


Рассмотрен

цикловой комиссией по общепрофессиональным дисциплинам и профессиональным модулям отделения «Электрификация и автоматизация сельского хозяйства»


Согласовано

зам. директора по ОМР

 Е. А. Ткаченко
«30» августа 2017 г.

Протокол № 1 от «30» августа 2017 г.

Председатель комиссии:

 Т. В. Невзорова

Методические рекомендации
по организации внеаудиторной самостоятельной
работы студентов по учебной дисциплине:

**ОП. 05 Основы программирования и баз
данных**

Специальность 09.02.02. Компьютерные сети

Грязовец
2017 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Целью самостоятельной работы студентов при изучении дисциплины ОП.05 «Основы программирования и баз данных» является научить студентов составлять алгоритмы и программы достаточно сложных задач.

По дисциплине «Основы программирования и баз данных» под самостоятельной работой подразумевается выполнение комплекса индивидуальных заданий.

Требования к оформлению самостоятельных работ

1. Каждая самостоятельная работа оформляется отдельно.
2. Отчет выполняется на листах формата А4 (210x297), которые подшиваются в папку.
3. Отчет обязательно содержит тему, цель и номер варианта самостоятельной работы.
4. Дальнейшее содержание отчета корректируется в каждом задании отдельно.
5. Каждый студент выполняет индивидуальные задания по варианту в соответствии со списком в журнале учебных занятий.

Организационные требования по изучению дисциплины

1. Иметь собственный конспект.
2. Помнить, что отсутствие на занятиях или невыполнение практических и лабораторных работ приравнивается к неудовлетворительной оценке, которая должна быть исправлена устным ответом (если это была лекция) или выполнением практической или лабораторной работы.
3. Иметь положительные ответы при опросах.
4. Выполнить все самостоятельные работы.

Самостоятельная работа по заданной теме выполняется в течение недели после проработки данной темы на занятиях. При сдаче работы на неделю позже оценка снижается на балл. При более поздней сдаче работы оценка не может быть выше «3» .

При оценке самостоятельной работы учитывается правильность выполнения задания, грамотность и аккуратность.

Методические указания к выполнению самостоятельной работы

Наиболее распространенными являются две формы описания алгоритма: словесное на естественном языке и в виде структурных схем или блок-схем. При втором способе записи алгоритм представляется в виде последовательности специальных символов-блоков, каждому из которых соответствует определенный этап решения задачи. В таблице даны обозначения некоторых блоков в соответствии с ГОСТом 19002-80, 19003-80.

Наименование символов	Обозначение символов	Примечания
Процесс		Арифметический блок, определяющий действия которые необходимо выполнить.
Предопределенный процесс		Обращение к подпрограммам
Принятие решения		Логический блок проверяющий истинность или ложность некоторого условия
Передача данных		Ввод и вывод информации
Терминатор (Прерывание)		Начало, конец, пуск, остановка, вход в программу
Модификация		Организация циклического процесса
Соединитель		Указание связи между прерванными линиями потока информации в пределах одной страницы.
Межстраничный соединитель		Указание между частями схемы, расположенными на разных листах.

Блоки соединяются линиями потока информации. Внутри блоков записываются выполняемые действия. Линия определяет направление вычислений, причем сверху вниз и слева направо. Если необходимо отразить другое направление (снизу вверх и справа налево), то необходимо на линиях ставить стрелки. Блоки на схеме нумеруются цифрами, которые ставятся в разрыве верхней линии слева.

Практически любой сложный алгоритм представляет собой комбинацию трех типов структур: *линейного, разветвляющегося и циклического.*

Линейный алгоритм состоит из последовательности операций, выполняющихся только один раз в порядке их следования.

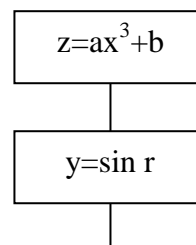
Примером линейного алгоритма может служить алгоритм вычисления значения функции $y = \sin((ax^3 + b)^2 + \ln(ax^3 + b))$

Процесс решения задачи можно разбить на этапы и записать алгоритм на:

1) естественном языке

1. Вычислить $z = ax^3 + b$
2. Вычислить $s = z^2$
3. Вычислить $t = \ln z$
4. Вычислить $r = s + t$
5. Вычислить $y = \sin r$

2) в виде блок-схемы



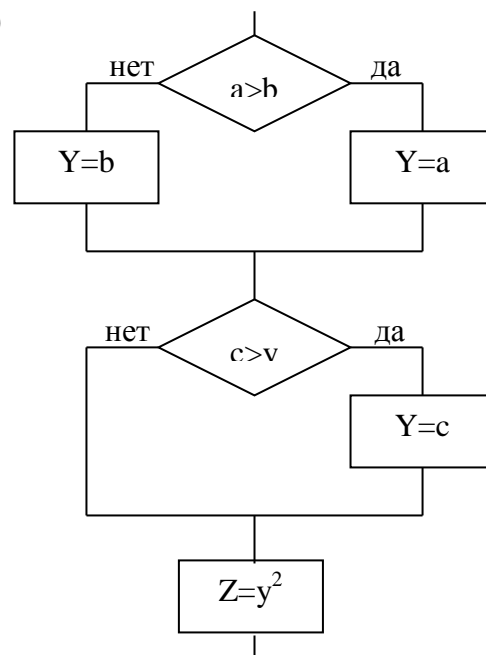
Разветвляющийся алгоритм содержит блок или блоки проверки некоторого условия, и в зависимости от результата проверки выполняются та или иная последовательность операций, называемая ветвью. При этом форма разветвления: может быть как, полная так и сокращенная

Рассмотрим в качестве примера алгоритма нахождения квадрата наибольшего из трех заданных чисел: a, b, c. Вначале сравниваются два числа: a и b. Большее из них принимается за максимальное. Затем производится сравнение полученного результата с третьим числом c. Если значение c оказывается больше, то оно принимается за максимальное и возводится в квадрат. В противном случае наибольшим считается результат сравнения чисел a и b.

Опишем алгоритм на естественном языке и в виде блок-схемы.

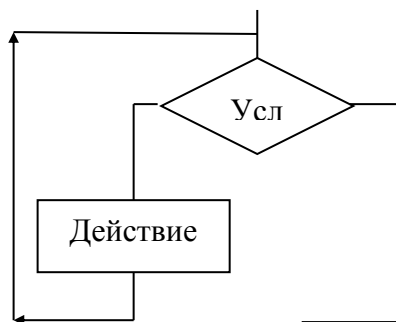
- 1) 1. Сравним a и b. Если $a > b$, то $y = a$.
В противном случае $y = b$.
2. Сравним c и y. Если $c > y$, то $y = c$.
В противном случае y оставить без изменения.
3. Вычислить $z = y^2$.

2)



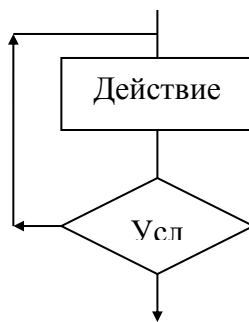
Циклический алгоритм содержит некоторую последовательность операций, выполняемую многократно. Любой циклический алгоритм содержит несколько типовых блоков. Основной блок, называемый телом цикла, производит требуемые вычисления. Остальные блоки имеют вспомогательное значение, они организуют циклический процесс; устанавливают начальные и новые значения данных, проверяют условия окончания или продолжения циклического процесса. Различают три типа структур цикла: **цикл с предусловием** (а), **цикл с постусловием** (б) и **цикл с параметром** (в).

а)



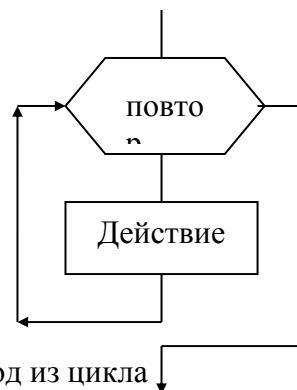
выход из цикла

б)



выход из цикла

в)



выход из цикла

Циклический алгоритм позволяет компактно описать большое число одинаковых вычислений над разными данными для получения необходимого результата.

Для примера рассмотрим алгоритм Евклида, формулируемый следующим образом: для двух любых положительных чисел найти их наибольший общий делитель.

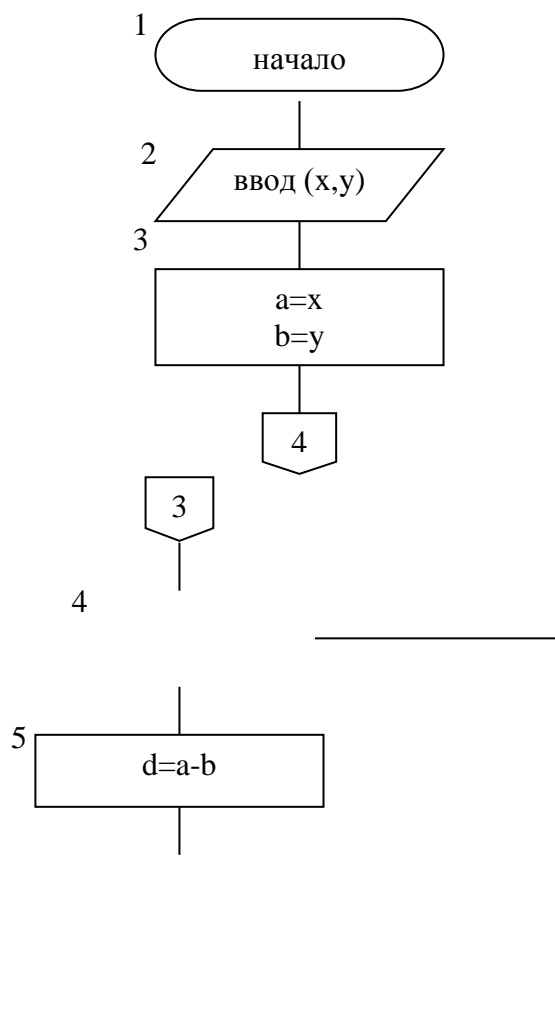
Для решения задачи необходимо получить убывающую последовательность чисел. Если первое число больше второго, то оно уменьшается на величину второго, в противном случае второе число уменьшается на величину первого. И так до тех пор, пока разность между этими числами не станет равной нулю. Тогда одно из этих и будет наибольшим делителем.

