**Кисленко Николай Петрович, каф. Прикладной математики, оф. 436, тел. 266-05-41,** [**kis@sibstrin.ru**](mailto:kis@sibstrin.ru)

**3 семестр: базовая информатика (16 ч. лекций + зачет в форме тестирования test.sibstrin.ru или фэпо.рф)**

**4 семестр: основы работы в MathCAD + экзамен**

**5 семестр: численные методы и решение инженерных задач в Mathcad + экзамен**

**Пособие: Информатика. Базовый курс (в 3 частях)**

**НГАСУ, 2011-12**

**Ч.1. Воробьева А.П. 1. Теоретич.информатика 2. Архитектура ЭВМ**

**Ч.2. Ершова Е.Е. 3. Программное обеспечение 4. Моделирование**

**Ч.3. Кисленко Н.П. 5. Алгоритмизация и прогр-е. 6. Сети и Интернет**

**В библиотеке; или sibstrin.ru – Студентам – Учебные материалы – Информатика**

* 1. **Данные и единицы из измерения**

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Символ** | **Степень 2** | **~Степень 10** |
| **Байт** | **Б** | **0** | **0** |
| **Килобайт** | **Кб** | **10** | **3** |
| **Мегабайт** | **Мб** | **20** | **6** |
| **Гигабайт** | **Гб** | **30** | **9** |
| **Терабайт** | **Тб** | **40** | **12** |
| **Петабайт** | **Пб** | **50** | **15** |

**Пр.1. Имеется 233 бит. Объем в Мб = ?**

**=2^33/8/1024/1024 (1024)**

**Строчка вниз по таблице соответствует делению на 1024 (210)**

**Пр.2. 1 Тб = ? Кб**

**=1\*1024\*1024\*1024 или =2^40/2^10**

**Пр.3. Скорость передачи данных = 256000 бит/сек. Передача файла заняла 2 мин. Размер файла в Кб=?**

**256000/8 = 32000 байт/сек \* 120 сек /1024**

**(3750 Кб)**

**Пр.4. Ск.передачи = 28800 бит/сек. Ск. времени нужно для передачи файла в 72000 байт?**

**28800/8 = 3600 байт/сек**

**72000 / 3600 = 20 сек.**

* 1. **Двоичное кодирование данных**

**Формула Хартли: N = 2M,**

**где N – кол-во различных значений, к-рые можно закодировать**

**M – кол-во бит, требуемых для кодирования**

**Пр.1.Сколько бит нужно для кодирования инф-ции о колоде из 36 карт (4 масти, 9 карт каждой масти)**

* **Найти ближайшую сверху степень "двойки" к кол-ву разных кодируемых значений**

**Т.к. 2^2 >= 4 то 2 бита на масть**

**Т.к. 2^4 >= 9 то 4 бита на величину (4+2=6)**

**Или т.к. 2^6 >= 36**

**Пр.2. Ск.бит нужно для кодирования а) 26 букв латиницы б) 33 букв кириллицы а) 5 б) 6**

**Пр.3. Инф-ний объем 1 точки 256-цветного растрового изображения = ? байт**

**"2 в какой степени >=256?" 8**

**Пр.4. Как изменится объем файла, если исходное кол-во цветов = 216, а затем его сохранили с 232 цветов?**

**2^16 цветов – 16 бит на точку**

**2^32 – 32 бита на точку**

**Если пренебречь заголовком файла, объем увеличится в 2 раза**

**Пр.5.Есть 32 мяча, из к-рых X – желтые. Вынимается 1 мяч. Сообщение "извлечен мяч не желтого цвета" несет 4 бита информации. X=?**

**Варианты: 32 30 16 8**

**Формула Шеннона: кол-во информации о событии I связано с его вероятностью p формулой**

**I = LOG 2 (1/p)**

**Если кол-во инф-ции, полученной во время события, равно I бит, то вер-ть этого события = 2-I. Т.к. фраза несет 4 бита инф-ции, вер-ть того, что мяч не желтый = 1/24 = 1/16. Значит, не желтых мячей 1/16 часть 32/16 = 2. Тогда желтых 30**

**Или решая ур-ние с известным I=4, получим вер-ть 1/16**

**Пр.6. Ск. есть различных посл-стей символов "a" и "b" длиной 5 ?**

**A кодируем битом 0, b – битом 1 … то 2^5 = 32**

**Пр.7. Ск.есть различных посл-тей из 3 символов "а" и 2 символов "б"**

* **Для выборок с *повторениями*, часть эл-тов которых неразличима (шары одного цвета, одинаковые буквы) действуют правила:**
* **Расположение N объектов по K различным ячейкам имеет NK вариантов (при N=2 – формула Хартли)**
* **На мн-ве из N эл-тов, среди которых k1 1-го вида, k2 2-го вида, …, km m-го вида, кол-во различных перестановок =**

