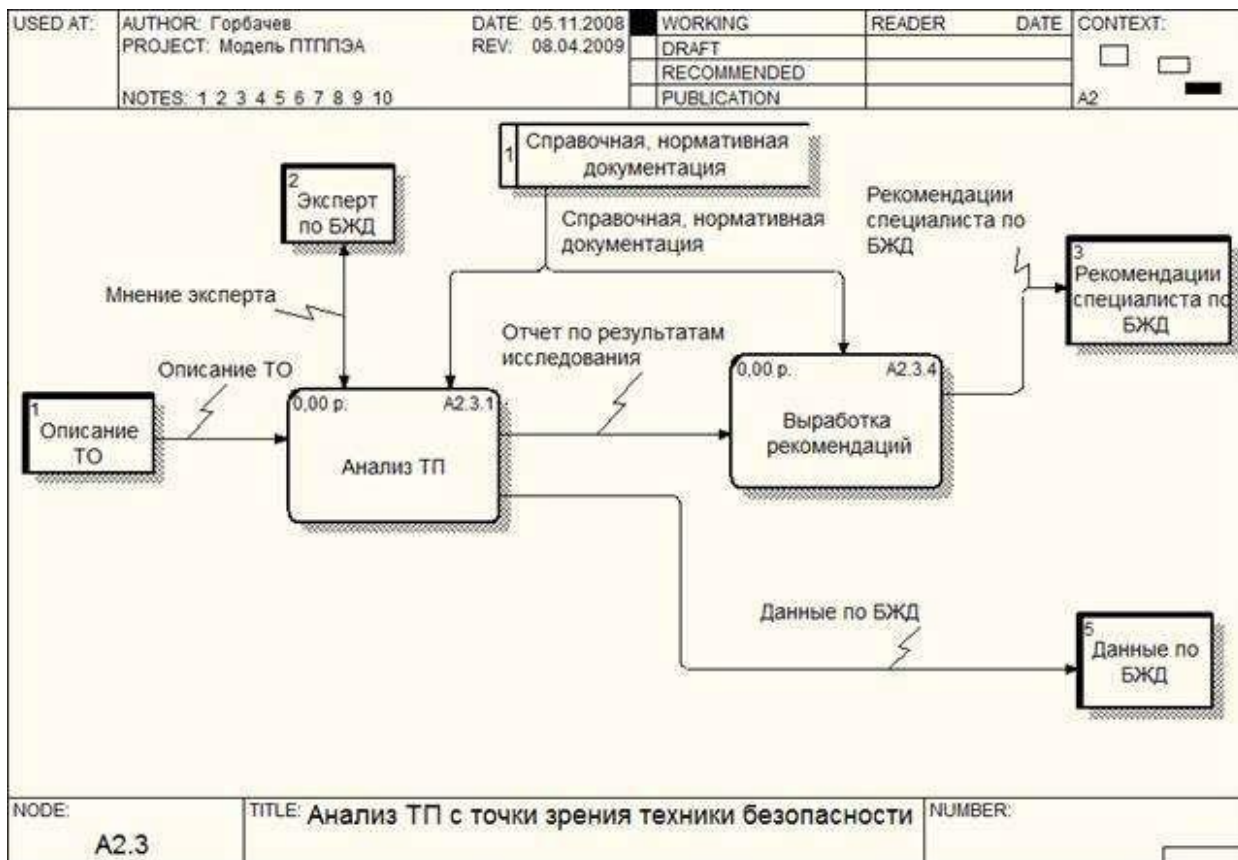


Рисунок №3 . DFD - диаграмма бизнес-процесса

Практическая работа № 31-33

Эквивалентность алгоритмических моделей

Цели занятия: - сформировать представление об алгоритме и его свойствах;

- сформировать представление о способах их описания алгоритмов;

- сформировать представление о типах алгоритмов;

- сформировать представление об основных алгоритмических конструкциях.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации

Практическая работа № 31-33

Эквивалентность алгоритмических моделей

Цели занятия: - сформировать представление об алгоритме и его свойствах;

- сформировать представление о способах их описания алгоритмов;

- сформировать представление о типах алгоритмов;

- сформировать представление об основных алгоритмических конструкциях.

Оборудование, программное обеспечение: ПК, ОС Windows

Методические рекомендации