



**ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«ГУМАНИТАРНЫЙ ТЕХНИКУМ ЭКОНОМИКИ И ПРАВА»**

105318, Россия, г. Москва, Ибрагимова ул., д. 31, к.1. Тел: +7(499) 166-02-27

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Учебной дисциплины

ОП.04 Основы алгоритмизации и программирования

программы подготовки специалистов среднего звена
по специальности

09.02.07 Информационные системы и программирование

Квалификации- программист.



Москва 2023г.

ГУМАНИТАРНЫЙ ТЕХНИКУМ
ЭКОНОМИКИ И ПРАВА

Фонд оценочных средств учебной дисциплины «Основы алгоритмизации и программирования» разработан на основе:

- Приказ Минобрнауки России от 09.12.2016 N 1547 "Об утверждении федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование" (с изменениями и дополнениями от 17 декабря 2020 г., 1 сентября 2022 г.);

РАССМОТРЕН

На заседании ПЦК

«Информационные технологии»

Председатель ПЦК


Е.В. Чегодаева

Организация-разработчик: Профессиональное образовательное учреждение «ГУМАНИТАРНЫЙ ТЕХНИКУМ ЭКОНОМИКИ И ПРАВА»



ГУМАНИТАРНЫЙ ТЕХНИКУМ
ЭКОНОМИКИ И ПРАВА

СОДЕРЖАНИЕ

Пояснительная записка	4
Оценка освоения учебной дисциплины.....	7
Текущий контроль освоения ОП.4. Основы алгоритмизации и программирования.....	7
Промежуточная аттестация по ОП.4. Основы алгоритмизации и программирования	17
Вопросы для подготовки к экзамену	17
Билеты для проведения экзамена	18
Информационное обеспечение обучения	32



Пояснительная записка

Фонд оценочных средств (далее – ФОС) является неотъемлемой частью нормативно-методического обеспечения системы оценки качества освоения обучающимися основной профессиональной образовательной программы подготовки специалистов среднего звена и обеспечивает повышение качества образовательного процесса.

Целью создания ФОС является установление соответствия уровня подготовки обучающихся на конкретном этапе обучения требованиями Федерального государственного стандарта среднего профессионального образования и иным нормативным документам.

ФОС по общепрофессиональной дисциплине ОП.04 Основы алгоритмизации и программирования представляет собой совокупность контролирующих материалов, позволяющих оценить освоенные компетенции при проведении текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине.

Цель и планируемые результаты освоения дисциплины

Код ПК, ОК	Умения	Знания
ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, ПК 2.5	<ul style="list-style-type: none">- разрабатывать алгоритмы для конкретных задач;- использовать программы для графического отображения алгоритмов;- определять сложность работы алгоритмов;- работать в среде программирования;- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;- оформлять код программы в соответствии со стандартом кодирования;- выполнять проверку, отладку кода программы.	<ul style="list-style-type: none">- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;- эволюцию языков программирования, их классификацию, понятие системы программирования;- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;- подпрограммы, составление библиотек подпрограмм;- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения



2. Оценка освоения учебной дисциплины УД

2.1 Текущий контроль освоения УД

РАЗДЕЛ

«Введение в программирование»

Задания с выбором одного варианта правильного ответа

Вариант 1

1. Алгоритм – это

- a. последовательность команд, выполнение которых приводит нас к решению поставленной задачи; +
- b. последовательность действий, выполнив которые, мы можем запустить программу на выполнение;
- c. задача, которая имеет решение.

2. Какой из документов является алгоритмом?

- a. правила техники безопасности;
- b. рецепт приготовления торта; +
- c. список книг в библиотеке;
- d. расписание движения поездов;
- e. режим дня.

3. Запись алгоритма на каком-либо языке программирования называется

- a. текстом программы; +
- b. блок-схемой алгоритма;
- c. текстовой записью алгоритма.

4. Алгоритмизация – это

- a. разбор постановки задачи;
- b. процесс составления алгоритма; +
- c. ввод текста программы;
- d. оптимизация заданного алгоритма.

5. Прекращение процесса через определенное число шагов с выдачей искомого результатов или сообщения о невозможности продолжения вычислительного процесса называется

- a. детерминированностью;
- b. дискретностью;
- c. результативностью; +
- d. массовостью.

6. Пригодность алгоритма для решения всех задач заданного класса называется

- a. детерминированностью;
- b. дискретностью;
- c. результативностью;
- d. массовостью. +

7. Какое из ниже перечисленных свойств не относится к основным свойствам алгоритма?

- a. точность;



- b. корректность; +
- c. результативность;
- d. массовость.

Вариант 2

1. Набор символов и правил образования и истолкования конструкций из этих символов для записи алгоритмов называется

- a. языком программирования;
- b. алгоритмическим языком; +
- c. блок-схемой;
- d. псевдокодом.

2. Если содержание этапов вычислений задается на естественном языке в произвольной форме с требуемой детализацией, то способ записи алгоритма называется

- a. словесным; +
- b. формульно-словесным;
- c. псевдокодом;
- d. с помощью языка программирования.

3. Если для записи алгоритма применяется задание инструкций с использованием математических символов и выражений в сочетании со словесными пояснениями, то способ записи алгоритма называется

- a. словесным;
- b. формульно-словесным; +
- c. псевдокодом;
- d. с помощью языка программирования.

4. Если для записи алгоритма применяется графическое изображение логической структуры алгоритма, то способ записи алгоритма называется

- a. структурной диаграммой;
- b. блок-схемным; +
- c. псевдокодом;
- d. с помощью языка программирования.

5. Если для записи алгоритма применяется формальное изображение логики программы без использования синтаксических особенностей конкретного языка программирования, то способ записи алгоритма называется

- a. структурной диаграммой;
 - b. блок-схемным;
 - c. псевдокодом; +
 - d. словесным.
- 6. Блок-схема позволяет:**
- a. получить ответ решения задачи;
 - b. разбить задачу на блоки;
 - c. определить вид алгоритма;



- d. описать исходные данные;
- e. наглядно изобразить последовательность шагов решения задачи. +

7. Когда необходимо составлять блок-схему программы?