**N! / (k1!\*k2!\*…\*km)**

**У нас N=5, k1=3, k2=2, 5! / (3!\*2!) = 10**

**Пр.8.Ск. есть способов переставить красный, зеленый и синий шарики?**

* **Число *перестановок* из N различных объектов по K выбранных = N! / (N-K)!**

**3!/0! = 6, т.к. 0! по определению = 1**

**или по пред. Формуле = 3! / (1!\*1!\*1!) = 6**

**Пр.9. Ск. способов выбрать 2 шарика из 4?**

**= 4! / (4-2)! = 12**

* **Если имеем не перестановка, а *комбинацию* (порядок эл-тов в выбранной группе не имеет значения), то число *сочетаний* из N объектов по K =**

**C NK = N! / (K! \* (N-K)! )**

**Пр.10. Перестановки по 2 шарика из 3**

**3! / (3-2)! = 6 =ПЕРЕСТ(N;K)**

**12 21 13 31 23 32 в Excel!**

**Комбинации по 2 шарика из 3**

**3! / (2!\*1!) = 3 =ЧИСЛКОМБ(N;K)**

**12 13 23**

**Дома:**

**Пр.11. Объём сообщ. = 32 Кб, 64 страницы по 16 строк по 64 символа. Алфавит, к-рым записано сообщение, включает не более ? символов**

**=64\*16\*64 = 65536 символов**

**=32\*1024\*8 = 262144 бит для кодирования**

**Поделив, получили 4 бита на символ**

**Можно закодировать не более 24=16 значений (емкость алфавита)**

**Пр.12. Имеется 3 красных и синих шарика. Ск. способов их переставить?**

**Перестановки, 2 вида объектов**

**= 6! / (3!\*3!) = 120/6 = 20 способов**

**Пособие ч.1, 1.1-1.3**

* 1. **Системы счисления (с.с.)**

**Число A в позиционной с.с. представляется как**

**A = an-1\*pn-1 + an-2\*pn-2+ … a0\*p0+ a-1\*p-1+ … +**

**a-m\*p-m**

**p – основание с.с.,**

**ai – цифры числа,**

**n – кол-во цифр в целой части,**

**m – в дробной**

**Пр. 123.510 = 1\*102 + 2\*101 + 3\*100 + 5\*10-1**

**Число в с.с. с основанием P включает цифры 0,1,..P-1. Если P от 11 до 36, вместо недостающих цифр исп. буквы A … Z**

**Макс. N-разрядное число в с.с. с осн. P =**

**PN-1 - 1 – для чисел со знаком**

**PN – 1 – для чисел без знака**

**Пр. Для 2-байтового целого со знаком макс. значение = … 215 – 1 = 32767**

**Перевод A10 -> QN : в целых числах делим A на N, выписываем остатки от деления. Закончить, когда останется число <N. Q = последовательности остатков, прочитанной справа налево**

**Пр. 1210 - ?2**

|  |  |  |  |
| --- | --- | --- | --- |
| **12** | **6** | **3** | **1 <2, читаю** |
| **остатки** | **0** | **0** | **1** |

**= 1100**

**Пр. 25410 - ?16**

|  |  |  |  |
| --- | --- | --- | --- |
| **254** | **15** | **15=F <16,** | **читаю** |
| **остатки** | **14=E** |  |  |

**= FE**

**Перевод AN -> Q10 : цифры числа A нумеруем справа налево с нуля и последовательно умножаем их на соответствующие степени числа N; произведения складываем**

**Пр. 378 -> ?10**

|  |  |  |
| --- | --- | --- |
| **3** | **7** |  |
| **3\*81** | **7\*80** | **=24+7 = 31** |

**В стандартном Калькуляторе: выбрать Вид – Программист, реализованы переводы между с.с. с осн. 2,8,10,16**

**Переводы между с.с. с осн. 16 и 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **16-ричная цифра** | **Двоичная запись** | **16-ричная цифра** | **Двоичная запись** |
| **0** | **0000** | **8** | **1000** |
| **1** | **0001** | **9** | **1001** |
| **2** | **0010** | **A** | **1010** |
| **3** | **0011** | **B** | **1011** |
| **4** | **0100** | **C** | **1100** |
| **5** | **0101** | **D** | **1101** |
| **6** | **0110** | **E** | **1110** |
| **7** | **0111** | **F** | **1111** |

**Пр. 110011110 = 0001 1001 1110 = 19E**

**(справа налево разбили на четверки цифр, если надо, слева дополнили нулями, перевели по таблице)**

**Пр. -201310 –> ?2**

**Переводим +2013 , получаем 111 1101 1101**

**Дописываем слева нули до нужного количества бит, напр., для 2 байт 0000 0111 1101 1101**

**Инвертируем (обращаем) все биты**

**1111 1000 0010 0010**

**В двоичной с.с. прибавляем 1 к обратному числу**

**1111 1000 0010 0011**

* 1. **Основы матем.логики**

**Высказывание – повествовательное предложение, о к-ром однозначно можно сказать, что оно истинно (true) или ложно (false).**

