|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ | | Ульяновский авиационный колледж | | |
| Профессиональный цикл | | |
| ОП.05 Основы программирования  *Практические и лабораторные работы*  для специальности СПО  09.02.03 Программирование в компьютерных системах  (базовый уровень)  Ульяновск  2016 | | |
| ОДОБРЕНО  на заседании ЦМК  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Протокол № \_\_\_  от «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_\_г.  Председатель ЦМК  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/  Подпись Ф.И.О. | | УТВЕРЖДАЮ  Зам. директора по УПР  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/  Подпись Ф.И.О.  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 20 \_\_ г. |
| РАЗРАБОТЧИК: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |

Методические указания для студентов по выполнению практических и лабораторных работ содержат цели, формируемые образовательные результаты, краткие теоретические сведения, методические указания к заданиям, контрольные вопросы для проверки по учебной дисциплине «Основы программирования»

Данные методические указания составлены в соответствии с ФГОС СПО по специальности по специальности 09.02.03 Программирование в компьютерных системах (базовый уровень)

# **СОДЕРЖАНИЕ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  | | стр |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА | | | | |  |
|  | | | | |  |
| ПРАКТИЧЕСКИЕ РАБОТЫ | | | | |  |
|  | ПР 1 | | | Составление и отладка линейных программ |  |
|  | ПР 2 | | | Составление и отладка линейных программ с вводом и выводом данных |  |
|  | ПР 3 | | | Составление и отладка программ с ветвлениями |  |
|  | ПР 4 | | | Составление и отладка программ с циклами |  |
|  | ПР 5 | | | Составление и отладка программ с массивами |  |
|  | ПР 6 | | | Составление и отладка программ со строками |  |
|  | ПР 7 | | | Составление и отладка программ с файлами |  |
|  | ПР 8 | | | Составление программ сортировки структур данных |  |
|  | ПР 9 | | | Составление программ поиска данных в структурах |  |
|  | ПР 10 | | | Составление и отладка программ с функциями пользователя |  |
|  | ПР 11 | | | Составление и отладка программ с процедурами |  |
|  | ПР 12 | | | Разработка классов и создание объектов в соответствии с принципами ООП |  |
|  | ПР 13 | | | Разработка оконного приложения в среде Delphi |  |
|  | ПР 14 | | | Разработка СУБД в среде Delphi |  |
|  |  | | |  |  |
| ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА | | | | |  |

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данные методические рекомендации предназначены для студентов по выполнению практических и лабораторных работ УД ОП.05 Основы программирования

обучающихся на специальности СПО 09.02.03 Программирование в компьютерных системах (базовый уровень)

На УД ОП.05 Основы программирования

формируются следующие образовательные результаты:

|  |  |
| --- | --- |
|  | УМЕНИЯ |
| У 1 | работать в среде программирования |
| У 2 | реализовывать построенные алгоритмы в виде программ на конкретном языке программирования |
|  |  |
|  | ЗНАНИЯ |
| З 1 | этапы решения задач на компьютере |
| З 2 | типы данных |
| З 3 | базовые конструкции изучаемых языков программирования |
| З 4 | принципы структурного и модульного программирования |
| З 5 | принципы объектно-ориентированного программирования |
|  |  |
|  | ПРОФЕССИОНАЛЬНЫЕ КОМПЕТЕНЦИИ |
| ПК 1.1 | Выполнять разработку спецификаций отдельных компонент |
| ПК 1.2 | Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля |
| ПК 1.3 | Выполнять отладку программных модулей с использованием специализированных программных средств |
| ПК 1.4 | Выполнять тестирование программных модулей |
| ПК 1.5 | Осуществлять оптимизацию программного кода модуля |
| ПК 3.1 | Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения |
|  |  |
|  | ОБЩИЕ КОМПЕТЕНЦИИ |
| ОК1 | Участвовать в мероприятиях профессиональной направленности |
| ОК 2 | Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. |
| ОК 3 | Использовать информационно-коммуникационные технологии  в профессиональной деятельности |
| ОК 4 | Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. |
| ОК 5 | Выбирать адекватные ситуациям методы и средства контроля, оценки и коррекции собственной деятельности с информационными источниками;  Осуществлять анализ и оценку информации с использованием информационно-коммуникационных технологий (электронно-методические комплекты, интернет-ресурсы, электронные носители и т.д.) |
| ОК 6 | Использовать информационно – коммуникационные технологии в профессиональной деятельности. |
| ОК7 | Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями. |
| ОК8 | Организовывать самообразование (повышение квалификации) в соответствии с выбранными направлениями |
| ОК9 | Быть готовым к смене технологий в профессиональной деятельности. |