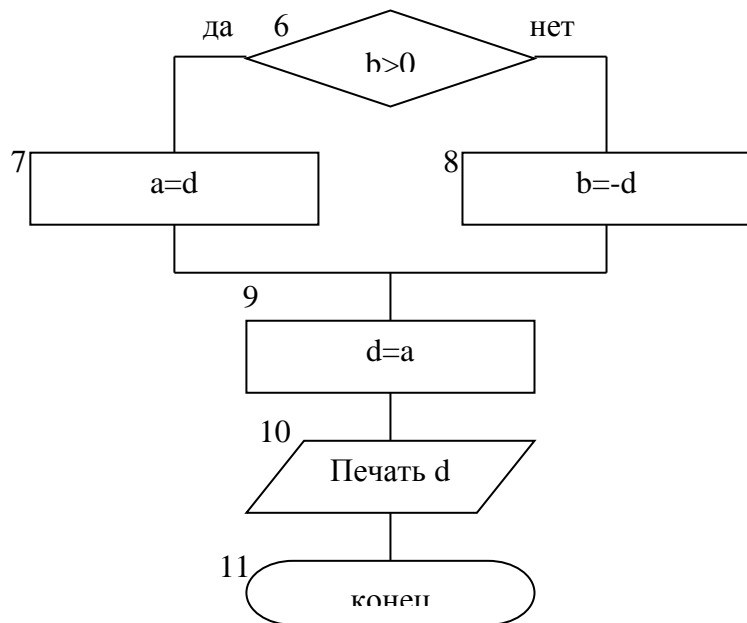
Опишем алгоритм: даны два числа $x > 0$, $y > 0$. Требуется найти наименьший общий делитель d .

1) на естественном языке

1. Принять $a=x$, $b=y$.
2. Сравнить a и b . Если $a=b$, то принять за наибольший общий делитель $d=a$ и закончить вычисления. В противном случае перейти к пункту 3.
3. Принять $d=a-b$.
4. Если $d > 0$, то $a=b$ в противном случае принять $b=-d$. Выполнить шаг два

2) в виде блок-схемы





Алгоритм Евклида содержит два линейных участка, одно разветвление и один цикл, который включает блоки 4 и 5-8.

При составлении алгоритма решения задачи следует руководствоваться следующими **правилами построения схемы алгоритма задачи (основные принципы алгоритмизации):**

1. Выявить исходные данные, результаты, назначить им имена.
2. Выбрать метод (порядок) решения задачи.
3. Разбить метод решения задачи на этапы (с учетом возможностей ЭВМ).
4. Изобразить каждый этап в виде соответствующего блока схемы алгоритма и указать стрелками порядок их выполнения.
5. В полученной схеме при любом варианте вычислений:
 - предусмотреть выдачу результатов или сообщений об их отсутствии;
 - обеспечить возможность после выполнения любой операции так или иначе перейти блоку **Прерывание** (к выходу схемы).

Отладкой алгоритма называется процесс выявления и исправления ошибок в нем. Суть отладки алгоритма в том, что выбирается некоторый набор исходных данных, называемый *тестовым набором (тестом)*, и задача с этим набором решается дважды: один раз – исполнением алгоритма, второй раз – каким-либо иным способом, исходя из условия задачи (вручную). **При совпадении результатов алгоритм считается верным.** В качестве тестового набора можно выбрать любые данные, которые позволяют:

- обеспечить проверку выполнения *всех* операций алгоритма;
- свести объем вычислений к минимуму.

Рекомендации по решению задач обработки массивов (матриц)

1. При решении задач, оперирующих с матрицами большого или переменного размера рекомендуется процесс решения задачи прослеживать вначале на небольших матрицах конкретного размера и составлять алгоритм именно для таких матриц с последующим обобщением его на случай матриц заданного размера.
2. Специфика рассматриваемых задач: в них в качестве подзадач выступают обычно операции по обработке всей матрицы в целом, части матрицы, столбца, строки и т.д., т.е.

набор их невелик. Из этого набора и следует выбирать наиболее крупные операции на каждом шаге решения задачи.

3. Для решения матричной задачи часто главное — определить последовательность вложенных подзадач убывающей сложности. В этом случае полезно бывает начинать с выбора не самой крупной операции, а, наоборот, с самой мелкой и искать операции возрастающей сложности.

Задание

Тема Линейные и разветвляющиеся алгоритмы.

Цель занятия

1. Освоение основных принципов алгоритмизации.
2. Получение навыков составления алгоритмов указанных видов.

Постановка задачи По заданию варианта составить блок-схему алгоритма.

Содержание отчета

1. Постановка задачи.
2. Блок-схема алгоритма.
3. Тестовый набор и результаты отладки алгоритма.
4. Выводы.

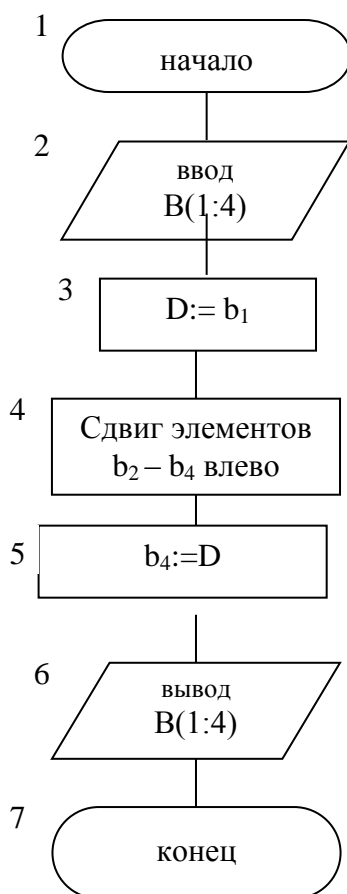
Методические указания

Задан массив $V(1:4)$. Каждому элементу массива присвоить значение соседнего с ним справа. Последнему элементу присвоить значение первого.

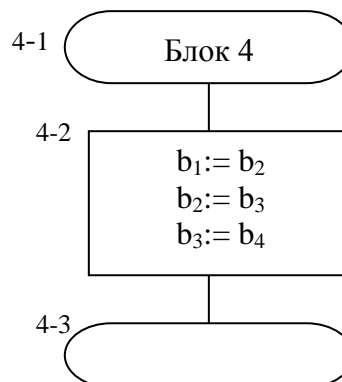
Исходные данные: массив $V(1:4)=\{b_1, b_2, b_3, b_4\}$.

1. Необходимо ввести в рассмотрение дополнительную переменную D и первая операция будет такая: $D:=b_1$.
2. Значения элементов $b_2 - b_4$ переносим на место элементов $b_1 - b_3$ соответственно.
3. Последняя операция – присвоим значение b_1 в b_4 , т.е. $b_4:=D$.

Все сказанное можно изобразить следующей схемой. Это - укрупненная схема алгоритма задачи.



Далее для каждой подзадачи отдельно составляется схема алгоритма ее решения. В нашей задаче в этом нуждается лишь задача блока 4. Процесс решения указанной задачи можно изобразить в виде схемы:



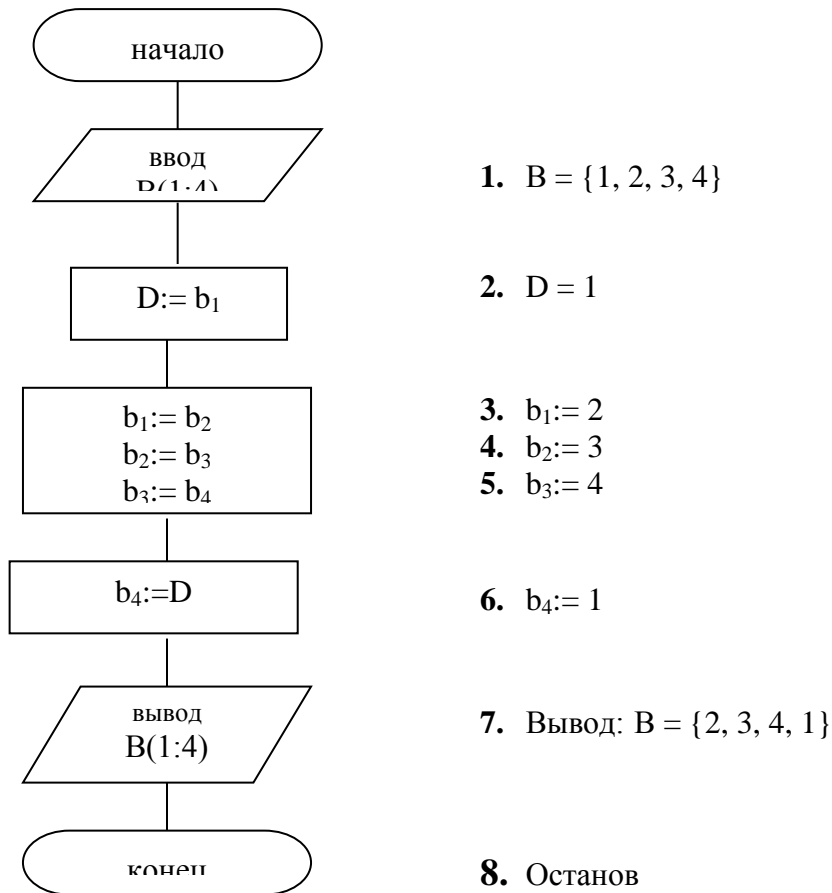
Последний блок (4-3) такой схемы оставляем незаполненным, так как здесь он обозначает не прекращение вычислений, а завершение некоторого участка схемы алгоритма.

Далее объединяем схему алгоритма решения всей задачи со схемой блока 4.

В результате получим подробную схему алгоритма всей задачи. Рядом со схемой алгоритма приведено исполнение алгоритма, приняв при этом в качестве тестового набора следующие значения элементов массива: $B(1:4) = \{b_1, b_2, b_3, b_4\} = \{1, 2, 3, 4\}$.

Анализ результатов работы алгоритма говорит о том, что алгоритм правильно решает задачу.

Исполнение алгоритма:



Составление разветвляющихся алгоритмов выполняется в соответствии с основными принципами алгоритмизации. Здесь работает та же схема:

- формулировка задачи
- метод
- алгоритм.