**Логические операции служат для построения составных высказываний из простых. Описываются *таблицами истинности*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выск.A** | **Выск.B** | **Отрицание**  **-A** | **Конъюнкция**  **A ^ B** | **Дизъюнкция**  **A v B** |
| **0** | **0** | **1** | **0** | **0** |
| **0** | **1** |  | **0** | **1** |
| **1** | **0** | **0** | **0** | **1** |
| **1** | **1** |  | **1** | **1** |

**Отрицание: NOT, "не", -A**

**Конъюнкция: AND, "и", A ^ B, A & B, лог.умножение**

**Дизъюнкция: OR, "или", A v B, A | B, лог.сложение**

**Строгая дизъюнкция XOR отличается от OR тем, что 1 на 1 = 0**

**Старшинство (по убыванию): 1) NOT 2) AND 3) OR и XOR**

**Пр. A and not B or C ===**

**(A and (not B)) or C**

**Пр. –X v Y ^ X v Z, если X=0, Y=Z=1**

**1 v 1 ^ 0 v 1**

**1 v 0 v 1 == 1**

**Для упрощения лог.выражений используются законы логики (см. с .62 эл.версии)**

**Дома: ч.1, пп. 1.4-1.6**

**Упростить:**

1. **Not (A and B) or B = Not A Or Not B or B =**

**Not A Or 1 = 1**

1. **Not (A or B) and B or C = Not A And Not B And B Or C = Not A And 0 Or C = 0 Or C = C**

**Логические элементы ЭВМ – см. ч. 1 с. 64-66 эл. Версии**

**Тема 2 (ч. 1) – на сам.изучение**

**Тема 3. Программное обеспечение (ПО) ЭВМ (ч.2 пособия)**

**Классификация ПО:**

**Системное ПО – для обеспечения работы ЭВМ и сетей – операционные с-мы, драйверы**

**Прикладное ПО – для решения задач в конкретных предм.областях**

**ОС – комплекс программ, предназначенный для:**

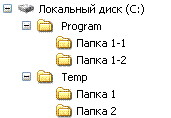
* **Управления работой устройств ЭВМ;**
* **Управления выполнением приложений;**
* **Управления файловой системой;**
* **Поддержки интерфейса пользователя.**

**Классификация ОС:**

**Однозадачные/однопользовательские (DOS, мобильные)**

**Многозадачные/многопользовательские (Windows-, Unix-совместимые)**

**Основные концепции ОС Windows:**

* **Мультизадачность;**
* **Визуальная ориентированность;**
* **Объектная ориентированность;**
* **Универсальный графич.интерфейс (GUI);**
* **Технология внедрения и связывания объектов;**
* **Технология обмена данными (Буфер Обмена)**

**Файловая система.**

**Задача 1. Пути к файлам**

**1. Полный путь (с указанием диска)**

**c:\Program\text.txt**

1. **Путь от корня диска (*головного каталога*)**

**\Temp\Папка1\Привет.docx –**

**"файл находится в папке \Temp\Папка1\ *текущего диска*"**

1. **Путь от текущей папки: ./Вася.bmp – "файл находится в текущей папке"**

**Папка1/Вася.bmp – "… в папке Папка1, вложенной в текущую"**

1. **Путь от родительской папки:**

**../../Temp/Папка 1 – "путь из Папка 1-1 в Папка 1"**

**Задача 2. Маски в именах файлов/папок – позволяют описать сразу группу файлов с похожими именами:**

**\* в имени файла/папки – любое непустое кол-во любых символов;**

**? -=- - один допустимый символ или пусто**

**\*.\* - любые файлы любых типов**

**\*.txt – все файлы с типом "txt"**

**A\*.\* - все файлы с именем на "А"**

**?????.txt – файлы типа txt с именем до 5 букв**

**Технологии обработки текстовой информации**

|  |  |  |
| --- | --- | --- |
| **Тип файла** | **Описание** | **Стандартная программа** |
| **txt** | **Неформатированный текст с разделением на абзацы** | **Блокнот, Notepad++** |
| **html/xml** | **Текстовый, с разметкой текстовыми командами (тегами) и связанными объектами** | **Браузеры** |
| **pdf** | **Бинарная копия страницы с защитой от изменений** | **Adobe Acrobat, браузеры** |
| **docx** | **Бинарный, с внедренными или связанными объектами, программами-макросами** | **MS Office** |
| **rtf** | **Текстовый, размечен тегами, нет макросов** | **MS Office** |

**Кодировка – соответствие между символами и кодирующими их двоичными числами. Символы с кодами 0-127 (ASCII) одинаковы во всех кодировках;**

|  |  |  |
| --- | --- | --- |
| **Название** | **Размер символа, байт** | **Область применения** |
| **Windows-1251** | **1** | **ОС Windows** |
| **DOS-866** | **1** | **OC DOS** |
| **KOI-8R** | **1** | **ОС Unix** |
| **Unicode** | **2 или более** | **Все совр.ОС** |