Для данной УД ОП.05 Основы программирования, где формируются следующие образовательные результаты:

|  |  |  |
| --- | --- | --- |
| №  практ.  Лабор. | Наименование  практического занятия | Формируемые образовательные  результаты ( У, З, ОК, ПК) |
| ПР 1 | Составление и отладка линейных программ | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 2 | Составление и отладка линейных программ с вводом и выводом данных | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 3 | Составление и отладка программ с ветвлениями | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 4 | Составление и отладка программ с циклами | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 5 | Составление и отладка программ с массивами | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 6 | Составление и отладка программ со строками | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 7 | Составление и отладка программ с файлами | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 8 | Составление программ сортировки структур данных | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 9 | Составление программ поиска данных в структурах | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 10 | Составление и отладка программ с функциями пользователя | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 11 | Составление и отладка программ с процедурами | У1, У2, З1-З4, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 12 | Разработка классов и создание объектов в соответствии с принципами ООП | У1, У2, З1-З3, З5, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 13 | Разработка оконного приложения в среде Delphi | У1, У2, З1-З3, З5, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |
| ПР 14 | Разработка СУБД в среде Delphi | У1, У2, З1-З3, З5, ПК1.1-ПК1.5, ПК3.1, ОК1-9 |

ОТЧЕТ по каждой практической или лабораторной работе составляется на отдельных листах формата А4 по следующему образцу:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *5*  *40* | **Отчет по практической работе**  ЦЕЛЬ:  ЗАДАНИЯ ( номер, условие каждого задания выполненное в виде постановки задачи, распечатка текста программы, данные для тестирования задачи, результаты тестирования, результат выполнения программы.)  1 – Фамилия разработчика  2 – Фамилия проверяющего  3 – Код-обозначение документа (практической или лабораторной работы)  4 – Тема практической (или лабораторной) работы  5 – Учебное заведение, группа  6 – Литер – у (учебный документ)  7 - № листа данного документа  8 – Количество листов в данном документе  *1 2 3 4 5 6 7 8*  *185*  *7 10 23 15 10 120* | | | | | | | | | | |
|  |  |  |  |  | *15*  *15 15 20*  *15 5 5* | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  | | *Фамилия* | *Подпись* | *Дата* | *70* | *Лит.* | | | *Лист* | *Листов* |
| *Разработ.* | |  |  |  |  | *у* |  |  |  |
| *Проверил* | |  |  |  | 50 | | | | |
|  | |  |  |  |
|  | |  |  |  |
|  | Оформление практической работы  **ПР** – практическая работа  **Код специальности 09.02.03** – Программирование в компьютерных системах  **0000-00** – индекс дисциплины по учебному плану:  ОП.05 – Теория алгоритмов  **№ группы** 14 (для 14пк-1), 15 (для 15пк-1), 16(для 16пк-1) и т.д.  **№ варианта** – номер студента по списку в журнале  **№ практической** – порядковый номер практической работы  **00 год** – 16 (для 2016), 17 (для 2017) и т.д.  *ПР. 09.02.03 ОП.05. 00. 00. 00. 00.*  *код индекс № № № год*  *специальности дисциплины, группы работы варианта* | | | | | | | | | | |
|  |  |  |  |  | ПР-09.02.03-ОП.05-14-00-01-16 | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
|  | | *Фамилия* | *Подпись* | *Дата* | *Название практической работы* | *Лит.* | | | *Лист* | *Листов* |
| *Разработ.* | | *Студент* |  |  |  | *у* |  | *1* | *1* |
| *Проверил* | | *преподаватель* |  |  | УАвиаКгр. 14пк-1 | | | | |
|  | |  |  |  |
|  | |  |  |  |

**ПРАКТИЧЕСКАЯ РАБОТА №1**

**Составление и отладка линейных программ**

* **ЦЕЛЬ РАБОТЫ**

*Формирование навыков работы в средах PascalABC.NET и TurboC.*

*Формирование навыков составления, отладки и тестирования линейных программ на языках Паскаль и Си.*

* **ОБЕСПЕЧЕННОСТЬ ЗАНЯТИЯ**

*- Среда программирования PascalABC.NET*

*- Среда программирования TurboC*

* **ЛИТЕРАТУРА**
* СОДЕРЖАНИЕ ЗАДАНИЙ

1. Составление программы определения значения заданной функции при заданном значении аргумента.
2. Составление программы определения принадлежности точки, заданной ее координатами, фигуре на плоскости, заданной графически.
3. Составление программы решения математической задачи.

* ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЙ

*Задания выполняются в любом порядке*

* **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Структура программы на языке Паскаль

{ДЕКЛАРАЦИОННАЯ ЧАСТЬ}

program имя\_программы; {раздел заголовка}

uses {раздел подключения библиотек}

label {раздел меток}

const {раздел констант}

type {раздел пользовательских типов}

var {раздел переменных}

procedure {процедура пользователя}

function {функция пользователя}

{ИСПОЛНЯЕМАЯ ЧАСТЬ}

BEGIN

Оператор;

{комментарий}

[\\комментарий](file:///\\комментарий) в одну строку

END.

Структура программы на языке Си

/\*Директивы препроцессора\*/

#include <имя библиотеки> /\*подключение библиотек\*/

Тип\_результата Имя (аргументы)**;** /\*заголовки функций пользователя\*/

Тип\_результата main (аргументы) /\*заголовок главной функции\*/

{

/\*Исполняемая часть главной функции\*/

}

Тип\_результата Имя (аргументы)

{

/\* Исполняемая часть функции пользователя\*/

}

* **ЗАДАНИЯ И ИНСТРУКЦИИ ПО ВЫПОЛНЕНИЮ**

***Задание 1***

Составьте программы на языках Паскаль и Си для реализации алгоритма решения задачи. Подтвердите правильность работы программы тестом. Получите и запишите результат решения задачи.

***Задача:***

***(по заданию 1 ПР1 УД ОП.08 Теория алгоритмов)***

Вычислите значение функции y(x), при заданном значении х.

1. 
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 
14. 
15. 
16. 
17. 
18. 
19. 
20. 
21. 
22. 
23. 
24. 
25. 
26. 
27. 
28. 
29. 
30. 

***Задание 2***

Составьте программы на языках Паскаль и Си для реализации алгоритма решения задачи. Подтвердите правильность работы программы тестом. Получите и запишите результат решения задачи.

***Задача:***

***(по заданию 2 ПР1 УД ПО.08 Теория алгоритмов)***

Запишите выражение, зависящее от координат точки, и принимающее значение ИСТИНА, если точка принадлежит заштрихованной области, и ЛОЖЬ, если не принадлежит. Определите принадлежность точки заштрихованной области.

1. Координаты исследуемой точки: (0,5;0,5)

Область задана на рисунке 1.

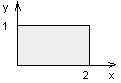
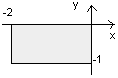
  

Рисунок 1 Рисунок 2 Рисунок 3

1. Координаты исследуемой точки: (1,5;0,5)

Область задана на рисунке 2.

1. Координаты исследуемой точки: (0,2;0,9)

Область задана на рисунке 3.

1. Координаты исследуемой точки: (0,75;-0,3)

Область задана на рисунке 4.

1. Координаты исследуемой точки: (0,2;0,45)

Область задана на рисунке 5.

Рисунок 4 Рисунок 5 Рисунок 6 Рисунок 7

1. Координаты исследуемой точки: (0,5;-2,5)

Область задана на рисунке 6.

1. Координаты исследуемой точки: (0,2;0,7)

Область задана на рисунке 7.

1. Координаты исследуемой точки: (1,0;1,5)

Область задана на рисунке 8.

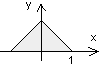
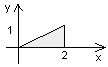
   

Рисунок 8 Рисунок 9 Рисунок 10 Рисунок 11

1. Координаты исследуемой точки: (-0,5;0,9)

Область задана на рисунке 9.

1. Координаты исследуемой точки: (1,5;0,0)

Область задана на рисунке 10.

1. Координаты исследуемой точки: (0,5;1,2)

Область задана на рисунке 11.

1. Координаты исследуемой точки: (-1,5;0,75)

Область задана на рисунке 12.

Рисунок 12 Рисунок 13 Рисунок 14 Рисунок 15

1. Координаты исследуемой точки: (0,5;1,5)

Область задана на рисунке 13.

1. Координаты исследуемой точки: (3,5;-0,5)

Область задана на рисунке 14.