Варианты задания (линейные алгоритмы)

1. В списке учащихся из 20 человек первую по списку фамилию поставить на последнее место. В результате второй элемент списка станет первым, третий элемент – вторым, и т.д. Здесь список можно рассматривать как одномерный текстовый массив.
2. В матрице $B(1:3, 1:3)$ определить суммы элементов первой строки и последнего столбца.
3. В массиве $A(1:3, 1:3)$ элементы главной диагонали поставить на место соответствующих элементов третьей строки и определить сумму угловых элементов ($a_{11}, a_{13}, a_{31}, a_{33}$).
4. В массиве $F(1:4, 1:3)$ определить сумму S элементов первой строки и каждый элемент второго столбца умножить на S .
5. Имеется 15 клеток с кроликами, расположенных в ряд. В каждой клетке один кролик. Требуется пересадить каждое животное в соседнюю слева клетку, а из первой клетки пересадить в последнюю. Размеры клетки не позволяют помещать в одну клетку более одного кролика. Здесь ряд клеток можно рассматривать как одномерный массив.
6. В массиве $C(1:4, 1:4)$ элементы первой строки поставить на место соответствующих элементов главной диагонали и определить сумму элементов главной диагонали.
7. В матрице $B(1:3, 1:3)$ определить суммы элементов последней строки и первого столбца.
8. Первый столбец в матрице $D(1:2, 1:6)$ – это количество выпускников в какой-либо школе города, второй столбец матрицы – количество медалистов в данной школе. Определить общее число медалистов в городе и процентное отношение числа медалистов к общему числу выпускников в каждой школе города.
9. В массиве $F(1:4, 1:6)$ определить сумму S элементов первого столбца и каждый элемент второй строки умножить на S .
10. В массиве $D(1:5, 1:5)$ элементы второго столбца поставить на место соответствующих элементов главной диагонали и определить сумму квадратов элементов главной диагонали.
11. В массиве $A(1:3, 1:3)$ элементы главной диагонали поставить на место соответствующих элементов первой строки и определить сумму угловых элементов ($a_{11}, a_{13}, a_{31}, a_{33}$).
12. В матрице $B(1:3, 1:3)$ умножить элементы первого столбца на N (образовать массив $R(1:3)$) и вычислить значение S по формуле $S=b_{12}*r_1+b_{22}*r_2+b_{32}*r_3$.
13. В массиве $F(1:2, 1:3)$ определить сумму S элементов второй строки и каждый элемент третьего столбца умножить на S .
14. В массиве $F(1:5, 1:5)$ элементы четвертой строки поставить на место соответствующих элементов главной диагонали и определить произведение элементов главной диагонали.
15. В массиве $C(1:3, 1:3)$ элементы главной диагонали поставить на место соответствующих элементов первого столбца и определить сумму угловых элементов ($c_{11}, c_{13}, c_{31}, c_{33}$).
16. В массиве $D(1:3, 1:4)$ определить сумму R элементов первой строки и каждый элемент второго столбца умножить на R .
17. В матрице $B(1:5, 1:5)$ определить суммы элементов второй строки и последнего столбца.
18. В массиве $D(1:4, 1:4)$ элементы первого столбца поставить на место соответствующих элементов главной диагонали и определить произведение элементов главной диагонали.
19. В матрице $A(1:3, 1:3)$ умножить элементы второго столбца на N (образовать массив $R(1:3)$) и вычислить значение S по формуле $S=a_{11}*r_1+a_{21}*r_2+a_{31}*r_3$.
20. В массиве $F(1:4, 1:6)$ определить сумму S элементов третьей строки и каждый элемент второго столбца умножить на S .
21. В массиве $D(1:4, 1:4)$ элементы третьего столбца поставить на место соответствующих элементов главной диагонали и определить сумму квадратов элементов главной диагонали.
22. В массиве $C(1:3, 1:3)$ элементы главной диагонали поставить на место соответствующих элементов третьего столбца и определить сумму угловых элементов ($c_{11}, c_{13}, c_{31}, c_{33}$).
23. В массиве $F(1:4, 1:6)$ определить сумму S элементов шестого столбца и каждый элемент четвертой строки умножить на S .

24. Первый столбец в матрице A(1:2, 1:4) – это количество выпускников в какой-либо школе города, второй столбец матрицы – количество медалистов в данной школе. Определить общее число медалистов в городе и процентное отношение числа медалистов к общему числу выпускников в городе.
25. В массиве F(1:5, 1:5) элементы четвертого столбца поставить на место соответствующих элементов главной диагонали и определить сумму квадратов элементов главной диагонали.
26. В массиве D(1:3, 1:4) определить сумму R элементов третьей строки и каждый элемент четвертого столбца умножить на R
27. В матрице B(1:3, 1:3) умножить элементы третьего столбца на N (образовать массив R(1:3)) и вычислить значение S по формуле $S = b_{12} * r_1 + b_{22} * r_2 + b_{32} * r_3$.
28. В массиве A(1:3, 1:3) элементы третьей строки поставить на место соответствующих элементов главной диагонали и определить сумму элементов главной диагонали.
29. В массиве D(1:4, 1:4) элементы четвертого столбца поставить на место соответствующих элементов главной диагонали и определить произведение элементов главной диагонали.
30. В массиве F(1:4, 1:6) определить сумму S элементов второй строки и каждый элемент пятого столбца умножить на S.

Варианты задания (разветвляющие алгоритмы)

Приведенные в условиях задач величины с индексами следует рассматривать как элементы массивов (одномерных или двумерных).

1. Вычислить значение функции Z при одном значении X: $Z = Y^2 + X^2$, где

$$Y = \begin{cases} |X| + 2, & \text{если } X < 3; \\ X^2 - a_1, & \text{если } X = 3; \\ X - C, & \text{если } 3 < X < 10; \\ a_2^2, & \text{если } 10 \leq X \end{cases}$$

2. Вычислить значение функции Z по одной из формул: $Z = \begin{cases} X - b_1, & \text{если } X < 6; \\ X^2 + b_2, & \text{если } X = 15 \text{ или } 20; \\ b_3 + X & \text{в остальных случаях.} \end{cases}$
3. Определить значение наибольшего элемента главной диагонали матрицы A(1:3, 1:3).
4. Вывести значения X_1, X_2, X_3 в порядке возрастания.
5. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} \ln |X + a^2|, & \text{если } X < 7; \\ \sqrt{|b_{11} + b_{21}|} + b_{22}, & \text{если } X = 10; \\ \frac{3.5b_{12}}{0.5|b_{11} \cdot b_{22}| + 2}, & \text{в остальных случаях.} \end{cases}$$

6. Определить значение наименьшего элемента седьмого столбца матрицы B(1:3, 1:7).
7. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} \sqrt[3]{\frac{\ln x + 5.5|a_1|}{1.5x}}, & \text{если } 15 < x < 21; \\ a_1 x^2 + a_2 x + a_3 & \text{в остальных случаях.} \end{cases}$$

8. Определить значение наибольшего элемента второй строки матрицы $C(1:3, 1:3)$.

9. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} 1.5 - \sqrt{|c_1|} + e^{c_2}, & \text{если } x < 10; \\ x \ln |c_1| 0.5 c_3^2, & \text{если } 15 < x \leq 25; \\ \text{вывести : " значение } Y \text{ в остальных} \\ \text{случаях не определено."} \end{cases}$$

10. Вычислить значения функций w и v :

$$w = \begin{cases} \frac{c_1 + c_3 + \sqrt{|c_1|}}{4x - 3b}, & \text{если знаменатель не равен нулю или } x < 5; \\ \ln |c_1 x + 2.5b|, & \text{в остальных случаях} \end{cases}$$

$$v = w^2 + c_1 x$$

11. Определить значение наименьшего элемента главной диагонали матрицы $B(1:3, 1:3)$.

12. Массив $B(1:3)$ – целый. Какому элементу массива равна величина D ? $D = a + a^2 + a^3$ (a – целая величина).

13. Вывести значения $X_1, X_2, X_3, X_4, X_5, X_6$ в порядке убывания.

14. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} \ln |X + a^2|, & \text{если } X < 7; \\ \sqrt{|b_{11} + b_{22}|} + b_{33}, & \text{если } X = 10; \\ \frac{5b_{13}}{|b_{11} \cdot b_{12}| + 4}, & \text{в остальных случаях.} \end{cases}$$

15. Вычислить значение функции Z по одной из формул: $Z = \begin{cases} b_1, & \text{если } X \leq 0; \\ X^2 + b_2, & \text{если } 0 < X \leq 1; \\ b_3 + X^4 & \text{в остальных случаях.} \end{cases}$

16. Вычислить значение функции Z при одном значении X : $Z = Y^2 + X^2$, где

$$Y = \begin{cases} |X| + 2a_3, & \text{если } X < 6; \\ X^2 - a_1, & \text{если } X = 6; \\ X - C, & \text{если } 6 < X < 20; \\ a_2^2, & \text{если } 20 \leq X \end{cases}$$

17. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} \ln |X + b_{21}^2|, & \text{если } X < 5; \\ \frac{b_{12}}{|b_{11} \cdot b_{22}| + 2}, & \text{в остальных случаях.} \end{cases}$$

18. Определить значение наименьшего элемента четвертого столбца матрицы $B(1:3, 1:4)$.

19. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} \ln x + 9a_1, & \text{если } x > 0 \\ -\frac{xa_2}{x^2a_3 - 7}, & \text{в остальных случаях} \end{cases}$$

20. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} x^2 + 3x + 9a_1, & \text{если } x \leq 3; \\ -\frac{a_2 \sin x}{x^2a_3 - 9}, & \text{в остальных случаях}; \end{cases}$$

21. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} \ln x + 9a_1, & \text{если } x > 0 \\ -\frac{xa_2}{x^2a_3 - 7}, & \text{в остальных случаях} \end{cases}$$

22. Вычислить значение Y по одной из заданных формул: $Y = \begin{cases} 9a_1 - x, & \text{если } x > 1.1 \\ -\frac{\sin 3xa_2}{x^4a_3 + 1}, & \text{если } x < 1.1 \end{cases}$

23. Определить значение наименьшего элемента третьей строки матрицы $B(1:3, 1:4)$.

24. Вычислить значения функций w и v :

$$w = \begin{cases} \frac{a_1 + a_3 + |a_2|}{4x - 3b}, & \text{если знаменатель не равен нулю или } x < 15; \\ \ln |a_1x + 2|, & \text{в остальных случаях} \end{cases}$$

$$v = w + (a_1 + a_2 + a_3) \cdot x$$

25. Вычислить значение Y по одной из заданных формул:

$$Y = \begin{cases} 5 - \sqrt{|c_1|}, & \text{если } x < 10; \\ x \ln |c_2| 0.5c_3^2, & \text{если } 25 < x \leq 35; \\ \text{вывести : " значение } Y \text{ в остальных} \\ \text{случаях не определено."} \end{cases}$$

26. Вычислить значение Y по одной из заданных формул: $Y = \begin{cases} 1.2x^2 - 3x - 9b_1, & \text{если } x > 3 \\ \frac{12.1b_2}{2x^2 + b_3 + 1}, & \text{если } x \leq 3 \end{cases}$

27. Вычислить значение Y по одной из заданных формул: $Y = \begin{cases} x^2 + 4x + 5a_1, & \text{если } x \leq 2 \\ \frac{a_2}{x^2 + 4x + 5a_3}, & \text{если } x > 2 \end{cases}$

28. Вычислить значение Y по одной из заданных формул: $Y = \begin{cases} -3x + 9b_1, & \text{если } x > 3 \\ \frac{x^3 + b_2}{x^2 + b_3 + 8}, & \text{если } x \leq 3 \end{cases}$

29. Вычислить значение Y по одной из заданных формул: $Y = \begin{cases} -x^3 + 9b_1, & \text{если } x \leq 13 \\ -\frac{3b_2}{b_3 + 1}, & \text{если } x > 13 \end{cases}$
30. Вычислить значение Y по одной из заданных формул: $Y = \begin{cases} 45x^2 + 5a_1, & \text{если } x > 3.6 \\ \frac{5x + a_2}{10x^2 + a_3 + 1}, & \text{если } x \leq 3.6 \end{cases}$

Задание

Тема Циклические алгоритмы

Цель занятия

1. Освоение основных принципов алгоритмизации.
2. Получение навыков составления алгоритмов указанных видов.