- a. до начала составления самой программы; +
- b. после составления программы;
- c. в процессе составления программы.

Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность

1. Требовалось написать программу, при выполнении которой с клавиатуры вводится последовательность из шести неотрицательных целых чисел, не превышающих 10^6 , подсчитывается и выводится сумма введённых чётных чисел или 0, если чётных чисел в последовательности нет. Обучающийся написал такую программу:

Бейсик	Паскаль
<pre> DIM N, S AS LONG DIM I AS INTEGER S = 1 FOR I = 1 TO 6 INPUT N IF I MOD 2 = 0 THEN S = S + N END IF NEXT I PRINT S END </pre>	<pre> var n, s: longint; i: integer; begin s:=1; for i:=1 to 6 do begin readln(n); if i mod 2 = 0 then s := s + n; end; write(s); end. </pre>
Си++	Алгоритмический
<pre> #include <iostream> using namespace std; int main(){ long int n, s; int i; s = 1; for (i=1; i<=6; ++i) { cin >> n; if (i%2 == 0) s = s + n; } cout << s; } </pre>	<pre> алг нач цел n, s цел i s := 1 нц для i от 1 до 6 ввод n если mod(i,2) = 0 то s := s + n все кц вывод s кон </pre>
Python	
<pre> s = 1 for i in range(1, 7): n = int(input()) if i % 2 == 0: s = s + n print(s) </pre>	

Последовательно выполните следующее.

1. Напишите, что выведет эта программа при вводе последовательности 1, 1, 2, 3, 5, 8.
2. Приведите пример последовательности, при вводе которой программа выдаст верный ответ.
3. Найдите в программе все ошибки (их может быть одна или несколько).

Для каждой ошибки выпишите строку, в которой она допущена, и приведите эту же строку в исправленном виде. Обратите внимание: Вам нужно исправить приведённую программу, а не написать свою. Вы можете только заменять ошибочные строки, но не можете удалять строки или добавлять новые. Заменять следует только ошибочные строки: за исправления, внесённые в строки, не содержащие ошибок, баллы будут снижаться.

Решение:

. При вводе последовательности 1, 1, 2, 3, 5, 8 программа выведет число 13.

Комментарий. Приведённая программа вместо подсчёта суммы чётных чисел считает сумму чисел, стоящих на чётных позициях в последовательности, и дополнительно увеличивает эту сумму на 1 из-за неверной инициализации.

2. Пример последовательности, для которой программа даёт верный ответ:

2, 1, 1, 2, 1, 2.

Комментарий. При любом вводе данных результатом программы будет сумма второго, четвертого и шестого чисел, увеличенная на 1. Чтобы результат был верным, нужно подобрать такую последовательность, в которой эта сумма совпадет с суммой чётных чисел. Например, в приведённой выше последовательности обе суммы равны 6.

3. Ошибки содержатся в двух строках программы:

1) неверная инициализация переменной s : вместо значения 1 нужно присваивать 0;

2) неверная проверка условия: вместо чётности i нужно проверять четность n .

Критерии оценивания

Критерии оценивания выполнения задания	Баллы
<p>Правильно выполнены все действия:</p> <ul style="list-style-type: none"> – указан верный результат для приведённого примера входных данных; – дан пример последовательности, для которой программа с ошибками выдаёт верный результат; – указаны и исправлены две ошибки в программе; – не указаны в качестве ошибочных никакие другие строки программы. 	3
<p>Не выполнены условия, позволяющие поставить 3 балла, и имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия (верный результат при указанных данных, верный пример последовательности с правильным результатом), найдена и исправлена одна ошибка в программе, ни одна верная строка не названа ошибочной.</p> <p>2. Выполнены два первых действия (верный результат при указанных данных, верный пример последовательности с правильным результатом), найдены и исправлены две ошибки в программе, одна верная строка названа ошибочной.</p> <p>3. Выполнено одно из первых двух действий (верный результат при указанных данных или верный пример последовательности с правильным результатом), найдены и исправлены две ошибки в программе, ни одна верная строка не названа ошибочной.</p>	2
<p>Не выполнены условия, позволяющие поставить 2 или 3 балла, и имеет место один из следующих случаев.</p> <p>1. Выполнены два первых действия (верный результат при указанных данных, верный пример последовательности с правильным результатом). При этом несущественно, насколько правильно выполнено третье действие.</p> <p>2. Найдены и исправлены две ошибки в программе, не более чем одна верная строка названа ошибочной. При этом несущественно, насколько правильно выполнены действия 1 и 2</p>	1
<p>Не выполнены условия, позволяющие поставить 1, 2 или 3 балла.</p>	0
Максимальный балл	3

2. Определите, что будет напечатано в результате работы следующего фрагмента программы:



Бейсик	Python
<pre> DIM K, S AS INTEGER S = 0 K = 0 WHILE K < 30 K = K + 3 S = S + K WEND PRINT S </pre>	<pre> s = 0 k = 0 while k < 30: k += 3 s += k print(s) </pre>
Паскаль	Алгоритмический язык
<pre> var k, s: integer; begin s:=0; k:=0; while k < 30 do begin k:=k+3; s:=s+k; end; write(s); end. </pre>	<pre> алг нач цел k, s s := 0 k := 0 нц пока k < 30 k := k + 3 s := s + k кц вывод s кон </pre>
Си++	
<pre> #include <iostream> using namespace std; int main() { int s, k; s = 0, k = 0; while (k < 30) { k = k + 3; s = s + k; } cout << s << endl; return 0; } </pre>	

eqe.sdamaia.ru

Решение.

Цикл while выполняется до тех пор, пока истинно условие $k < 30$, т.е. переменная k определяет, сколько раз выполнится цикл.

Так как последовательность k представляет собой арифметическую прогрессию, найдем n из неравенства:

$k_n = k_1 + (n-1)d < 30$, $k_1 = 0$, $d = 3$ (т.к. $k := k+3$). Воспользовавшись методом интервалов, находим первое натуральное n , при котором нарушается условие: $n = 11$.