**Шрифты. Ед.измерения – пункты (pt). 72pt = 1 дюйм (на печати)**

**2 типа шрифтов:**

**Матричные (значок "принтер" или "A") – описание символа хранится как матрица точек (пикселов)**

**(-) ниже качество, нет произв.размера**

**(+) выше ск. обработки и печати**

**Векторные (значок "TT" или "O") – описание символа – это его геометрич.модель, составленная из примитивов (отрезок, дуга, …) и отношений между ними**

**(-) трудоемкость разработки и обработки**

**(+) высокое качество, произв.размер**

**Свойства:**

* **Размер в pt;**
* **Начертания (Ж, *К*, Ч);**
* **Кернинг (межсимвольные интервалы).**

**Стандартные:**

|  |  |
| --- | --- |
| **Название** | **Описание и применение** |
| **Times** | **Шрифты печатной машинки; основной текст (для печати)** |
| **Arial** | **Заголовочные (без отточий); заголовки и экранный текст** |
| **Courier** | **Моноширинные (все символы одной ширины); листинги и др. тексты с фикс. отступами** |

**Структура и форматирование текст.док.**

|  |  |
| --- | --- |
| **Элемент** | **Команды Word 2007 и выше** |
| **Символ** | **Вставка, Символ** |
| **Абзац** | **Главная, Абзац** |
| **Колонка** | **Разметка страницы, Колонки** |
| **Колонтитулы** | **Вставка, Колонтитулы** |
| **Страница** | **Разметка страницы** |
| **Раздел** | **Разметка страницы, Разрывы** |
| **Документ** | **Ссылки, Рецензирование** |

**Технология обработки электронных таблиц (Excel)**

**В ячейках листа Excel располагаются объекты 3 типов: Число; Строка; Формула (со знака =)**

**Принцип: "1 ячейка = 1 величина"**

**Всё форматирование: комб. клавиш Ctrl+1 (Формат ячеек)**

**В формулы могут входить: числа, ссылки на ячейки, 5 арифм.действий (^ степень, \* / + - )**

**Станд. Функции: =имя(аргумент1; …; аргументN)**

**СУММ – сумма аргументов**

**СРЗНАЧ – арифм.среднее**

**МАКС, МИН – максимум и минимум**

**ПРОИЗВЕД – произведение**

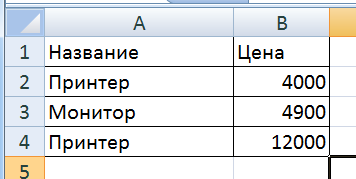
**Аргументами м.б. : отдельные ячейки и диапазоны**

**=СУММ(A1:B2) – сложить числа из A1, A2, B1, B2**

**=СУММ(A:A) – сложить все числа из столбца A**

**Условные расчеты.**

1. **=СУММЕСЛИ(Д1;У;Д2) – просматривая диапазон Д1, складывает числа из диапазона Д2, отвечающие условию У**

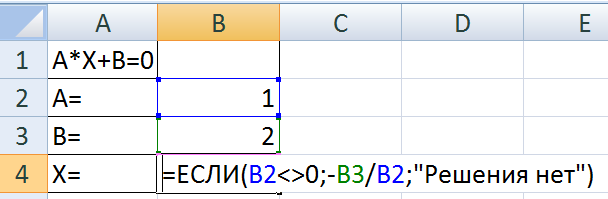
****

**=СУММЕСЛИ(A:A;"Принтер";B:B) – сложил цены только принтеров**

1. **=СЧЁТЕСЛИ(Д1;У) – количество ячеек в диапазоне Д1, отвечающих условию У**

**=СЧЕТЕСЛИ(A:A;"Принтер") – подсчитал кол-во записей о принтерах**

1. **=ЕСЛИ(У; Ф1; Ф2) Если условие истинно, считается формула Ф1, иначе Ф2**

****

**Операции And, Or реализованы как функции И, ИЛИ:**

**Пр. В столбце B – зарплаты, в C – пометки "+" з/п от 20 до 40 т.р.**

**=ЕСЛИ(И(B2>=20000;B2<=40000);"+";"-")**

1. **ЕСЛИ можно вкладывать в любые функции счёта для выполнения условных расчётов любой сложности:**

**=ПРОИЗВЕД(ЕСЛИ(A:A=0;1;A:A))**

**Перемножит числа из столбца без нулей (заменили на 1)**

**=МАКС(ЕСЛИ(E:E<0;E:E;-1E30))**

**Максимум по столбцу E без отрицательных чисел (заменили на -1030)**

**Формула с диапазоном завершается комбинацией клавиш Ctrl+Shift+Enter !**

**Технологии обработки графической инф-ции**

**Классификация графики:**

1. **По типу кодирования:**

**Растровая (точечная) – хранится инф-ция о цвете каждой точки рисунка**

**(+) простота хранения и кодирования, высокое качество представления**

**(-) при любом масштабировании качество только ухудшается**

**Векторная - хранится матем.модель рисунка, составленная из графич.примитивов (дуг, отрезков и т.д.)**

**(+)качество не меняется при масштабировании**

**(-) более ограничены выразит.средства**

**2. по типу хранения:**

**Сжатая – рисунок хранится как архив;**

**Сжатая с потерей качества – при архивировании исключается инф-ция о малых цветовых различиях;**

**Несжатая – файл хранится "как есть"**

**3. по кол-ву представляемых цветов**

|  |  |
| --- | --- |
| **Бит на точку** | **Описание** |
| **1** | **Монохромная (ч/б)** |
| **2** | **4-цветная графика CGA** |
| **4** | **16-цветная EGA/VGA** |
| **8** | **256-цветная SuperVGA** |
| **16** | **Режимы High Color (65536 цв.)** |
| **24** | **Режимы Real Color (~16.7 млн)** |
| **32** | **True Color (>4 млрд.)** |