Постановка задачи По заданию варианта составить блок-схему алгоритма

Содержание отчета

1. Постановка задачи.
2. Блок-схема алгоритма.
3. Тестовый набор и результаты отладки алгоритма.
4. Выводы.

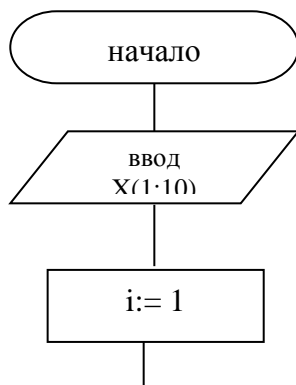
Методические указания

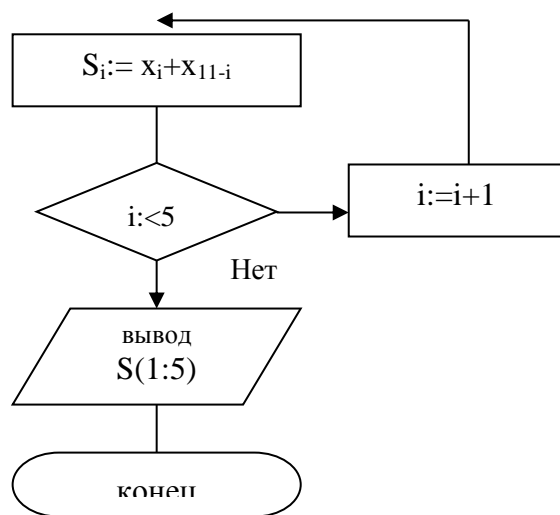
Задаана последовательность чисел: 3, 5, 6, 8, 9, 11, 15, 16, 20, 21. Вычислить суммы пар чисел: первого и десятого, второго и девятого, ..., пятого и шестого.

Исходные данные: X (1:10).

Результат: S (1:5).

1. Суммируем попарно элементы массива X в порядке, указанном в условии задачи: x_1 и x_{10} , x_2 и x_9 , и т.д.
2. Запишем операции, выполняемые на первом, втором и третьем этапах решения задачи и исходя из них выявим операцию i -го этапа. Затем запишем операцию последнего этапа.
 Этап 1. $s_1 = x_1 + x_{10}$
 Этап 2. $s_2 = x_2 + x_9$
 Этап 3. $s_3 = x_3 + x_8$
 Этап i . $s_i = x_i + x_{11-i}$ (*)
 Этап 5. $s_5 = x_5 + x_6$
3. Формула (*) и есть рабочая формула. В ней единственная независимая переменная – величина i . Выявим все параметры этой переменной: начальное значение – 1, конечное значение – 5, закон изменения – $i := i + 1$.
4. Изобразим блок-схему алгоритма:





Варианты задания

1. Матрицу $A(1:n, 1:m)$ умножить на k и найти максимальный элемент 2-го столбца.
2. Найти сумму матриц $A(1:n, 1:m)$ и $B(1:n, 1:m)$. Умножить каждый элемент 1-й строки матрицы A на соответствующий элемент последней строки матрицы B .
3. Найти сумму элементов каждой строки матрицы $A(1:n, 1:m)$ и максимальный элемент 2-й строки.
4. Определить наибольший элемент в каждом столбце матрицы $A(1:m, 1:k)$. Вычислить сумму элементов 2-го столбца.
5. Записать элементы матрицы $A(1:n, 1:m)$ в виде массива $B(1:n*m)$. Найти минимальный элемент 2-ой строки.
6. В матрице $A(1:n, 1:m)$ сдвинуть каждую строку, начиная со второй на одну вверх. Первую строку поставить на место последней. Найти разность максимального элемента 6-ой строки и минимального элемента 2-ого столбца.
7. Переставить строки матрицы $A(1:n, 1:m)$: первую с последней, вторую с предпоследней и т.д. Умножить третью строку на сумму элементов 5-го столбца.
8. Элементы матрицы $A(1:n, 1:m)$, большие 5 и меньшие 20 записать в массив $B(1:n*m)$. Вычислить разность соответствующих элементов первой и последней строки.
9. Образовать матрицу $A(1:n, 1:n)$, все элементы главной диагонали которой равны единице. Угловые элементы матрицы принять равными m , все остальные элементы – равными 10.
10. Элементы массива $B(1:k)$ расположить в порядке возрастания их значений. Вычислить произведение максимального элемента массива на сумму всех элементов массива.
11. Определить имеются ли равные элементы в массиве $B(1:k)$, и если имеются то заданный массив умножить на A .
12. Определить номера строк матрицы $A(1:n, 1:m)$, совпадающих с массивом $B(1:k)$. Последний столбец матрицы умножить на P .
13. В матрице $A(1:n, 1:m)$ определить наибольший элемент и умножить его на сумму элементов главной диагонали.
14. Определить наименьший элемент каждой четной строки матрицы $A(1:n, 1:m)$ (n - четное) и сумму этих элементов.
15. Переставить значения каждой пары элементов матрицы $A(1:n, 1:n)$, симметричных относительно главной диагонали. Найти максимальный элемент первого столбца.
16. Матрицу $A(1:n, 1:m)$ умножить на k и найти максимальный элемент 2-го строки.
17. Найти сумму матриц $A(1:n, 1:m)$ и $B(1:n, 1:m)$. Умножить каждый элемент 1-го столбца матрицы A на соответствующий элемент последнего столбца матрицы B .
18. Найти произведение элементов каждой строки матрицы $A(1:n, 1:m)$ и максимальный элемент 2-го столбца.
19. Определить наименьший элемент в каждом столбце матрица $A(1:m, 1:k)$. Вычислить сумму элементов 2-ой строки.

20. Записать элементы матрицы $A(1:n, 1:m)$ в виде массива $B(1:n*m)$. Найти максимальный элемент 2-ой строки.
21. В матрице $A(1:n, 1:m)$ сместить каждую строку, начиная со второй на одну вверх. Первую строку поставить на место последней. Найти разность максимального элемента 1-ой строки и минимального элемента последнего столбца.
22. Переставить строки матрицы $A(1:n, 1:m)$: первую с последней, вторую с предпоследней и т.д. Умножить первую строку на сумму элементов последнего столбца.
23. Элементы матрицы $A(1:n, 1:m)$, большие 5 и меньшие 20 записать в массив $B(1:n*m)$. Вычислить разность соответствующих элементов первого и последнего столбцов.
24. Образовать матрицу $A(1:n, 1:n)$, все элементы главной диагонали которой равны нулю. Угловые элементы матрицы принять равными 5, все остальные элементы – равными 9.
25. Элементы массива $B(1:k)$ расположить в порядке убывания их значений. Вычислить произведение минимального элемента массива на сумму всех элементов массива.
26. Определить имеются ли равные элементы в массиве $B(1:k)$, и если имеются то заданный массив умножить на 25.
27. Определить номера строк матрицы $A(1:n, 1:m)$, совпадающих с массивом $B(1:k)$. Последнюю строку матрицы умножить на P.
28. В матрице $A(1:n, 1:m)$ определить наименьший элемент и умножить его на сумму элементов главной диагонали.
29. Определить наибольший элемент каждой четной строки матрицы $A(1:n, 1:m)$ (n - четное) и сумму этих элементов.
30. Переставить значения каждой пары элементов матрицы $A(1:n, 1:n)$, симметричных относительно главной диагонали. Найти максимальный элемент первой строки.

Объектно-ориентированное программирование.

Задания:

Вопрос 1. Каков будет результат выполнения программы

```

1. public class Test {
2.     public Test() {
3.     }
4.
5.     public static void main(String[] args) {
6.         Test test = new Test();
7.         int i = 5;
8.         while(i = 5){
9.             System.out.println(i++);
10.        }
11.    }
12. }
```

Варианты ответов

1. а) Компилятор выдаст сообщение об ошибке в строке 8
2. б) На консоль будут последовательно выведены значения 01234
3. в) На консоль будут последовательно выведены значения 43210
4. г) Программа откомпилируется, но на консоль ничего выведено не будет

Правильный ответ а). В операторе while может быть использовано только булево значение. В данном случае используется оператор присваивания, а не сравнения, таким образом, компилятор выдаст ошибку.

Вопрос 2. Приведенная ниже программа должна вывести на консоль Hello World! Выберите строки, которые нужно модифицировать в программе, что бы получить правильный результат.

```
1. public class Test {
2. public Test() {
3. }
4. public static void main(String[] args) {
5. Test test = new Test();
6. String [] arr = {"H", "e", "l", "l", "o", " ",
7. "w", "o", "r", "l", "d", "!"};
8. String result = "";
9. int i= 0;
9. for(;;){
10. result += arr[i++];
11. }
12. System.out.println(result);
13. }
14. }
```

Варианты ответов

1. а) Заменить строку 9 на `for (i = 0; i < arr. length ;){`
2. б) Заменить строку 9 на `for(int int i = 0; i < arr.length;){`
3. в) Заменить строку 9 на `for(i = 0; i < arr.length;i++){`
4. г) Заменить строку 9 на `for(i = 1; i <= arr.length;i++){`

Правильный ответ а). Ответ б) неверен, так как переменная `i` уже определена в методе `main`. Область видимости переменной (если она объявлена в цикле `for`) будет ограничиваться лишь телом цикла. Однако в данном случае переменная с таким именем уже объявлена в теле метода и соответственно находится в той же области видимости. Если переменная была бы объявлена, как переменная класса, то этот код откомпилировался бы вполне успешно. Ответ в) не является верным так как, увеличение значения `i` в теле цикла будет произведено дважды, т.о. на печать будут выведены лишь четные элементы массива. Ответ г) не является верным по двум причинам. Первая – элементы массива нумеруются с 0, соответственно первым будет выбран второй элемент массива, вторая – когда выполнится условие окончания цикла, будет нарушена граница массива и будет вызвано исключение `IndexOutOfBoundsException`.