Значение s есть сумма первых n членов арифметической прогрессии. $b = \frac{2a_1 + (n-1)d}{2}n$, b — сумма первых n членов прогрессии, d — разность прогрессии, n — количество членов.

$$s = \frac{2s_1 + (n-1)d}{2}n = \frac{0 + (11-1) \cdot 3}{2}11 = 165.$$

Ответ: 165.



Критерии оценивания теста (пример) (разрабатывается обязательно):

Оценка «5» ставится при правильном выполнении 85% заданий теста.

Оценка «4» ставится при правильном выполнении 60% заданий теста.

Оценка «3» ставится при правильном выполнении 45% заданий теста.

Оценка «2» ставится при правильном выполнении 15% заданий теста.

РАЗДЕЛ

«Программирование на алгоритмическом языке»

Задания с выбором одного варианта правильного ответа

Вариант 1

- Что представляет собой оператор присваивания в Python?**
 - :=
 - =:
 - = +
 - ◇.
- Для возведения в квадрат в языке Python применяется функция:**
 - Pow +
 - exp
 - sqrt
 - sqr
- В выражении $summa = x**2 + 3*a$ переменными являются:**
 - a, x, summa; +
 - x, a;
 - только x;
 - только a.
- Укажите логическое выражение, возвращающее True при $x=5$:**
 - $-10 < x < 10$ +
 - $0 < x$ and $x < 2$
 - $x > 3$ and $x < 5$
 - $x \% 2 == 0$ or $x < 5$
- Оператором вывода в языке Python является:**
 - readln;
 - input;
 - print; +
 - write.



6. Какой из указанных операторов называется оператором цикла с параметром?
- For; +
 - If;
 - Def;
7. Чему равно число итераций цикла, если начальное значение параметра больше конечного значения, а шаг – положительный?
- ни одного раза; +
 - один раз;
 - (Начальное значение – Конечное значение) / Шаг;
 - (Конечное значение – Начальное значение) / Шаг.
8. Если в цикле **for i in range (A, B)**: значение B больше, чем значение A, то блок инструкций в теле цикла
- не выполняется ни разу;
 - выполняется один раз;
 - выполняется (B-A) раз; +
 - выполняется (B-A-1) раз.
9. В операторе цикла с предусловием: **while B**: при **B=False** блок инструкций в теле цикла
- выполняется один раз;
 - не выполняется ни разу; +
 - выполняется до тех пор, пока B не станет равно True.
10. Если в конструкции **If –Else** условие ложно, то
- выполняются операторы, расположенные после «If <условие>:»;
 - выполняются операторы, расположенные после команды «Else:»; +
 - выполняются операторы, расположенные после конструкции If –Else.
11. Если элементы списка **D** равны соответственно 4, 1, 5, 3, 2, то значение выражения **D [D[2]] – D[D[4]]** равно
- 2;
 - 1; +
 - 3;
 - 1.
12. В алгоритмах и языках программирования величинами не являются:
- нет правильного ответа +
 - переменные
 - констант
13. Данные какого типа могут принимать значения **True** и **False**?
- Логического +
 - символьного
 - вещественного
 - целого



14. Как читается команда « \Leftarrow »?

- a. Присвоить +
- b. Приравнять
- c. вычислить
- d. выполнить

15. Какая из инструкций цикла является циклом с условием?

- a. For;
- b. If;
- c. Def;
- d. While. +

Вариант 2

1. При выполнении последовательности операторов

$a=1.0; b=3; x=(a+b)/a*b-a$

значение переменной x равно:

- a. 2;
- b. 1;
- c. 0.333; +
- d. 11.

2. Общее название для программ, преобразующих текст программы в машинные инструкции, - это

- a. трансляторы; +
- b. компиляторы;
- c. интерпретаторы.

3. Программа – это

- a. протокол взаимодействия компонентов компьютерной сети;
- b. набор команд операционной системы компьютера;
- c. алгоритм, записанный на языке программирования; +
- d. законченное минимальное смысловое выражение на языке программирования.

4. Как называется поименованная область памяти, предназначенная для временного хранения данных?

- a. константа;
- b. переменная; +
- c. функция;
- d. оперативная память.

5. Как называется область памяти для хранения данных, изменение которых во время работы программы не допускается?

- a. константа; +
- b. переменная;
- c. функция;
- d. оперативная память.
- b. **Список** — это ...
- a. поименованный набор фиксированного числа одноименных данных;
- b. ограниченная кавычками последовательность любых символов;



c. совокупность разнородных данных, описываемых и обрабатываемых как единое целое; +

d. именованный набор однотипных данных на диске;

7. Назначение списков:

a. хранение данных;

b. облегчение записи программы;

c. хранение больших чисел;

d. упорядоченное хранение данных. +

8. Что может содержать в себе элемент массива (таблицы)?

a. только положительные числа;

b. только отрицательные числа;

c. числа, совпадающие с адресом элемента;

d. любые данные; +

9. Что такое индекс в списке?

a. порядковый номер элемента списка; +

b. наибольший размер элемента списка;

c. размерность списка;

d. имя списка.

10. При инициализации произведения P элементов числового списка нужна следующая строка:

a. P=1 +

b. P=-1

c. P=0

d. P=2

11. При определении суммы S элементов числового списка нужна строка:

a. S=1

b. S=0 +

c. S=-1

d. S=100

12. Что не является подпрограммой в языках программирования?

a. Список +

b. функция

c. множество

d. процедура

13. Процедуры в программе описываются в:

a. до момента первого обращения к ней +

b. разделе описания переменных

c. основной программе

d. конце программы

14. Какой процедурой в Python осуществляется запись в файл?

a. Write +

b. read



- c. rewrite
- d. reset

15. **Как преобразовать данные из строкового представления в вещественное числовое:**

- a. int
- b. str
- c. float +
- d. def

Критерии оценивания теста (пример) (разрабатывается обязательно):

Оценка «5» ставится при правильном выполнении 85% заданий теста.

Оценка «4» ставится при правильном выполнении 60% заданий теста.

Оценка «3» ставится при правильном выполнении 45% заданий теста.