**Графические форматы:**

|  |  |  |
| --- | --- | --- |
| **Тип файла** | **Описание** | **Приложения** |
| **.bmp** | **Несжатая, растровая, от 2 до 224 цветов** | **Paint** |
| **.jpg** | **Растровая, Real Color, сжатая с потерей кач.** | **Интернет(фото), цифровая техника** |
| **.gif** | **Растровая,сжатая,до 256 цв.,эффекты** | **Интернет(баннеры),**  **GIF Animator** |
| **.png** | **Растровая,сжатая,True Color** | **Интернет,мобильные устр-ва,новые ОС** |
| **.psd** | **Растрово-векторная, сжатая** | **Photoshop** |
| **.cdr** | **Векторная** | **Corel Draw** |
| **.wmf** | **Векторная** | **MS Office** |

**Основная модель представления цвета в графике: RGB (Red-Green-Blue), цвет = совокупности интенсивностей 3 базовых цветов. В Real Color интенсивность каждой компоненты представлены 1 байтом, цвет запис. как #RRGGBB, где RR, GG, BB – 16-ричные интенсивности красного, зеленого и синего.**

**#000000 черный #FFFFFF белый**

**#FF0000 красный #000080 темно-синий**

**#CCCCCC серый, светлее #333333 серый, темнее**

**Основы систем управления базами данных (СУБД)**

**Принципы организации данных в БД:**

* **Полнота и непротиворечивость данных;**
* **Минимальная избыточность;**
* **Минимальная зависимость от особенностей прогр. или аппар.реализации**

**Уровни представления данных в БД:**

|  |  |
| --- | --- |
| **Внешний** | **Визуальное представление данных конечного пользователя** |
| **Языки управления БД (ЯУБД; SQL – станд.язык запросов)** | |
| **Концептуальный (логический)** | **Матем.модель, условное представление БД как объектов и связей между ними** |
| **Приложение СУБД (Access)** | |
| **Внутренний (физический)** | **Программно-аппаратная реализация** |

**Модели представления данных:**

1. **Иерархическая – "дерево", где узлами являются объекты данных, а ветви показывают связи. У каждого объекта 1 предок и N>=0 потомков**

**(+) простота и однозначность представления, лёгкость адресации**

**(-) недостаточная гибкость**

**Исп. – программирование, матем.моделирование**

**Пр. – дерево папок ОС**

1. **Сетевая – теоретически возможны любые связи между объектами**

**(+) высокая гибкость моделирования**

**(-) сложность моделирования и исп-я**

**Исп. – автоматиз.системы управления, экспертные системы**

**Пр. – служба www**

1. **Реляционная – представление данных в виде взаимосвязанных таблиц**

**(+) простота, относительная нез-сть от средств реализации**

**(-) существенная зависимость скорости обработки от объёма БД**

**Исп.-е – все существующие СУБД**

**Осн.понятия СУБД:**

**Поле (атрибут) – столбец таблицы, описывающий 1 свойство объекта**

**Запись (кортеж) – строка таблицы, запись об 1 объекте**

**Таблица (отношение) – совокупность записей с заполненными значениями атрибутов**

**Первичный ключ – поле, уникальным образом идентифицирующее запись. (Пр.: фамилия – нет, № зачетки и паспорта – да)**

**Внешний ключ – поле, содержащее ссылку на поле в другой таблице (для связывания таблиц)**

**Главная таблица**

|  |  |  |
| --- | --- | --- |
| **№ (перв.ключ)** | **ФИО** | **Должность** |

**Справочник "Оклады"**

|  |  |
| --- | --- |
| **Должность (внеш.ключ)** | **Оклад** |

**В справочнике 1 оклад для каждой должности;**

**В глав.таблице >1 записи с одинак.должностью**

**Структура БД (схема) – совокупность инф-ции о таблицах и их связях**

**Типы связей между данными:**

**1:1 ("один к одному") – запись в табл. A имеет не более 1 связанной записи в табл. B и наоборот.**

**Исп.е – разделение доступа, сохранение сведений, относящихся к подмножеству табл. A**

**Пр. "Студент" – "№ зачетки"**

**1:N ("один ко многим") – записи в табл. А может соотв. несколько записей в табл. B, но не наоборот**

**Исп.е – основная связь между главной и подчиненной таблицами**

**Пр. "Должность" – "Должностной оклад"; "Группа" – "Студент"**

**N:M ("многие ко многим") - = - = - И НАОБОРОТ**

**Реализуется как 2 отношения 1:N через связующую таблицу.**

**Пр. "Студенты" – "Дисциплины", "Заказы" – "Товары"**

**Заказы**

|  |  |
| --- | --- |
| **$ Код заказа** | **Заказчик** |
| **10001** | **Sibstrin** |
| **10002** | **…** |