Вопрос 3. Какая строка будет выдана на консоль после выполнения фрагмента кода приведенного ниже.

```
1. public class Test {
2. public Test () {
3. }
4. public static void main ( String [] args ) {
5. int i, j ;
6. lab : for ( i = 0; i < 6; i++){
7. for (j = 3; j > 1; j--){
8. if(i == j){
9. System.out.println(" " + j);
10. break lab;
11. }
12. }
13. }
14. }
15. }
```

Варианты ответов

1. а) 2345
2. б) 234
3. в) 3
4. г) 2

Правильный ответ б). Условие if в данном примере будет выполнено, когда переменные i и j будут равны 2. После чего на консоль будет выведено 2 и выполнится оператор break. Т.к. break содержит ссылку на метку, то будет прерван не текущий цикл (внутренний, по переменной j), а цикл по переменной i (внешний), т.о. выполнение программы будет прекращено.

Вопрос 4. Какой результат следует ожидать при компиляции и запуске приведенного кода:

```
String str=new String("Java");  
int i=1;  
char j=3;  
System.out.println(str.substring(i,j));
```

Варианты ответов

1. а) Выведено: Ja
2. б) Выведено: av
3. в) Выведено: ava
4. г) Ошибка: не существует метода substring(int,char).

Правильный ответ б). Числовые расширения используются для того, чтобы приводить операнды числовой операции к общему типу так, чтобы действие могло выполняться. В некоторых операциях, которые дают значения числового типа, одноместное числовое расширение применяется к отдельному операнду: если операнд имеет тип byte, short или char, одноместное числовое расширение расширяет его до значения типа int расширяющим преобразованием. Иначе, одноместный числовой операнд остается как есть и не преобразовывается.

Вопрос 5. Какой метод следует использовать, чтобы обнаружить позицию буквы v в строке str= "Java"?

Варианты ответов

1. а) mid (2, str)
2. б) str.charAt (2)
3. в) str.indexOf ('v')
4. г) indexOf (str,'v')

Правильный ответ в). Для решения этой или подобной задачи советую смотреть документацию по Java: <http://java.sun.com/javase/6/docs/api/>.

Вопрос 6. Что будет выведено в результате компиляции и запуска следующего кода:

```
String str =" ava "  
char ch ='J';
```

```
ch += str ;  
System.out.println ( ch );
```

Варианты ответов

1. а) Java
2. б) ava
3. в) avaJ
4. г) J
5. д) Ошибка во время компиляции.

Правильный ответ д). Не определена операция += для char String. И String не может привести к int и char к String.

Вопрос 7. Что будет результатом компиляции и выполнения следующего кода?

```
StringBuffer s= new StringBuffer("You Java");  
s.insert(2, "like ");  
System.out.print(s);
```

Варианты ответов

1. а) Yolike u Java
2. б) You like Java
3. в) Ylike ou Java
4. г) You Java like
5. д) Ошибка компиляции: метод insert() не объявлен для класса String-Buffer

Правильный ответ а). Класс StringBuffer является близнецом класса String, но, в отличие от последнего, содержимое и размеры объектов класса StringBuffer можно изменять. Метод insert в нём определён, индексирование для Java стандартное, т.е. с нуля.

Вопрос 8. Что будет выведено при попытке компиляции и запуска этой программы:

```
public class Quest6 {  
public static void main(String[] args){  
int a[] = new int[]{1,2,3,};  
System.out.print(a[1]);  
} }
```

Варианты ответов

1. а) Ошибка компиляции: не определен размер массива
2. б) Ошибка времени выполнения
3. в) Выведено 1
4. г) Выведено 2
5. д) Ошибка компиляции: неправильная инициализация

Правильный ответ г). Если ошибка синтаксиса должно было бы быть 1, если во времени выполнение непозволительные аргументы или с неопределённой ссылкой -2, в данном случае ошибок нет.

Вопрос 9. Что будет выведено при попытке компиляции и запуска программы?

```

public class Quest8{
    static int j=2;
    public static void result(int i){
        i *= 10;
        j += 2;
    }
    public static void main(String[] args){
        char i = '1';
        result(i);
        System.out.println(i+" "+j);
    } }

```

Варианты ответов

1. а) 1 2
2. б) 10 2
3. в) Ошибка: параметр метода result() не сочетается с передаваемой переменной
4. г) 10 4
5. д) 1 4

Правильный ответ д). Ответ в) неверен, т.к. char приводится к int (в этом его особенность и уникальность). Символ «1» переводится к числу «1», к j прибавляется функции result ещё 2 (итого $2 + 2 = 4$). Переменная j объявлена как static, но это означает лишь то, что она, как любая другая static переменная, является единственной для всех объектов данного класса, но изменять её значение можно.

Вопрос 10. Что будет выведено при компиляции и запуске кода?

```

public class Quest {
    {System.out.print("1");}
    static{System.out.print("2");}
    Quest(){System.out.print("3");}

    public static void main(String[] args) {
        System.out.print("4");
    } }

```

Варианты ответов

1. а) 34
2. б) 24
3. в) 14
4. г) 4
5. д) 1234
6. е) 234

Правильный ответ б). При запуске приложения выполняются только статические логические блоки класса приложения и тело метода main().

Вопрос 11. Какие из следующих утверждений истинные?

Варианты ответов

1. а) Частные методы не могут быть перегружены

2. б) Переопределенный метод не может включать исключения не обрабатываемые в базовом классе
3. в) Методы, объявленные как final, не могут быть переопределены
4. г) Статические методы не могут быть переопределены

Правильный ответ в).

Вопрос 12. Что произойдет в результате компиляции и запуска кода?

```
class Base {}
class A extends Base {}
public class Quest{
    public static void main(String[] args){
        Base b = new Base();
        A ob = (A) b;
    } }
```

Варианты ответов

1. а) Ошибка во время выполнения
2. б) Ничего: компиляция и выполнение без ошибок
3. в) Ошибка во время компиляции

Правильный ответ в). Во время выполнения при попытке преобразования с сужением типа будет сгенерировано исключение ClassCastException.

Вопрос 13. Что произойдет в результате компиляции и запуска кода?

```
abstract class QuestBase {
    abstract void show();
    static int i;
}
public class Quest2 extends QuestBase {
    public static void main(String[] args){
        boolean[] a = new boolean[3];
        for(i = 0; i < a.length; i++)
            System.out.print(" " + a[i]);
    } }
```

Варианты ответов

1. а) Ошибка времени компиляции: Quest2 должен быть объявлен как abstract
2. б) Ошибка времени выполнения: IndexOutOfBoundsException
3. в) будет выведено: true true true
4. г) будет выведено: false false false
5. д) Ошибка: массив a использован прежде, чем проинициализирован

Правильный ответ а). Код не будет откомпилирован, так как подкласс наследует абстрактный класс и при этом не реализует его абстрактный метод и сам не объявлен как абстрактный.

Вопрос 14. Какие из объявлений корректны, если:

```
class Outer {
    class Inner {
    } }
```

Варианты ответов

1. а) new Outer.Inner ()
2. б) Outer.new Inner ()
3. в) new Outer.new Inner ()
4. г) new Outer().new Inner()
5. д) Outer.Inner()
6. е) Outer().Inner()
7. ж) Ни одно из приведенных.

Правильный ответ г). Объявить объект внутреннего (нестатического) класса можно, только предварительно создав объект внешнего класса. Конструкторы обоих классов должны вызываться так же, как и для всех других классов, т.е. с помощью оператора new.

Вопрос 15. Что будет выведено в результате компиляции и выполнения следующего кода:

```
abstract class Abstract {
    abstract Abstract meth ();
}
class Owner {
    Abstract meth () {
        class Inner extends Abstract {
            Abstract meth() {
                System.out.print("Inner ");
                return new Inner();
            }
        }
        return new Inner();
    }
}
public abstract class Quest4 {
    public static void main(String a[]) {
        Owner ob = new Owner();
        Abstract abs = ob.meth();
        abs.meth();
    }
}
```

Варианты ответов

1. а) Inner
2. б) Inner Inner
3. в) Inner Inner Inner
4. г) Compile time error
5. д) Runtime error
6. е) Ошибка компиляции из-за двойного объявления meth() в классе Owner

Правильный ответ а). В результате выполнения кода Owner ob = new Owner(); будет создан объект Owner. Его метод meth() создаст объект типа Inner в результате выполнения кода Abstract abs = ob.meth(); При его выполнении на консоль ничего выведено не будет, так как метод meth() класса Inner, выводящий на консоль строку inner, будет вызван только один раз командой abs.meth().

Вопрос 16. Что будет выведено в результате компиляции и выполнения этого кода?

```
import java.awt.*;
public class Quest2 extends Frame{
    Quest2() {
```

```

Button yes = new Button("YES");
Button no = new Button("NO");
add(yes);
add(no);
setSize(100, 100);
setVisible(true);
}
public static void main(String[] args){
    Quest2 q = new Quest();
} }

```

Варианты ответов

1. а) Две кнопки рядом, занимающие весь фрейм, YES слева и NO справа
2. б) Одна кнопка YES, занимающая целый фрейм
3. в) Одна кнопка NO, занимающая целый фрейм
4. г) Две кнопки наверху фрейма, YES и NO

Правильный ответ в). По умолчанию у Frame менеджер компоновки BorderLayout, а у BorderLayout, если не указан второй аргумент, то элемент является BorderLayout.CENTER.

Вопрос 17. Какой менеджер компоновок размещает компоненты в таблице с ячейками равного размера?

Варианты ответов

1. а) FlowLayout
2. б) GridLayout
3. в) BorderLayout
4. г) CardLayout

Правильный ответ б). FlowLayout – менеджер поточной компоновки. CardLayout создает вкладки, содержимое которых отображается при выполнении щелчка на заголовке. BorderLayout позволяет позиционировать элементы в областях фиксированного размера, граничащих со сторонами фрейма, которые обозначаются параметрами: NORTH, SOUTH, EAST, WEST. Остальное пространство обозначается как CENTER. GridLayout разделяет фрейм на заданное количество рядов и колонок.

Вопрос 18. Какое выравнивание устанавливается по умолчанию для менеджера размещений FlowLayout?