Оценка «2» ставится при правильном выполнении 15% заданий теста.

РАЗДЕЛ **«Подпрограммы»**

Вопросы для устного ответа

1. Что такое подпрограмма? Перечислите основные свойства подпрограмм.
2. Охарактеризуйте структуру подпрограммы.
3. Чем отличается процедура от функции?
4. Что такое рекурсия? Охарактеризуйте ее.
5. Опишите основные особенности реализации алгоритмов с использованием модулей.
6. В чем состоит модульность при структурной разработке алгоритмов?
7. Что такое нисходящее и восходящее проектирование программ?
8. Что такое библиотека? Каким образом подключить ее в программе?

Критерии оценивания устного ответа:

Отметка «5» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Отметка «4» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Отметка «3» - ответил на часть вопросов в объеме лекционного материала и ОТВЕТИЛ на часть дополнительных вопросов.

Отметка «2» - допустил ошибки в определении базовых понятий, искажил их смысл, не ответил на дополнительные вопросы;

Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность.

Задача1:

Описать процедуру нахождения по заданной длине стороны треугольника и величинам двух прилежащих углов длин остальных сторон треугольника и величины третьего угла.

Алгоритм решения задачи:

По известным двум углам треугольника и стороне между ними можно найти остальные стороны, используя теорему синусов: отношение синусов углов треугольника к длинам противоположных сторон равны между собой. Для треугольника ABC получаем:

$$\sin A / BC = \sin B / AC = \sin C / AB.$$

$$\text{Отсюда } AC = (\sin B * AB) / \sin C,$$

$$BC = (\sin A * AB) / \sin C.$$

Если известны два угла треугольника, то третий угол легко вычисляется, учитывая, что сумма углов треугольника равна 180 градусам.

В Pascal функция `sin ()` принимает угол, выраженный в радианах. Чтобы перевести градусы в радианы нужно воспользоваться формулой:

$$\text{rad} = \text{Угол} * \pi / 180.$$

Программа на языке Паскаль:

var

K, L, M, KL, LM, KM: **real**;

Procedure triangle (**var** A, B, C, AB, BC, AC: **real**);

var radA, radB, radC: **real**;

begin

C := 180 - A - B;

radA := A * PI/180;

radB := B * PI/180;

radC := C * PI/180;

BC := (sin (radA) * AB) / sin (radC);

AC := (sin (radB) * AB) / sin (radC);

end;

begin

write('A = '); readln(K);

write('B = '); readln(L);

write('AB = '); readln(KL);

triangle(K,L,M,KL,LM,KM);

writeln('C = ',M:7:2);

writeln('BC = ',LM:6:2);

writeln('AC = ',KM:6:2);

readln;

end.

Задача2:

Описать функцию вычисления $f(x)$ по формуле:

$$f(x) = x^2 \text{ при } -2 \leq x < 2;$$

$$x^2 + 4x + 5 \text{ при } x \geq 2;$$

$$4 \text{ при } x < -2.$$

Используя эту функцию для n заданных чисел, вычислить $f(x)$. Среди вычисленных значений найти наибольшее.

Программа на языке Паскаль:

const

n = 5;

var

a, max: **integer**;

i: **byte**;

arr: **array**[1..n] **of integer**;

function func (x: **integer**): **integer**;

```

begin
  if (x >= -2) and (x < 2) then
    func := x * x
  else
    if x >= 2 then
      func := x * x + 4 * x + 5
    else
      func := 4;
  end;
end;

begin
  max := -32768;
  for i:=1 to n do begin
    write('a = '); readln(a);
    arr[i] := func(a);
    if arr[i] > max then
      max := arr[i];
  end;

  write('Result: ');
  for i:=1 to n do
    write(arr[i], ' ');
  writeln;
  writeln('max = ', max);

readln;
end.

```

Задача 3:

Число, представленное в шестнадцатеричной системе счисления перевести в десятичную систему счисления.

Алгоритм решения задачи:

Алгоритм аналогичен [переводу двоичного числа](#), за исключением того, что цифры, обозначенные буквами от А до F следует заменить цифрами от 10 до 15.

Программа на языке Паскаль:

```

var
  hexa: string; {шестнадцатеричное число}
  n: byte; {его длина}
  ch: char; {цифра очередного разряда как символ}
  digit: byte; {цифра очередного разряда}
  m: byte; {степень 16}
  decimal: integer; {десятичное число}
  i: byte;

function pow16(m: byte): integer;
var i: byte;
begin
  pow16 := 1;
  for i := 1 to m do
    pow16 := pow16 * 16;
  end;
begin
  write('Hexadecimal: ');
  readln(hexa);

  n := length(hexa);

  decimal := 0;

```

```

m := n;
for i:=1 to n do begin
  ch := hexa[i];
  if ch in ['A'..'F'] then
    case ch of
      'A': digit := 10;
      'B': digit := 11;
      'C': digit := 12;
      'D': digit := 13;
      'E': digit := 14;
      'F': digit := 15
    end
  else
    digit := ord(ch) - ord('0');
  m := m - 1;
  decimal := decimal + digit * pow16(m);
end;

writeln('Decimal: ', decimal);

readln
end.

```

Задача 4.

Алгоритм решения задачи:

Представленная ниже программа с процедурой вычисления корней квадратного уравнения не возвращает в основную программу ничего (просто выводит результат на экран). Однако можно написать такую процедуру, которая будет использовать глобальные переменные $x1$ и $x2$. В результате в основной ветке программы можно будет использовать полученные корни квадратного уравнения.

Программа на языке Паскаль:

```

var
  a, b, c: real;

procedure sq (a,b,c: real);
  var d, x1, x2: real;
  begin
    d := b * b - 4 * a * c;
    if d >= 0 then begin
      x1 := (-b + sqrt (d)) / (2 * a);
      x2 := (-b - sqrt (d)) / (2 * a);
      if x1 = x2 then
        writeln ('x1 = ', x1:6:2)
      else
        writeln ('x1 = ', x1:6:2, '; x2 = ', x2:6:2)
    end
  else
    writeln ('Корней нет!')
  end;

begin

  write ('a = '); readln (a);
  write ('b = '); readln (b);
  write ('c = '); readln (c);

  writeln (a:6:2, 'x*x + ', b:6:2, 'x + ', c:6:2, ' = 0');

  sq (a, b, c);

readln
end.