1. Координаты исследуемой точки: (-0,3;1,7)

Область задана на рисунке 15.

1. Координаты исследуемой точки: (-0,5;1,3)

Область задана на рисунке 16.

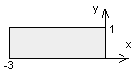
  

Рисунок 16 Рисунок 17 Рисунок 18

1. Координаты исследуемой точки: (1,3;-1,2)

Область задана на рисунке 17.

1. Координаты исследуемой точки: (-1,7;1,3)

Область задана на рисунке 18.

1. Координаты исследуемой точки: (-0,7;0,9)

Область задана на рисунке 19.

1. Координаты исследуемой точки: (-0,8;0,7)

Область задана на рисунке 20.

1. Координаты исследуемой точки: (1,3;-1,6)

Область задана на рисунке 21.

Рисунок 19 Рисунок 20 Рисунок 21 Рисунок 22

1. Координаты исследуемой точки: (-0,3;-0,8)

Область задана на рисунке 22.

1. Координаты исследуемой точки: (-0,6;-0,2)

Область задана на рисунке 23.

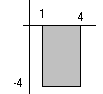
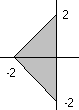
   

Рисунок 23 Рисунок 24 Рисунок 25 Рисунок 26

1. Координаты исследуемой точки: (-0,6;-0,2)

Область задана на рисунке 24.

1. Координаты исследуемой точки: (2,3;-3,5)

Область задана на рисунке 25.

1. Координаты исследуемой точки: (-1,3;-1,5)

Область задана на рисунке 26.

1. Координаты исследуемой точки: (0,9;-0,4)

Область задана на рисунке 27.

Рисунок 27 Рисунок 28 Рисунок 29 Рисунок 30

1. Координаты исследуемой точки: (2,3;1,5)

Область задана на рисунке 28.

1. Координаты исследуемой точки: (-0,7;-0,5)

Область задана на рисунке 29.

1. Координаты исследуемой точки: (2,3;-3,5)

Область задана на рисунке 30.

***Задание 3***

Составьте программы на языках Паскаль и Си для реализации алгоритма решения задачи. Подтвердите правильность работы программы тестом. Получите и запишите результат решения задачи.

***Задача:***

***(по заданию 3 ПР1 УД ПО.08 Теория алгоритмов)***

1. Вычислить периметр и площадь прямоугольного треугольника по заданным длинам двух катетов a и b .
2. Заданы координаты трех вершин треугольника. Найти его периметр и площадь.
3. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .
4. Вычислить расстояние между двумя точками с данными координатами.
5. Даны два действительных числа x и y . Вычислить их сумму, разность, произведение и частное.
6. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
7. Дана сторона равностороннего треугольника. Найти площадь этого треугольника, его высоты, радиусы вписанной и описанной окружностей.
8. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
9. Найти площадь кольца, внутренний радиус которого равен r , а внешний – заданному числу R. R >r .
10. Треугольник задан величинами своих углов и радиусом описанной окружности. Найти стороны треугольника.
11. Найти площадь равнобедренной трапеции с основаниями a и b и углом α при большем основании a.
12. Заданы координаты вектора, вычислить его длину.
13. Заданы прямая ax+by+c=0 и точка (x1;y1). Вычислить расстояние от точки до прямой.
14. Найти площадь треугольника, две стороны которого равны a и b , а угол между этими сторонами q.
15. В прямоугольном треугольнике с катетами a и b вычислить длину высоты проведенной к гипотенузе.
16. Найти сумму членов арифметической прогрессии, если известны ее первый член, знаменатель и число членов прогрессии.
17. Найти все углы треугольника со сторонами a, b , c. Предусмотреть в программе перевод радианной меры угла в градусы.
18. Три сопротивления 1 R , 2 R , 3 R соединены параллельно. Найдите сопротивление соединения.
19. Текущее показание электронных часов: m часов, n минут, k секунд. Какое время будут показывать часы через p ч q мин r c ?
20. Составить программу вычисления объема цилиндра и конуса, которые имеют одинаковую высоту H и одинаковый радиус основания R .
21. Даны длины трех отрезков a, b и с. Может ли существовать треугольник со сторонами равными длинам этих отрезков?
22. Дана величина A , выражающая объем информации в байтах. Перевести A в более крупные единицы измерения информации.
23. Составить программу для вычисления пути, пройденного лодкой, если ее скорость в стоячей воде v км/ч, скорость течения реки w км/ч, время движения по озеру 1 t ч, а против течения реки – 2 t ч.
24. Двоичное число записано 5 знаками. Определите запись этого числа в десятичной системе счисления.