Варианты ответов

1. а) Указывается явно
2. б) FlowLayout.RIGHT
3. в) FlowLayout.LEFT
4. г) FlowLayout.CENTER
5. д) FlowLayout.LEADING

Правильный ответ в).

Вопрос 19. Сколько кнопок будет выведено в апплет:


```
import java.applet.*;
import java.awt.*;
public class Quest4 extends Applet{
    Button b = new Button("Yes");
    public void init(){
        add(b);
        add(b);
        add(b);
        add(new Button("No"));
        add(new Button("No"));
        add(new Button("No"));
    }
}
```

Варианты ответов

1. а) 1 кнопка с надписью "Yes" и 1 кнопка с надписью "No"
2. б) 1 кнопка с надписью "Yes" и 3 кнопки с надписью "No"
3. в) 3 кнопки с надписью "Yes" и 1 кнопка с надписью "No"
4. г) 3 кнопки с надписью "Yes" и 3 кнопки с надписью "No"

Правильный ответ б). Команда add(b), вызванная во второй раз, пытается добавить на апплет уже существующий там объект. Команда add(new Button("No")) каждый раз добавляет новый объект.

Вопрос 20. Объект JCheckBox объявлен следующим образом:

```
JCheckBox ob = new JCheckBox();
```

Какая из следующих команд зарегистрирует его в блоке прослушивания событий?

Варианты ответов

1. а) addItemListener(this)
2. б) addItemListener()
3. в) ни одна из приведенных
4. г) ob. addItemListener ()
5. д) ob. addItemListener (this)

Правильный ответ д). Метод всегда вызывается объектом, который необходимо зарегистрировать. В качестве параметра должен передаваться объект приложения или апплета, в котором размещается данный компонент, если он реализует соответствующий интерфейс.

Вопрос 21. Что будет выведено в результате компиляции и выполнения следующего кода?

```
import java.awt.*;
class Quest5 extends Thread implements Runnable{
    public static void main(String[] args) {
        Component btn = new Button(" Кнопка 1");
        System.out.println(((Button) btn).getLabel());
    }
}
```

Варианты ответов

1. а) Кнопка 1
2. б) Ошибка компиляции: Quest5 должен наследоваться от Applet
3. в) Ошибка компиляции: не определен метод run() интерфейса
4. г) Ошибка компиляции: ссылка на Component не может быть инициализирована объектом Button
5. д) Ошибка времени выполнения
6. е) Ничего не будет выведено

Правильный ответ а). Объекты из пакета AWT могут объявляться и вызывать свои методы из любых приложений.

Вопрос 22. Какой код читает параметр count в переменную i в этом фрагменте кода?

```
< applet code=MyApplet.class width=200 height=200>  
< param name=count value=5>  
< /applet>
```

Варианты ответов

1. а) int i = getIntParameter("count")
2. б) int i = getParameter("count")
3. в) int i = new Integer(getIntParameter("count")).intValue()
4. г) int i = new Integer(getParameter("count"))
5. д) int i = new Integer(getParameter("count")).intValue()

Правильный ответ д).

Вопрос 23. В пользовательском методе show() был изменен цвет фона (background) апплета. Какой метод должен быть вызван, чтобы это было визуализировано?

Варианты ответов

1. а) draw ()
2. б) start ()
3. в) repaint()
4. г) setColor()
5. д) setbgcolor()

Правильный ответ в). Для того чтобы изменения цвета фона стали видны пользователю, требуется перерисовка всего атлета вызовом метода paint(). Это действие можно выполнить, вызвав методы repaint() или update().

Вопрос 24. Выбрать необходимое условие принадлежности класса к апплетам?

Варианты ответов

1. а) Наследник класса Applet и у него должен отсутствовать метод main()
2. б) Наследник класса Applet или его подкласса
3. в) Наследник класса Applet и в нем необходимо переопределить унаследованный от класса Applet метод paint()
4. г) Наследник класса Applet и в нем необходимо переопределить унаследованный от класса Applet метод init()

5. д) Наследник класса Applet и все его методы должны быть объявлены со спецификатором public

Правильный ответ б).

Вопрос 25. Что будет выведено в результате компиляции и запуска такого кода:

```
import java.awt.*;
import java.awt.event.*;
public class Quest3 extends Frame implements WindowListener{
    public Quest3(){
        setSize(300,300);
        setVisible(true);
    }
    public void windowClosing(WindowEvent e){
        System.exit(0);
    }
    public static void main(String args[]){
        Quest3 q = new Quest3 ();
    } }
```

Варианты ответов

1. а) Ошибка компиляции
2. б) Компиляция и запуск с выводом пустого фрейма
3. в) Компиляция без запуска
4. г) Ошибка времени выполнения

Правильный ответ а) Попытка компилировать данный код приведёт к ошибке вследствие того, что часть методов интерфейса WindowListener не реализована в классе Quest3.

Вопрос 26. Что будет выведено в результате компиляции и запуска приведенного кода?

```
class Quest implements Runnable{
    int i=0;
    public int run(){
        System.out.println("i="+ ++i);
        return i;
    }
}
public class Quest1 {
    public static void main(String[] args) {
        Quest ob = new Quest();
        ob.run();
    } }
```

Варианты ответов

1. а) Ошибка компиляции: неправильно определен метод run()
2. б) Ошибка времени выполнения: поток запускается методом start()
3. в) Ошибка компиляции: Объект ob нужно создавать, используя конструктор класса Thread
4. г) i=1
5. д) i=0

Правильный ответ а). Объект потока создаётся только после вызова конструктора класса Thread или его подкласса, но к ошибке компиляции создание такого объекта, как в примере, не приведёт. Поток всегда запускается вызовом метода start(). Результатом же вызова метода run() будет выполнение кода метода run, никак не связанное с потоком. В данной ситуации ошибка компиляции произойдёт из-за того, что сигнатура метода run() в интерфейсе Runnable не совпадает с его реализацией в классе Quest, т.е. метод не реализован и класс Quest должен быть объявлен как abstract.

Вопрос 27. Что будет выведено в результате компиляции и запуска приведенного кода?

```
class Quest5 extends Thread {
Quest5 () { }
Quest5 (Runnable r) { super(r); }
public void run() {
System.out.print("YES ");
}
public static void main(String[] args) {
Runnable r = new Quest5(); //1
Quest5 t = new Quest5(r); //2
t.run();
} }
```

Варианты ответов

1. а) YES YES
2. б) Ошибка компиляции в строке //1
3. в) Ошибка компиляции в строке //2
4. г) YES

Правильный ответ г). При запуске приложения будет создано два потока r и t, но стартует только второй. Поток t инициирован с использованием ссылки на первый поток. Это обстоятельство в данном контексте не оказывает влияния на выполнение второго потока. В итоге метод run() будет вызван только один раз.

Вопрос 28. Что произойдет при попытке компиляции и запуска программы?

```
import java.util.*;
class Quest4 {
public static void main (String args[]) {
Object ob = new HashSet();
System.out.print((ob instanceof Set)+",");
System.out.print(ob instanceof SortedSet);
}}
```

Варианты ответов

1. а) true, false
2. б) true, true
3. в) false, true
4. г) false, false
5. д) ничего из вышеперечисленного

Правильный ответ а).

Вопрос 29. Корректен ли следующий код? Если нет, то какие исправления можно предложить?

```
byte b[]={1, 2, 3};
Object o=b;
o=new String[]{"", "a", "b"};
String s[]=o;
```

Правильный ответ нет. В четвертой строке делается попытка неявного сужения типов от Object к String[]. Такое действие нужно делать явно: String s[]=(String[])o.

Вопрос 30. Корректно ли следующее обращение к переменной x?

```
public class Test {
    static void perform() {
        ...
    }
    private Test x;
    public static void main(String s[]) {
        x.perform(); // корректно ли это выражение?
    }
}
```

Правильный ответ «нет». Хотя при обращении к статическим элементам через имя переменной, используется лишь ее тип, а не значение, в данном примере производится попытка обратиться к динамической переменной из статического метода, чего делать нельзя, несмотря на то, что для вычисления выражения требуется лишь тип переменной.

Вопрос 31. Можно ли при переопределении некоторого абстрактного метода perform() использовать выражение super.perform()?

Правильный ответ нет. Выражение super.perform() означает полноценный вызов родительского метода, что невозможно, если у него отсутствует тело, что верно для абстрактных методов.

Вопрос 32. Как реализованы в классе Object методы equals(), toString(), hashCode() ?

Правильный ответ: в классе Object методы equals(), toString() и hashCode() имеют следующие реализации:

- equals() – возвращает true, если ссылки на объекты совпадают;
- toString() – возвращает строку, которая составляется следующим образом: название класса, символ '@', значение, возвращаемое вызовом метода;
- hashCode(), представленное в шестнадцатеричном виде;
- hashCode() – имеет native реализацию, возвращающую адрес, по которому хранится объект.

Вопрос 33. Какие действия необходимо предпринять для установления TCP-соединения между двумя Java-приложениями?

Правильный ответ: во-первых, на стороне сервера надо создать экземпляр класса Server Socket с указанием порта, и затем вызвать у этого объекта метод accept(). При входе в этот метод поток исполнения приостанавливает свою работу в ожидании подключения клиента. Клиенту

необходимо создать экземпляр класса Socket с указанием IP-адреса и порта сервера. После успешного выполнения конструктора на стороне сервера метод accept() вернет экземпляр класса Socket для взаимодействия двух приложений.

Вопрос 34. Какие объекты могут быть клонированы?

Правильный ответ: если клонирование производится встроенным методом Object.clone(), то такие классы необходимо специальным образом пометить, указывая, что они реализуют интерфейс Cloneable. Кроме этого, класс может переопределить метод clone() собственным образом, и обойтись без этого интерфейса

Основные понятия и свойства баз данных (БД)

I. Задания для самостоятельной работы

1. Объясните отличия между базой данных (БД), системой управления базой данных (СУБД) и автоматизированной информационной системой.
2. Определите термин “схема базы данных”.
3. Объясните значение понятия “приложение базы данных”.
4. Опишите основные сферы применения баз данных.
5. Перечислите основные функциональные возможности и назначение языков баз данных.

II. Рекомендации по выполнению заданий

При выполнении заданий самостоятельной работы студентам следует уделить внимание усвоению значений базовых понятий и используемой терминологии. Необходимо четко понимать, что такое база данных, чем база данных отличается от информационной системы, какие основные компоненты входят в информационные системы на основе баз данных, и т.д. Кроме того, студенты должны знать основные сферы применения баз данных и понимать преимущества, которые обеспечивает их использование современных областях производства, управления и др.

III. Рекомендуемые источники

Основная литература

1. Малыхина, М. П. Базы данных : основы, проектирование, использование / М. П. Малыхина – СПб. : БХВ-Петербург, 2006. – 528 с.