```

Задача5:

Найти разность двух матриц по модулю, т.е. из большего элемента одной матрицы вычесть меньший элемент второй матрицы, находящийся в той же позиции.

Алгоритм решения задачи:

Решение задачи нахождения разности двух матриц аналогично нахождению [суммы матриц](#). Поскольку нужно из большего элемента вычесть меньший, то можно было бы воспользоваться оператором ветвления. Однако проще использовать встроенную в Pascal функцию abs(), которая возвращает модуль переданного ей параметра. При этом не важно из большего элемента вычитается меньший (например, 5 - 3) или из меньшего больший (3 - 5), результат будет один и тот же.

Программа на языке Паскаль:

```
const N = 2; M = 5;
type arr = array[1..N,1..M] of integer;
var
    one,two,three: arr;
    i,j: byte;

procedure matrix(var a: arr);
begin
    for i:=1 to N do
        for j:=1 to M do
            a[i,j] := random(100);
        end;
    end;

procedure minus(var a: arr; b: arr; c: arr);
begin
    for i:=1 to N do
        for j:=1 to M do
            a[i,j] := abs(b[i,j]-c[i,j]);
        end;
    end;

procedure printer(a: arr);
begin
    for i:=1 to N do begin
        for j:=1 to M do
            write(a[i,j]:4);
        writeln;
    end;
end;

begin
    randomize;
    matrix(one);
    printer(one);
    writeln;
    matrix(two);
    printer(two);
    writeln;
    minus(three,one,two);
    printer(three);
end.
```

Задача 6:

Преобразовать строку так, чтобы буквы каждого слова в ней были отсортированы по алфавиту.

Алгоритм решения задачи:

Решение данной задачи в общей сложности сводится к двум подзадачам. Это 1) найти индексы первого и последнего очередного слова, 2) отсортировать буквы между этими индексами. Эти две подзадачи должны выполняться столько раз, сколько слов в строке.

Вся строка в цикле перебирается посимвольно. Вводится флаговая переменная, которая сигнализирует о том, находимся ли мы внутри слова или нет. Если очередной символ не

пробел и флаг показывает, что мы не в слове, то значит, текущий символ - первая буква слова. Присвоим его индекс переменной `start`, поменяем значение флаговой переменной (теперь она указывает, что мы внутри слова).

Если очередной символ не пробел, и флаг указывает, что мы внутри слова, то присвоим текущий индекс переменной `finish`. Если впоследствии за ним будет идти буква, то `finish` изменится, если же пробел - то эта переменная будет иметь индекс последней буквы слова.

Если встречается пробел, то флаг сбрасывается (мы вне слова). Также буквы с индексами от `start` до `finish` сортируются. Если после последнего слова строки нет пробела, то оно не будет отсортировано. Поэтому после цикла перебора символов строки выполняется еще одна сортировка.

Сортировка оформлена в виде отдельной процедуры. В качестве параметров передаются сама строка, начало и конец среза для сортировки. Используется сортировка выбором, т.к. можно использовать индексы непосредственно. В то время как в сортировке пузырьком находится разность между ними, что в данном случае приведет к ошибке.

Программа на языке Паскаль:

```
var
  s: string;
  l, i, start, finish: byte;
  flag: boolean;
procedure sort(var s: string; first, last: byte);
  var
    max, i: byte;
    c: char;
  begin
    while last > first do begin
      max := first;
      for i:=first+1 to last do
        if s[i] > s[max] then
          max := i;
      c := s[max];
      s[max] := s[last];
      s[last] := c;
      last := last - 1;
    end;
  end;
begin
  readln(s);
  l := length(s);
  start := 1;
  finish := 1;
  flag := False;
  for i:=1 to l do
    if s[i] = ' ' then begin
      flag := False;
      if start < finish then
        sort(s, start, finish);
    end
    else
      if flag = False then begin
        flag := True;
        start := i;
      end
      else
        finish := i;

    if start < finish then // для последнего слова, когда после него не было
пробела
      sort(s, start, finish);

  writeln(s);
end.
```

Задача 7:



С помощью подпрограммы заполнить матрицы случайными числами. Написать подпрограмму, вычисляющую сумму двух матриц. Вывести на экран две исходные матрицы и их сумму (используя процедуру).

Описание переменных:

one, two - исходные матрицы;
three - матрица-сумма двух предыдущих;
matrix - процедура, заполняющая массив случайными числами;
printer - процедура, выводящая содержимое массивов на экран;
plus - процедура, вычисляющая сумму матриц.

Алгоритм решения задачи:

Под суммой матриц будем понимать сложение их элементов, находящихся в одинаковых позициях (имеющих одинаковые индексы). Таким образом уместно складывать матрицы одинаковой размерности. При этом будет получена третья матрица с такой же размерностью как исходные.

При решении подобной задачи лучше использовать подпрограммы (процедуры или функции), так как нам приходится заполнять несколько массивов и выводить их на экран. Без подпрограмм в коде будет содержаться много почти идентичного кода.

Используются процедуры, а не функции, так как при заполнении массивов в подпрограмму передается переменная, а не значение. Таким образом, процедура заполняет "внешнюю" для нее матрицу.

В процедуру, вычисляющую сумму матриц, передается три параметра: переменная матрицы-суммы, значения первой и второй матрицы.

В основной ветке программы процедуры последовательно вызываются.

Следует обратить внимание, что randomize вызывается единожды в основной ветке программы. Если данную команду вставить в процедуру заполнения массива, то оба массива будут заполнены одинаковыми числами. Это связано с тем, что "зерно" зависит от таймера, а между двумя вызовами проходит мало времени, чтобы таймер изменил значение. Таким образом, "зерно" в программе надо получать один раз. В этом случае при повторном вызове процедуры отсчет по формуле генерации псевдослучайных чисел продолжается, а не инициализируется заново.