**Товары**

|  |  |
| --- | --- |
| **$ Код товара** | **Товар** |
| **21** | **Принтер** |
| **22** | **…** |

**Основные объекты БД:**

**Таблицы + схема данных (связи)**

**Запросы – таблица, автоматически сформированная по данным одной или нескольких таблиц с помощью SQL**

**Формы – окно для просмотра и редактирования данных из таблиц**

**Отчеты – документ для анализа/печати таблиц или запросов**

**Задача.** Создать базу данных "Сотрудники" со следующей структурой:

|  |  |
| --- | --- |
| * Табельный номер сотрудника (1-й ключ); * Фамилия и.о.; * Пол; * Дата рождения; * *Код подразделения*; * *~~Наименование подразделения~~*; | * Должность; * *Разряд по тарифной сетке;* * *~~Оклад в соответствии с разрядом~~*; * *Стаж работы, лет;* * *~~Надбавка за стаж~~* |

**Дома: ч. 2 пособия тема 2 ("Моделирование")**

**Тема 5. Алгоритмизация и программирование**

**(ч.3, тема 1)**

**1) Определение и свойства алгоритма – с. 7**

**Алгоритм – однозначно определенная последовательность действий, записанная на понятном исполнителю *алгоритмическом языке*, определяющая процесс перехода от *исходных данных* к *результату*.**

**Свойства алгоритма:**

**1. Дискретность – состоит из отдельных инструкций (шагов);**

**2. Однозначность (детерминированность) – в одинаковых начальных условиях получаются одинаковые результаты, каждый шаг понимается исполнителем единственным образом;**

**3. Результативность – за конечное число шагов достигается некий результат;**

**4. Массовость – алгоритм решает класс задач, различающихся исходными данными.**

**Формы записи:**

1. **Текстовая (псевдокод) – естественный, частично формализованный язык**

**Пр. нач ввод a,b,c (вещ.)**

**d = формула1**

**если d>=0 то нач**

**x1 = …**

**x2 = …**

**вывод x1,x2**

**кон**

**иначе вывод "Корней нет"**

**для x от 1 до 10 шаг 1 нц**

**кц**

**кон**

1. **Графическая (блок-схема) – шагам алгоритма соотв. геометрические фигуры, а посл-ть шагов задана связями между фигурами.**

**Стандартные элементы блок-схем, с.9-10**

|  |  |  |
| --- | --- | --- |
| **Название** | **Элемент** | **Комментарий** |
| **1 Процесс** |  | **Расчет (в прямоуг. формулами или словами)** |
| **2 Решение** |  | **Условный оператор, в ромбе – условие, 2 выхода** |
| **3 Модификация** |  | **Внутри записан закон изменения счетчика цикла:**  **x от 0 до 2 шаг 0.5**  **x = 0,0.5...2**  **x = 0;2;0.5**  **- д.б. известны границы изменения величины и шаг!** |
| **4 Документ** |  | **Вывод или печать данных (описаны внутри)** |
| **5 Ввод/вывод** |  | **Ввод или вывод данных (описано внутри)** |
| **6 Начало/конец** |  | **Начало и конец** |
| **Стрелки** |  | **Связь между блоками** |

**Существуют 3 базовых алгоритмич.структуры (со с. 11):**

* **Следование (линейный алгоритм)**
* **Ветвление (разветвляющийся алгоритм)**
* **Цикл**

**I У линейного алгоритма 3 основных оператора:**

1. **Ввод – позволяет задать исходные значения данных алгоритма. Обозн. – блок 5, ввод, read, readln**
2. **Присваивание – исп. для задания значений переменных, вид = или := ("присвоить")**

***Сначала берется (вычисляется) значение справа от знака, затем оно пишется в переменную слева от знака*.**

**Пр. k:=0 {В k записали 0}**

**k=k+1 {Значение k увеличили на 1}**

**c=a; a=b; b=c;**

**{значения a и b меняются местами}**

**r = sqrt ( sqr(x2-x1) + sqr(y2-y1) )**

**r = расстояния между точками (x1,y1) и (x2,y2)**

**sqr – квадрат, sqrt – квадратный корень**

**"стандартные функции" выполняют типовые вычисления. Обычно исп. с-му обозначений из pascal:**

|  |  |  |  |
| --- | --- | --- | --- |
| **sin(x)**  **cos(x)**  **tan(x)** | **Тригонометр.**  **функции** | **pow(x,y), x^y** | **X в степени Y** |
| **exp(x)**  **ln(x)** | **e в степени x, нат.логарифм** | **Sqr(x)**  **Sqrt(x)** | **Квадрат и**  **Квадр. корень** |
| **max(a,b)**  **min(a,b)** | **Максимум и минимум из значений** | **Round(x)**  **Floor(x) или trunc(x)**  **Ceil(x)**  **Frac(x)** | **Округлить**  **Округл.вниз**  **Округл.вверх**  **Дробная часть числа** |
| **Abs(x)** | **Абс.значение (модуль)** |  |  |

|  |  |  |
| --- | --- | --- |
| **Приоритет** | **Операции** | **Описание** |
| **1** | **\***  **/**  **div**  **mod или %** | **Умножить**  **Поделить**  **Делить нацело**  **Остаток от дел.** |
| **2** | **+**  **-** | **Сложение и вычитание** |