Дополнительная литература

2. Конолли, Т. Базы данных. Проектирование, реализация и сопровождение / Т. Конолли, К.Бегг. – М. : Вильямс, 2000. – 1120 с.

IV. Контрольные вопросы для самопроверки

1. Из каких основных компонент состоит автоматизированная информационная система?
2. Какие основные преимущества обеспечивает применение систем баз данных?
3. Что такое транзакция базы данных и зачем транзакции используются?
4. Зачем используются языки баз данных, и в какие другие средства взаимодействия с базой данных Вы знаете?
5. Какие конкретные информационные системы на основе баз данных Вы знаете?

Назначение и основные компоненты системы управления базой данных (СУБД)

I. Задания для самостоятельной работы

1. Опишите, какие преимущества обеспечивает применение СУБД при построении АИС.
2. Перечислите основные программные компоненты, входящие в состав типичной СУБД.
3. Опишите основные виды архитектур взаимодействия клиентской и серверной частей приложений баз данных.
4. Объясните достоинства трехуровневой модели ANSI/SPARC.

II. Рекомендации по выполнению заданий

При выполнении самостоятельных заданий данного раздела студенты должны уделить внимание следующим основным темам. Во-первых, необходимо усвоить, какие задачи и основные функции обеспечивает использование систем управления базами данных, и понимать отличие СУБД от хранения данных в обычных файлах. Далее следует изучить, какие программное обеспечение и разнообразные компоненты входят в состав современных СУБД, и сформулировать их назначение. Кроме того, студентам необходимо разобраться с архитектурой современных приложений баз данных, в частности, способы взаимодействия клиентских программ с серверами баз данных (клиент-серверная модель, Web-приложения, и т.д.).

III. Рекомендуемые источники

Основная литература

1. Малыхина, М. П. Базы данных : основы, проектирование, использование / М. П. Малыхина – СПб. : БХВ-Петербург, 2006. – 528 с.

Дополнительная литература

2. Конолли, Т. Базы данных. Проектирование, реализация и сопровождение / Т. Конолли, К. Бегг. – М. : Вильямс, 2000. – 1120 с.

IV. Контрольные вопросы для самопроверки

1. Что такое сервер базы данных?
2. Какие основные функции обеспечивает сервер базы данных?
3. Какие преимущества обеспечивает применение клиент-серверной архитектуры?
4. Какие информационные системы, доступные в системе Internet, использующие функции баз данных, Вы знаете?

Модели данных и языки запросов БД. Реляционная модель данных

I. Задания для самостоятельной работы

1. Определите понятия «структуры данных», «операции с данными» и «ограничения» в модели данных?
2. Перечислите известные вам модели данных.
3. Сравните концепции модели данных БД и типов данных традиционных языков программирования.
4. Опишите достоинства и недостатки реляционной модели данных.
5. Опишите основные функции языка базы данных.

II. Рекомендации по выполнению заданий

При изучении материала данного раздела основная задача студентов состоит в получении представлений о том, что такое модель данных, какие модели данных существуют и используются в современной технологии баз данных. Рекомендуется провести аналогии между структурами данных в реляционной модели с типами и структурами данных, используемыми в языках программирования, известными студентам, и выявить различия. Кроме того, следует выяснить отличия между языком запросов базы данных, и традиционными процедурными языками программирования.

III. Рекомендуемые источники

Основная литература

1. Малыхина, М. П. Базы данных : основы, проектирование, использование / М. П. Малыхина – СПб. : БХВ-Петербург, 2006. – 528 с.

Дополнительная литература

2. Конолли, Т. Базы данных. Проектирование, реализация и сопровождение / Т. Конолли, К. Бегг. – М. : Вильямс, 2000. – 1120 с.

IV Контрольные вопросы для самопроверки

1. Какие преимущества обеспечивает использования концепции “модель данных” при работе с базами данных?
2. Какие отличия существуют между языками баз данных и традиционными языками программирования?
3. Объясните смысл термина «множественные операции» в языках запросов.
4. Какие модели данных используются в настоящее?
5. Какие способы взаимодействия пользователя с БД существуют, помимо языков запросов, и в чем их достоинства и недостатки?

Жизненный цикл БД и приложений баз данных

I. Задания для самостоятельной работы

1. Определите значение понятия “жизненный цикл” информационной системы и базы данных.
2. Перечислите основные действия, выполняемые при разработке автоматизированных информационных приложений.

3. Опишите основные риски, возникающие при разработке АИС.
4. Определите, в чем состоит отличие между процессами проектирования базы данных и информационной системы.

II. Рекомендации по выполнению заданий

Данная тема посвящена изучению современной технологии разработки приложений баз данных, и основное внимание при ее изучении следует уделить следующим основным аспектам данного процесса. Прежде всего, необходимо изучить набор этапов, называемых “жизненным циклом”, и последовательность действий, выполняемых на каждом этапе. Особое внимание следует уделить задачам, решаемым на стадиях анализа и проектирования системы. Так же рекомендуется уделить внимание вопросу, какие негативные факторы могут влиять на процесс разработки ПО и проектирование баз данных.

III. Рекомендуемые источники

Основная литература

1. Малыхина, М. П. Базы данных : основы, проектирование, использование / М. П. Малыхина – СПб. : БХВ-Петербург, 2006. – 528 с.

Дополнительная литература

2. Конолли, Т. Базы данных. Проектирование, реализация и сопровождение / Т. Конолли, К. Бегг. – М. : Вильямс, 2000. – 1120 с.
3. Энсор, Д. Oracle. Проектирование баз данных / Д. Энсор, Й. Стивенсон. – Киев :BNV, 1998. – 557 с.
4. Фатрелл, Р. Управление программными проектами / Р. Фатрелл, Д. Шафер, Л. Шафер. – М. : Вильямс, 2003. – 1136 с.

IVКонтрольные вопросы для самопроверки

1. Что такое жизненный цикл АИС?
2. В чем состоят основные функции анализа и проектирования информационной системы?
3. Что такое функциональные требования к АИС?
4. Какие задачи выполняются на стадии эксплуатации приложения баз данных?

Обзор современных СУБД

I. Задания для самостоятельной работы

1. Перечислите основные современные системы баз данных.
2. Сформулируйте достоинства системы OracleServer.
3. Перечислите, какие программные продукты и компоненты входят в систему OracleServer11, и кратко объясните их назначение.
4. Объясните принципы взаимодействия клиентских программ с сервером в системе Oracle.

II. План практических занятий

Тема: «Ознакомление с базовым программным обеспечением OracleServer».

1. Ознакомьтесь с интерфейсом программы доступа к OracleServer.
2. Войдите в систему и измените пароль своей учетной записи при помощи команды ALTERUSER.
3. Ознакомьтесь с основными настройками программы и измените количество хранимых команд (параметр HISTORY).
4. Ознакомьтесь с настройками сессии и установите настройки, соответствующие русскому алфавиту (при помощи команды ALTERSESSION).
5. Исполните SQL-запрос (по заданию преподавателя) и сохраните в файле текст запроса и его результат.
6. При помощи следующего запроса ознакомьтесь с параметрами собственной учетной записи:

```
select * from user_users
```

III. Рекомендации по выполнению заданий и подготовке к практическому занятию

Данная практическая работа предназначена для получения студентами базовых знаний о работе с программой доступа к системе ORACLEServer, изменению разнообразных настроек этой программы и сессии, и вводу и исполнению запросов.

Практическая работа выполняется студентами в учебном классе. Полный текст задания расположен на учебном сервере.

IV. Рекомендуемые источники

Основная литература

1. Малыхина, М. П. Базы данных : основы, проектирование, использование / М. П. Малыхина – СПб. : БХВ-Петербург, 2006. – 528 с.
2. Заставной, Д. Язык баз данных OracleSQL. Запросы и команды манипулирования данными / Д.А.Заставной - Российская таможенная академия, Ростовский филиал, 2008. – 80 с.
3. Моисеенко, С. SQL. Задачи и решения / С. Моисеенко – СПб. : Питер, 2006. – 256 с.

Дополнительная литература

4. Бобровский, С. OracleDatabaseXEдляWindows. Эффективное использование/ С. Бобровский. – М. : Лори. 2009. – 512 с.

V. Контрольные вопросы для самопроверки

1. Какие программные средства входят в «типичную» современную систему баз данных?
2. Что такое «средства разработки приложений» баз данных?
3. Какие параметры необходимо использовать при подключении пользовательской программы к OracleServer?
4. Что такое «пользовательский сеанс», какие основные настройки Вы знаете, и каким образом эти настройки можно изменить?

=Общие сведения

I. Задания для самостоятельной работы

1. Опишите основные категории команд языка SQL.
2. Объясните, что такое SQL-таблица как структура данных.
3. Опишите виды выражений, используемых в языке SQL, и их структуру.
4. Сформулируйте формальное правило вычисления SELECT-запроса с упорядочиванием данных.

II. План практического занятия

Тема: «Составление простых SQL-запросов».

1. Посмотреть структуру таблицы Clients.
2. Определить количество записей в таблице.
3. Вывести все записи таблицы.
4. Вывести все записи таблицы и присвоить колонкам результирующего множества имена на русском языке.
5. Выбрать из таблицы Clients при помощи трех отдельных запросов все существующие значения для:

- различных названий улиц
- различных названий городов
- различных пар значений “название города – название улицы”.

6. Выполнить следующий запрос и объяснить его значение:

```
select count( distinct city ), count( distinct street ),
```

```
count( distinct city || street ) from clients
```

7. Составить запрос, выводящий информацию из таблицы в виде отчета приблизительно в следующем виде:

Клиент Гр.Бабенко Сергей Александрович, проживает по адресу г.Ростов-на-Дону,
Таганрогская, 11/3

(Рекомендация: используйте функцию RTRIM() от значений требуемых полей таблицы для удаления незначащих пробелов)

8. Вывести все записи таблицы, выполнив упорядочивание по:

- дате регистрации
- названию улицы
- номеру дома
- названию улицы и номеру дома
- по выражению `street || apartment`

(упорядочивание выполняется в отдельных запросах)

9. Выполнить следующие запросы и объяснить их значения:

select 1,2,3 from clients

select 1 + 2 +3 from clients

select 1 || 2 || 3 from clients

III. Рекомендации по выполнению заданий и подготовке к практическому занятию

Данная практическая работа является первой работой по освоению языка запросов SQL, и его основная цель – получить студентами начальные сведения о структуре SELECT-выражения и применения запросов к извлечению из базы данных информации. При выполнении заданий по созданию запросов, в этом и последующих практических заданиях, студенты должны четко понимать практические аспекты применения языка SQL и вырабатывать способность самостоятельно создавать запросы в соответствии с формулировками заданий. Поэтому многие задания строятся по следующему принципу – сначала студент должен ввести предлагаемый преподавателем запрос, исполнить его и объяснить смысл результата, а затем на основе этого запроса создать новый в соответствии со следующим заданием.