Программа на языке Паскаль:

```
const N = 2; M = 5;
type arr = array[1..N,1..M] of integer;
var
    one,two,three: arr;
    i,j: byte;

procedure matrix(var a: arr);
begin
    for i:=1 to N do
        for j:=1 to M do
            a[i,j] := random(100);
        end;
    end;

procedure plus(var a: arr; b: arr; c: arr);
begin
    for i:=1 to N do
        for j:=1 to M do
            a[i,j] := b[i,j]+c[i,j];
        end;
    end;

procedure printer(a: arr);
begin
    for i:=1 to N do begin
        for j:=1 to M do
            write(a[i,j]:4);
        end;
        writeln;
    end;
end;
```

```

        end;
    end;
begin
    randomize;
    matrix(one);
    printer(one);
    writeln;
    matrix(two);
    printer(two);
    writeln;
    plus(three, one, two);
    printer(three);
end.

```

РАЗДЕЛ «Программирование в объектно-ориентированной среде»

1. Укажите, какой из нижеприведённых компонентов служит для отображения текста на пользовательской форме:
 - a. Title
 - b. Label +
 - c. Entry
 - d. Button
2. Укажите, какой из нижеприведённых компонентов служит для отображения поля ввода на пользовательской форме:
 - a. Title
 - b. Label
 - c. Entry +
 - d. Button
3. Укажите, какой из нижеприведённых компонентов служит для отображения кнопки на пользовательской форме:
 - a. Title
 - b. Label
 - c. Entry
 - d. Button +
4. Укажите, какой из нижеприведённых компонентов служит для отображения текста в строке заголовка на пользовательской форме:
 - a. Title +
 - b. Label
 - c. Entry
 - d. Button
5. Укажите общий метод для изменения атрибутов компонента:
 - a. Config +
 - b. pack
 - c. destroy
 - d. add_command
6. Укажите метод для упаковки компонента на пользовательскую форму:
 - a. config
 - b. pack +
 - c. destroy
 - d. add_command
7. Укажите метод уничтожений виджета:



- a. config
 - b. pack
 - c. destroy +
 - d. add_command
8. Укажите метод добавления пункта меню:
- a. config
 - b. pack
 - c. destroy
 - d. add_command +
9. Каким методом осуществляется запуск цикла обработки события:
- a. Mainloop +
 - b. pack
 - c. destroy
 - d. root
10. Как называется библиотека для создания графического приложения:
- a. mainloop
 - b. grid
 - c. tkinter +
 - d. root
11. Как называется компонент холста для создания графического приложения:
- a. Canvas +
 - b. grid
 - c. tkinter
 - d. root
12. Как называется метод для отрисовки на холсте дуги эллипса:
- a. create_oval
 - b. create_arc +
 - c. create_poligon
 - d. create_rectangle
13. Как называется метод для отрисовки на холсте многоугольника:
- a. create_oval
 - b. create_arc
 - c. create_poligon +
 - d. create_rectangle
14. Какие значения не может принимать якорь привязки компонента контейнере:
- a. nw
 - b. ne
 - c. se
 - d. se +
15. Процесс преобразования исходный программы в исполняемую – это:
- a. интерпретация
 - b. преобразование
 - c. компиляция +
 - d. трансформация

Критерии оценивания теста (пример) (разрабатывается обязательно):

Оценка «5» ставится при правильном выполнении 85% заданий теста.

Оценка «4» ставится при правильном выполнении 60% заданий теста.

Оценка «3» ставится при правильном выполнении 45% заданий теста.

Оценка «2» ставится при правильном выполнении 15% заданий теста.

Вопросы для устного опроса

1. В чем заключается визуальное объектно-ориентированное программирование?
2. Какие особенности языка программирования Python заслуженно называют «магией»?
3. Какие средства используются системами быстрой разработки программ?
4. Что включает разработка приложения в Python?
5. Как создать интерфейс приложения?
6. В чем заключается функциональность приложения?
7. Что представляют собой поля, методы, свойства?
8. Что включают классы в Python?
9. Какие свойства объектов являются свойствами классов?
10. Назначение палитры компонентов?

Критерии оценивания устного ответа:

Отметка «5» - ответил на вопросы в объеме лекционного и дополнительного материала, дал полные грамотные ответы на все дополнительные вопросы.

Отметка «4» - грамотно изложил ответы на вопросы, но содержание и формулировки имеют отдельные неточности (допускается нечеткая формулировка определений), в полной мере ответил на заданные дополнительные вопросы.

Отметка «3» - ответил на часть вопросов в объеме лекционного материала и ответил на часть дополнительных вопросов.

Отметка «2» - допустил ошибки в определении базовых понятий, исказил их смысл, не ответил на дополнительные вопросы;

Вопросы с развернутым ответом

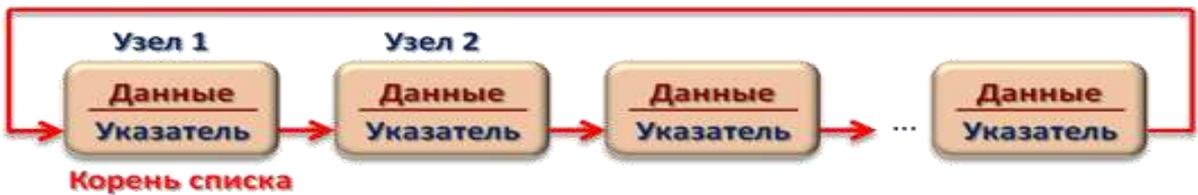
Формируемые компетенции: ОК1, ОК2, ОК4, ОК5, ОК9, ОК10, ПК 1.1, ПК 1.2.

1. Каковы этапы жизненного цикла программного продукта?
2. Каковы основные принципы структурного программирования?
3. Какие вы знаете технологии программирования?
4. Что представляет собой программа, описанная в стиле модульного программирования?
5. Чем отличаются утилитарные программы от программных продуктов?
6. Каковы достоинства структурного программирования?
7. Каковы основные принципы объектно-ориентированного программирования?
8. Что такое инкапсуляция?
9. Что представляет собой полиморфизм?
10. Что представляет собою наследование?
11. Каковы преимущества разделения программы на подпрограммы (модули)?
12. Что такое подпрограмма?
13. Что представляет собой рекурсивная функция?
14. Что такое класс в концепции программы?
15. Каковы составляющие класса программы?
16. Чем объект отличается от класса?