**2016 div 100 = 20 2016 mod 100 = 16**

**Следить за скобками! Н-р, a+b/2 – это не (a+b)/2**

1. **Вывод – отобразить результаты работы алгоритма. Обозн. – блок 4 или 5, вывод, write, writeln**

**II Ветвление.**

**Как правило, условия пишутся с помощью 6 операций отношения (ОО):**

**= или == != или <> >= <= < >**

**Равно не равно**

**ОО исп. только в виде АВ1 ОО АВ2 (между 2-мя арифм.выраж.)**

**(a<x) and (x<b) – верно**

**a<x<b - неверно**

**Составное условие может включить логические оп-ции AND, OR, NOT.**

**Пр. "x принадлежит [a,b]"**

**(x>=a) and (x<=b)**

**"x не принадлежит [a,b]"**

**not ( (x>=a) and (x<=b) ) или**

**(x<a) or (x>b)**

**"хотя бы одно из a,b,c положительно"**

**(a>0) or (b>0) or (c>0)**

**"заданы a,b,c , возможность построения треугольника со сторонами a,b,c"**

**(a<b+c) and (b<a+c) and (c<a+b)**

**Короткий условный оператор.**

**если условие то оператор;**

**if then**

**Если условие истинно, выполнится оператор, иначе он будет пропущен.**

**Пр. Если x и y – одного знака, вывести сообщ.**

**если x\*y>0 то вывод "Одного знака"**

**Пр. Если введено знач.<0, сообщить об ошибке и прервать**

**ввод x**

**если x<0 то нач**

**вывод "Ошибка"**

**halt (стоп)**

**кон**

**Полный условный оператор, всегда выбирает 1 из 2.**

**Если условие то оп1 иначе оп2**

**if then else**

**Пр. Проверить, принадлежит ли точка с коорд. (a,b) прямоугольнику с левым верхним углом (x1,y1) и правым нижним (x2,y2)**

**if (a>x1) and (a<x2) and (b>y2) and (b<y1)**

**then write 'принадлежит'**

**else write 'не принадлежит'**

**Пр. Корректно найти значение c=a/b**

**if b<>0 then c:=a/b**

**else write 'Деление на ноль!'**

**Составной условный оператор выбирает одну из нескольких ветвей алгоритма:**

**if условие1 then оператор1**

**else if условие2 then оператор2**

**else if условие3 then оператор3**

**else оператор4; {если все 3 условия ложны!}**

**Для точки, не лежащей на осях координат, определить номер ее координатной четверти**

**Ввод x,y**

**Если x>0 то нач**

**Если y>0 то n=1**

**Иначе n=4**

**Кон**

**Иначе нач**

**Если y>0 то n=2**

**Иначе n=3**

**кон**

**вывод n**

**III. Циклы.**

**Цикл – характеризуется повторением вычислений над некоторым набором данных. Числом повторений *тела цикла* управляет специальная переменная – *счётчик* или *управляющая переменная* (УП). На УП накладывается *условие*, определяющее, до каких пор выполняется цикл.**

|  |  |
| --- | --- |
| image001 | |
| **Цикл с предусловием – может не выполниться ни разу, если условие сразу же ложно**  **пока условие нц**  **тело цикла**  **кц**  **или**  **while условие do begin**  **тело цикла**  **end;** | **Цикл с постусловием – обязательно выполнится хотя бы раз**  **нц**  **тело**  **кц пока условие;**  **или**  **repeat**  **тело;**  **until условие;** |

**Обычно циклы взаимозаменяемы:**

|  |  |
| --- | --- |
| **x = 1**  **while x<=10 do begin**  **выч. y(x)**  **x = x + 1**  **end** | **x = 1**  **do begin**  **выч. y(x)**  **x = x + 1**  **end while x<=10;** |

**Оба этих цикла выполняются по 10 раз, УП x принимает значения 1,2,…,10:**

* **до цикла УП должна получить начальное значение;**
* **в условии задаётся конечное значение УП;**
* **в конце тела цикла УП должна изменяться.**

**В теле любого цикла должно быть обеспечено изменение УП, чтобы цикл мог завершиться!**

**Любой цикл с конечным числом шагов удобнее записать в виде *цикла с параметром*.**

**Цикл с параметром – форма записи цикла с предусловием, применяемая для:**

* **компактной записи закона изменения УП в заданных пределах;**
* **обработки векторов и матриц (списков и таблиц).**