В данной работе используются запросы с простейшими видами селекции данных, упорядочиванием и удалением дубликатов. Обязательно следует выполнить и разобраться с различиями запросов в задании 9.

IV. Рекомендуемые источники

Основная литература

1. Заставной, Д. Язык баз данных OracleSQL. Запросы и команды манипулирования данными / Д.А.Заставной - Российская таможенная академия, Ростовский филиал, 2008. – 80 с.
2. Андон, Ф. Язык запросов SQL : Учебный курс/ Ф.Андон, В. Резниченко. – СПб.: Питер, 2006. – 415 с.
3. Малыгина, М. П. Базы данных : основы, проектирование, использование / М.П. Малыгина – СПб. : БХВ-Петербург, 2006. – 528 с.
4. Моисеенко, С. SQL. Задачи и решения / С. Моисеенко – СПб. : Питер, 2006. – 256 с.

Дополнительная литература

5. Бобровский, С. Oracle Database XE для Windows. Эффективное использование/ С. Бобровский. – М. : Лори. 2009. – 512 с.

V. Контрольные вопросы для самопроверки

1. Что такое результирующее множество запроса?
2. Что такое литеральные константы?
3. Каким образом следует использовать в именах объектов базы данных символы русского алфавита?
4. Что такое значение SELECT-запроса с формальной и неформальной точек зрения?

Встроенные функции и преобразование типов

I. Задания для самостоятельной работы

1. Перечислите основные категории встроенных функций в языке OracleSQL.
2. Объясните, что такое явное и неявное преобразование типов.
3. Перечислите основные строковые функции и функции работы со значениями-датами.
4. Опишите назначение функции CASE.

II. План практического занятия

Тема: «Использование встроенных функций».

1. При помощи SELECT-запроса, обращающегося к псевдотаблице DUAL, вычислить значения функций SIN(0.34), SQRT(346.45), POWER(12, 3) и ABS(-23.56)
2. Использовать преобразование значений функций из запроса предыдущего задания при помощи функции TO_CHAR() в научный и денежный форматы.
3. Для значений функций запросов из задания 1 выполнить округление при помощи функций ROUND(), CEIL(), FLOOR(), и отбрасывание знаков дробной части до 0, 4 и 10 знаков, используя функцию TRUNC().
4. Выполнить следующий запрос и объяснить отличия значений функций:

```
SELECT title, LENGTH( title ), LENGTH( RTRIM( title ) )
```

```
FROM clients
```

5. Выполнить запрос с упорядочиванием записей по длине поля title с учетом отсека пробелов.
6. Вывести значения поля registrdate, преобразуя символы в нижний регистр и в формате с заглавной буквой (используйте функции LOWER и INITCAP).
7. Вывести имена клиентов, заменяя начальное сочетание 'Гр.' на строку 'Физическое лицо ' (используйте функцию REPLACE с тремя параметрами).
8. Изменить запрос задания 6, выводя только записи со сведениями о физических лицах (т.е. значения поля title которых действительно начинаются с сочетания 'Гр.').

(Рекомендации: необходимо использовать конструкцию WHERE и функцию SUBSTR).

9. Вывести для всех записей значения даты регистрации и результаты ее округления к первому и последнему дням соответствующего месяца (используйте функции TRUNC с форматом 'MONTH' и LAST_DAY).
10. Вывести для всех записей значения даты регистрации и результаты ее округления к первому и последнему дням соответствующего года.

(используйте функцию ADD_MONTHS(registrdate, 12) для округления даты к первому дню следующего года).

III. Рекомендации по выполнению заданий и подготовке

к практическому занятию

В практической работе студенты должны освоить методы работы со встроенными функциями языка SQL, выучить наиболее типичные и употребляемые из них, и научиться их употреблять в создаваемых запросах. Основное внимание при выполнении работы следует уделять строковым функциям и функциям работы со значениями-датами, а так же функциям преобразования типов TO_CHAR() и TO_DATE() и изучаемых функций округления.

Дополнительные рекомендации и подсказки содержатся в текстах заданий.

IV. Рекомендуемые источники

Основная литература

1. Андон, Ф. Язык запросов SQL : Учебный курс/ Ф. Андон, В. Резниченко. – СПб. : Питер, 2006. – 415 с.
2. Моисеенко, С. SQL. Задачи и решения / С. Моисеенко – СПб. : Питер, 2006. – 256 с.
3. Заставной, Д. Язык баз данных OracleSQL. Запросы и команды манипулирования данными / Д.А.Заставной - Российская таможенная академия, Ростовский филиал, 2008. – 80 с.

Дополнительная литература

4. Бобровский, С. Oracle Database XE для Windows. Эффективное использование/ С. Бобровский. – М. : Лори. 2009. – 512 с.

V. Контрольные вопросы для самопроверки

1. Какие преимущества обеспечивает использование встроенных функций в выражениях языка SQL?
2. Какие функции округления значений-дат Вы знаете?
3. Для каких целей используется функция CAST()?
4. Какое преобразование типов – явное или неявное – является более предпочтительным?

Язык Oracle SQL. Использование селекции данных

I. Задания для самостоятельной работы

1. Опишите структуру логических выражений, используемых в языке SQL.
2. Укажите приоритеты выполнения операций сравнения, логических и арифметических операций в выражениях.
3. Объясните правила формирования результирующего множества SELECT-запроса при использовании операций селекции (конструкции WHERE).
4. Постройте таблицы истинности для логических операций AND, OR и NOT.

II. План практических занятий

Тема: «Использование селекции данных. База данных “ГТД”».

1. Просмотреть все записи таблицы GTD06-07.
2. Определить количество записей в таблице GTD06-07.

(используйте функцию COUNT(*)).

3. Составить запрос, извлекающий только sum-поля, и добавить пояснительные синонимы для этих полей.
4. Составить запросы, определяющие записи об экспорте товаров
 1. из Германии;
 2. из Германии, Польши и Франции;

3. из Германии, Польши и Франции через таможенную "КРАСНОДАРСКАЯ";
4. из Германии через таможенную "КРАСНОДАРСКАЯ" или из Франции через таможенную "СТАВРОПОЛЬСКАЯ";
5. не из Китая.
5. Вывести записи таблицы, упорядочивая записи по:
 1. суммарной стоимости;
 2. суммарной стоимости, году и названию таможни.

(используйте ключевое слово ORDERBY).

6. Вывести все различные:
 1. названия таможен ЮФО;
 2. названия отраслей;
 3. стран импорта/экспорта.
7. Набрать запрос и объяснить его смысл:

```
SELECT 'Через таможенную ' || "First-таможня" || ' в ' || "год" ||
' экспортировали ' || "First-группа" || ' в страну ' || "страны" ||
' на сумму ' || "Sum-стоим" || ' р.'
AS "Экспортные операции"
FROM GTD06_07
WHERE "тип ГТД" = 'ЭК'
ORDERBY "страны";
```

Тема: «Использование селекции данных. База данных "Земля"».

1. Определить количество клиентов, зарегистрированных на 'ул.Стачки' (Здесь и далее используемые имена в базе данных изменены).
2. Вывести все записи, содержащие сведения о клиентах – физических лицах, проживающих на 'ул.Стачки, 218'. Вывести все остальные записи, не соответствующие условию селекции задания 2.
3. Вывести записи, в поле Title которых встречается комбинация символов 'Михаиф' двумя способами: при помощи предиката LIKE и при помощи функции INSTR(title, 'Михаиф'), которая возвращает позицию в исходной строке (первый параметр), начиная с которой начинается строка (второй параметр); если строка не найдена, значение функции – ноль.
4. Вывести записи о клиентах, зарегистрированных после 1 января 2000 года.
5. Вывести записи о юридических лицах, зарегистрированных в течение 2003 года двумя способами: при помощи комбинации проверок '<' и '>', и при помощи предиката BETWEEN.
6. Вывести записи о клиентах, дата регистрации которых приходится на сентябрь (используйте функцию TO_CHAR с параметром форматирования даты 'MONTH')
7. Выполнить следующий запрос и объяснить его структуру и значение:

```
select * from clients
```

where TO_NUMBER(TO_CHAR(REGISTRDATE , 'DD'))

BETWEEN 1 AND 10

8. Вывести сведения о клиентах - юридических лицах, которые имеют учредительную форму 'ООО', 'ОАО', 'ТОО' или 'ЗАО' двумя способами: при помощи комбинации проверок на равенство, и при помощи предиката IN (для получения первых трех символов строки см. задание 2)

III. Рекомендации по выполнению заданий и подготовке

к практическим занятиям

Практические работы этого раздела посвящены изучению и практическому применению более сложных видов селекции данных в SELECT-запросах с использованием реальных баз данных ГТД и «Земля». К изучаемому материалу, на который следует обратить внимание, относятся: логические операции и операции сравнения, предикаты LIKE, IN и BETWEEN. Кроме того, в данных работах продолжается изучение встроенные функции.

Дополнительные рекомендации и подсказки содержатся в текстах заданий.

IV. Рекомендуемые источники

Основная литература

1. Андон, Ф. Язык запросов SQL : Учебный курс / Ф. Андон, В. Резниченко. – СПб. : Питер, 2006. – 415 с.
2. Моисеенко, С. SQL. Задачи и решения / С. Моисеенко – СПб. : Питер, 2006. – 256 с.
3. Заставной, Д. Язык баз данных OracleSQL. Запросы и команды манипулирования данными / Д.А.Заставной - Российская таможенная академия, Ростовский филиал, 2008. – 80 с.

Дополнительная литература

4. Бобровский, С. Oracle Database XE для Windows. Эффективное использование/ С. Бобровский. – М. : Лори. 2009. – 512 с.

V. Контрольные вопросы для самопроверки

1. Какие существуют основные логические предикаты в языке SQL?
2. Как выразить предикаты IN и BETWEEN через операции сравнения?

Язык Oracle SQL. Операция группировки и агрегатные функции

I. Задания для самостоятельной работы

1. Опишите правила вычисления SELECT-запросов с группировкой данных.
2. Объясните отличие между операциями группировки и удалением дубликатов (DISTINCT).
3. Перечислите основные агрегатные функции языка SQL и объясните правила их вычисления.

4. Перечислите основные множественные операции и их использование.