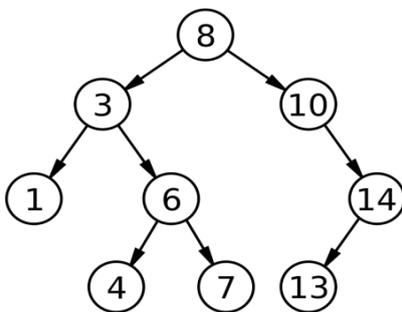
17. Что такое тип данных в программировании?
18. Что такое стек?
19. Что такое очередь?
20. Какая структура данных изображена на рисунке?



21. Какая структура данных изображена на рисунке?



22. Какая структура данных изображена на рисунке?



23. Какая структура данных изображена на рисунке?



24. Даны три целых числа: А, В, С. Как можно записать в виде условного на языках С++/С# выражения следующее утверждение: «Двойное неравенство $A < B < C$ »?
25. Даны три целых числа: А, В, С. Как можно записать на языках С++/С# в виде условия следующее утверждение: «Хотя бы одно из чисел А, В, С положительное»?
26. Какие спецификации доступа к членам класса на языках С++/С# Вы знаете?
27. Дайте определение конструктора класса. Каково назначение конструктора?
28. Сколько конструкторов может быть в классе? Какие виды конструкторов создаются по умолчанию?
29. Что выполнит данная программа?


```
#include <iostream>
```



```

using namespace std;
int length(char *s)
{int i;
  for(i=0;
    *s!='\0';s
    ++i)++;
  return(i);
}
int main()
{ char
  s[]="lalalalalalalalal
  ala";cout<<length(s);
  return 0;}

```

30. Что выполнит данная

программа?#include

```

<iostream>
#include
<fstream>
using
namespace
std;int
main()
{
  ifstream
  in_file("f.txt"
  );char x;
  if (in_file.fail())
  {
    cout << "Error";
    exit(1);
  }
int
  cou
  nt1
  = 0;
  int
  cou
  nt2
  = 0;
  while (in_file >> x)
  {
    if (x == '{') count1++;
    if (x == '}') count2++;
  }
  in_file.close();
  if (count1 == count2)
    cout << "Program good!";
}

```



Контрольно-оценочные средства для промежуточной аттестации

1. Перечислите этапы решения задач в правильной последовательности

- 1) построение алгоритма, математическая формализация, постановка задачи, перевод алгоритма на язык программирования, отладка и тестирование программы
- 2) построение алгоритма, перевод алгоритма на язык программирования, постановка задачи, математическая формализация, отладка и тестирование программы
- 3) постановка задачи, математическая формализация, построение алгоритма, перевод алгоритма на язык программирования, отладка и тестирование программы

2. Алгоритмом можно назвать:

- 1) описание решения квадратного уравнения;
- 2) расписание уроков в колледже;
- 3) технический паспорт автомобиля;
- 4) список группы в журнале.

3. Запись алгоритма с использованием фраз естественного и алгоритмического языка называется:

- 1) словесным;
- 2) графическим;
- 3) псевдокодом.

4. Алгоритмическая конструкция, предполагающая выполнение либо одного, либо другого действия в зависимости от истинности или ложности условия, называется:

- 1) линейной;
- 2) ветвлением;
- 3) циклической

5. Многократно повторяющаяся часть алгоритма называется:

- 1) параметром цикла;
- 2) телом цикла;
- 3) перебором.

6. Алгоритм какого типа изображен на блок-схеме



- 1) циклический;
- 2) разветвляющийся;
- 3) вспомогательный;
- 4) линейный;

7. Чем характеризуется переменная?

- 1) именем, типом, значением
- 2) именем, значением
- 3) значением, типом

8. Выражению $|x^2 - x^3| - \frac{7x}{x^3 - 15x}$ соответствует следующая запись на языке программирования:

1) $(x^2 - x^3) - 7*x/x^3 - 15*x$	2) $(x^2 - x^3) - (7*x)/(x^3 - 15)$
3) $ABS(x^2 - x^3) - 7*x/(x^3 - 15)*x$	4) $ABS(x^2 - x^3) - (7*x)/(x^3 - 15*x)$

9. Для понятности текстов программ в имени целых переменных включают приставку:

- 1) sng;
- 2) int;
- 3) str.

10. Каков объем занимает в памяти переменная, объявленная как Long:

- 1) 2 байта;
- 2) 4 байта;
- 3) 1 байт;
- 4) 8 байт.



11. Программа, представленная следующим кодом, называется:

```
strA = "Компьютер"  
MID(strA,2,1) = "o"
```

Print strA

- 1) линейной 2) разветвляющейся 3) циклической 4) вспомогательной

12. В результате выполнения команд переменная **bytA** получит значение:

```
bytA = LEN ("Язык программирования")
```

Print bytA

- 1) " Язык программирования " 2) 21
3) Язык программирования 4) 20

13. В результате преобразований , описанных в программном коде, в переменную **strA** будет записано:

```
strC = "база исходных данных"
```

```
strA = MID(strC,1,5)+ MID(strC,15,6)
```

- 1) база 2) данных база 3) база данных 4) данных

14. Индекс числового массива указывает на :

- 1) количество элементов в массиве ;
2) количество свободных ячеек под данный массив;
3) место элемента в числовом ряду;
4) номер числового ряда, в котором находится элемент.

15. Дана матрица , состоящая из 4 строк и 5 столбцов. Чему равен A (4,3) элемент :

24	13	1	45	15
3	16	8	9	52
7	90	23	8	31
12	43	6	16	5

- 1) 16 2) 6 3) 8 4) 23

16. В данном фрагменте программы подсчитывается

```
S = 0
```

```
FOR I = 1 To 20 STEP 2
```

```
S = S + A( I )
```

```
NEXT I
```

- 1) сумма значений всех элементов; 2) сумма четных значений элементов;
3) сумма значений четных элементов; 4) сумма значений нечетных элементов.