**Формы записи закона изменения УП:**

|  |  |
| --- | --- |
| **X=1,1.5,…,10** | **X меняется от 1 до 10 включительно с шагом 0.5** |
| **Z=0; 100; 5** | **Z принимает значения 0, 5, …, 100** |
| **I=1..10**  **I=1;10** | **Если шаг не указан, он равен 1** |

**На псевдокоде:**

**для УП от НачЗн до КонЗн шаг Шаг нц**

**тело**

**кц**

**или**

**for i:=1 to n do begin**

**тело**

**end**

**Чаще всего наборы данных описаны как массивы. Массив – упорядоченный набор однотипных переменных (элементов).**

**Пр. в математике**

**1-мерный массив = вектор (список),**

**2-мерный массив = матрица (таблица)**

**Элементы между собой различаются по номерам.**

**Номер переменной в массиве – её индекс. Обычно индексы нумеруются с 1.**

**Пр. Оценки 20 студентов за экзамен?**

**a : array [1..20] of integer;**

**имя массив нумерация целочисленный**

**float или real – вещественный**

**string – строковый**

**Пр. Координаты 10 точек на плоскости?**

**1) x,y : array [1..10] of real;**

* **т.к. величины м.б. не только целочисленными**

1. **P : array [1..10,1..2] of real;**

* **так тоже возможно; массив P состоит из 10 элементов, каждый элемент – это набор из 2 значений (X- и Y-координаты очередной точки).**

**Взять из массива одну величину (элемент) можно записью вида a[1] или a1 "1-й эл-т массива a"**

**b[2,3] или b2,3 "эл-т матрицы b из 2-й строки и 3-го столбца"**

**Все действия с массивами выполняются в цикле с параметром! (for, для) Поэлементно!**

**Пр.1. Пользователь вводит 5 эл-тов массива A**

**i = 1,2,…,5**

Ввод A[i]

**Для I := 1 до 5 выполнять ввод A[i]**

**Пр.2 В массиве A – 20 оценок за экз. Найти средний балл**

**S = 0 {искомый средний балл обозначили S}**

**для n от 1 до 20 шаг 1 нц**

**S = S + A[i] {к сумме прибавили очеред.оценку}**

**кц**

**S = S / 20 {Поделили сумму на 20 – средний балл}**

**Типовые алгоритмы – задают правила выполнения типовых расчётов.**

1. **Табулирование – составление списков или таблиц**
2. **Определить УП цикла и задать ей нач.значение – левая граница интервала её изменения**
3. **Пишется цикл с предусловием вида УП<=КЗ, где КЗ – правая граница интервала**
4. **В теле цикла вычисляется и формируется 1 строка таблицы**
5. **Последний оператор тела цикла – увеличение УП на шаг**

**Пр. Составить таблицу стоимости 1,2,…,K единиц товара при цене за единицу = Z руб.**

**ввод K,Z**

**i:=1 {A, УП – кол-во ед.товара}**

**пока i<=K нц {B}**

**стоимость := i\*Z**

**вывод i,стоимость {C}**

**i := i+1 {D}**

**кц**

1. **Вычисление суммы, количества или произведения**
2. **Для каждой искомой величины предусмотреть по переменной того же типа, что исходные данные (сумма целых – целое, вещественных – вещественное);**
3. **До цикла присвоить начальные значения всем величинам, которые в нём ищутся:**

**Сумме или количеству 0, произведению 1**

**С. В теле цикла (как в алгоритме 1), если выполнено условие подсчета, вычисляем по правилам:**

**Сумму: S := S + T**

**Произведение: P := P \* T**

**Количество: K := K + 1**

**где T – очередной элемент данных**

**Пр. В массиве A[10] определить, каких элементов больше – положительных или отрицательных**

**kp := 0; ko := 0 {A-B}**

**for i := 1 to 10 do begin {C}**

**if A[i]>0 then kp := kp + 1;**

**if A[i]<0 then ko := ko + 1;**

**end;**

**if kp>ko then write ('положительных больше')**

**else if kp<ko then write ('отрицат. больше')**

**else write ('поровну');**

1. **Поиск максимума или минимума**
2. **Для каждого искомого максимума или минимума задать по переменной того же типа, что данные**
3. **До цикла максимуму присвоить:**

* **1-й элемент данных;**
* **заведомо малое значение, например, -1030**

**минимуму присвоить:**

* **1-й элемент данных;**
* **заведомо большое значение, например, 1030**

**С) В теле цикла макс. ищется оператором вида**

**if t>max then max := t ;**

**мин. ищется оператором вида**

**if t<min then min := t ;**

**где t – очередной элемент данных**

**Пр. Для функции y(x) = sin x2 при x, принимающем значения [-1,-0.95,…,1] найти максимальное по модулю значение**

**max := -1030 {A-B}**

**x := -1**

**while x<=1 do begin**

**y := sin(sqr(x))**

**if abs(y)>max then max:=abs(y);**

**x:=x+0.05;**

**end;**