17. Укажите вид массива и тип элементов массива, если оператор объявления массива имеет вид :

```
Dim MAS (1 TO 9 , 1 TO 4 ) As string
```

- 1) одномерный массив целых чисел;
2) двумерный массив строковых данных;
3) двумерный массив вещественных чисел.

18. Каковы отличия функции от процедуры?

- 1) Процедура может иметь только одно значение, как и функция.
2) Функция может иметь несколько значений, а процедура только одно (оно и будет ее результатом).
3) Процедура может иметь несколько значений, а функция только одно (оно и будет ее результатом).

19. При работе с процедурами как связаны между собой формальные и фактические параметры?

- 1) Количество формальных и фактических параметров могут не совпадать
2) Порядок следования и количество формальных и фактических параметров совпадают
3) имена формальных и фактических должны совпадать



20. Какие переменные называются локальными?

- 1) Переменные, которые объявлены внутри подпрограммы и они могут быть использованы как внутри данной подпрограммы, так и в основной программе
- 2) Переменные, которые объявлены в основной программе и их могут использовать любые процедуры и функции данной программы.
- 3) Переменные, которые объявлены внутри подпрограммы и они могут быть использованы только внутри данной подпрограммы.

21 . Для чего предназначены базы данных?

- 1) для выполнения вычислений на компьютере;
- 2) для принятия управленческих решений;
- 3) для хранения, обновления и поиска данных.

22. Базы данных, в которых для связи между объектами используются структуры в виде деревьев, называются:

- 1) иерархические;
- 2) сетевые;
- 3) реляционные;
- 4) обычные.

23. Без каких объектов не может существовать база данных:

- 1) без форм
- 2) без таблиц
- 3) без запросов
- 4) без отчетов

24. Содержит ли какую-либо информацию таблица БД, в которой нет полей?

- 1) содержит информацию о структуре базы данных
- 2) содержит информацию о будущих записях
- 3) таблица без полей существовать не может

25. Таблицы в базах данных предназначены:

- 1) для отбора и обработки данных базы
- 2) для автоматического выполнения группы команд
- 3) для ввода данных базы и их просмотра
- 4) для хранения данных базы

26. Количество цифр, используемых для записи числа в системе счисления, называют:

- 1) коэффициентом;
- 2) основанием
- 3) разрядом;
- 4) кодом

27. Перевести число из десятичной системы счисления в двоичную $20,20_{10} \rightarrow X_2$

- 1) $10100,001_2$
- 2) $11001,001_2$
- 3) $100111,01_2$
- 4) $11101,01_2$

28. Перевести число из двоичной системы счисления в десятичную $110,11_2 \rightarrow X_{10}$

- 1) $3,02_{10}$
- 2) $6,75_{10}$
- 3) $5,75_{10}$
- 4) $12,02_{10}$



№ вопроса	Правильный вариант ответа	
	1 вариант	2 вариант
1	1	3
2	2	1
3	2	3
4	3	2
5	1	2
6	4	2
7	2	1
8	3	4
9	4	2
10	2	2
11	3	1
12	3	2
13	2	3
14	4	3
15	2	2
16	3	4
17	2	2
18	2	3
19	3	2
20	2	3
21	1	3
22	3	1
23	4	2
24	2	3
25	1	4
26	2	2
27	4	1
28	2	3
29	1	4
30	3	1



Экзаменационные билеты

Специальность 09.02.07 Информационные системы и программирование ОП.4. Основы алгоритмизации и программирования

Экзамен

Проверяемые компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9,
ОК 10, ПК 1.1- ПК 1.5, ПК 2.4 - 2.5

Билет №1

1. Логические операции. Построение логических выражений
2. Описать процедуру Swap(X, Y), меняющую содержимое переменных X и Y (X и Y — целочисленные параметры, являющиеся одновременно входными и выходными). С ее помощью для данных переменных A, B, C, D последовательно поменять содержимое следующих пар: A и B, C и D, B и C и вывести новые значения A, B, C, D.

Преподаватель _____

Председатель ПЦК _____

Специальность 09.02.07 Информационные системы и программирование ОП.4. Основы алгоритмизации и программирования

Экзамен

Проверяемые компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9,
ОК 10, ПК 1.1- ПК 1.5, ПК 2.4 - 2.5

Билет №2

1. Перечислите не менее 7 методов строк. Дайте их описание.
2. Даны катеты прямоугольного треугольника a и b. Найти его гипотенузу c и периметр P:
 $c = \sqrt{a^2 + b^2}, \quad P = a + b + c.$

Преподаватель _____

Председатель ПЦК _____



**Специальность 09.02.07 Информационные системы и программирование
ОП.4. Основы алгоритмизации и программирования**

Экзамен

Проверяемые компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9,
ОК 10, ПК 1.1- ПК 1.5, ПК 2.4 - 2.5

Билет №3

1. Что означает термин объектно-ориентированное программирование? Опишите основные принципы ООП.
2. Описать функцию CircleS(R) вещественного типа, находящую площадь круга радиуса R (R — вещественное). С помощью этой функции найти площади трех кругов с данными радиусами. Площадь круга радиуса R вычисляется по формуле $S = \pi R^2$.

Преподаватель _____

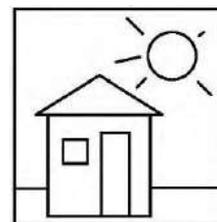
Председатель ПЦК _____

**Специальность 09.02.07 Информационные системы и программирование
ОП.4. Основы алгоритмизации и программирования**

Экзамен

Проверяемые компетенции ОК 1, ОК 2, ОК 4, ОК 5, ОК 9,
ОК 10, ПК 1.1- ПК 1.5, ПК 2.4 - 2.5

1. Опишите жизненный цикл программы. Перечислите основные этапы решения задач на компьютере.
2. Нарисовать на холсте главного окна приложения рисунок по образцу используя команды программируемой графики



Преподаватель _____

Председатель ПЦК _____