* **МЕТОДИКА АНАЛИЗА РЕЗУЛЬТАТОВ, ОБРАЗЕЦ ОТЧЕТА**

***Задание 1***

***Вариант 0***

Вычислите значение функции Y(x)=x2-7x+6, при заданном значении х=0,23.

*Программа на языке Паскаль.*

**program** PR1\_1;

**var** x,y:real;

**BEGIN**

writeln('Введите значение аргумента x');

readln(x);

y:=x\*x-7\*x+6;

writeln('y=',y:6:2);

**END**.

*Тестирование программы на языке Паскаль.*

|  |  |  |
| --- | --- | --- |
| Вводимое значение аргумента х | Ожидаемое значение функции Y(x) | Полученное значение  (на экране) |
| 1 | 0 | y= 0.00 |

Вывод: так как полученный результат совпал с ожидаемым, то программа работает верно.

*Результат работы программы (решения задачи).*

При заданном значении аргумента х=0,23 получено значение функции y= 4.44.

*Программа на языке Си.*

#include <stdio.h>

#include <math.h>

int main (viod)

{

float x,y;

printf("vvedite x\n");

scanf("%f",&x);

y=x\*x-7\*x+6;

printf("y=%6.2f\n",y);

}

*Тестирование программы на языке Си.*

|  |  |  |
| --- | --- | --- |
| Вводимое значение аргумента х | Ожидаемое значение функции Y(x) | Полученное значение  (на экране) |
| 1 | 0 | y= 0.00 |

*Результат работы программы (решения задачи).*

При заданном значении аргумента х=0,23 получено значение функции y= 4.44.

***Задания 2 и 3 оформляются аналогично.***

* КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите разделы линейной программы на языке Паскаль.
2. Перечислите разделы линейной программы на языке Си.
3. Определите понятие тип данных.
4. Перечислите простые типы данных языка Паскаль.
5. Перечислите простые типы данных языка Си.

* ПРИЛОЖЕНИЯ

Назначение функциональных клавиш в интегрированных средах

|  |  |  |
| --- | --- | --- |
| Клавиша | Команда меню | Функция |
| F1 | Help | Вызов окна справки |
| F2 | File >Save | Быстрое сохранение |
| F3 | File>Open… | Вызов диалога открытия файла |
| F4 | Run>Go to cursor | Выполнить программу до строки с курсором |
| F5 | Window>Zoom | Развернуть текущее окно на всю рабочую область |
| F6 | Window>Next | Перейти к следующему окну |
| F7 | Run>Trace into | Выполнить программу по шагам с заходом в подпрограммы |
| F8 | Run>Step over | Выполнить программу по шагам без захода в подпрограммы |
| F9 | Compile>Make | Компилировать программу, не выполнять |

Назначение комбинаций (горячих) клавиш в интегрированных средах

|  |  |  |
| --- | --- | --- |
| Комбинация клавиш | Команда меню | Функция |
| CTRL+F9 | Run>Run | Компилировать и выполнить программу |
| ALT+F5 | Window>User screen | Открыть окно результатов для просмотра |
| Alt+Литера выделенная красным цветом |  | Выбрать соответствующий пункт меню пользователя |
| Alt+Enter |  | Свернуть / развернуть окно |
| Alt+X |  | Закрыть окно среды разработки |

**ПРАКТИЧЕСКАЯ РАБОТА №2**

**Составление и отладка линейных программ с вводом и выводом данных**

* **ЦЕЛЬ РАБОТЫ**

*Формирование навыков работы в средах PascalABC.NET и TurboC.*

*Формирование навыков составления, отладки и тестирования линейных программ с вводом и выводом данных на языках Паскаль и Си.*

* **ОБЕСПЕЧЕННОСТЬ ЗАНЯТИЯ**

*- Среда программирования PascalABC.NET*

*- Среда программирования TurboC*

* **ЛИТЕРАТУРА**
* СОДЕРЖАНИЕ ЗАДАНИЙ

1. Форматирований вывод данных в языке Си.
2. Результаты функции вывода в языке Си.
3. Форматированный вывод с указанием формата с клавиатуры.
4. Решение расчетной задачи.

* ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЙ

*Задания выполняются в любом порядке*

* **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Язык Паскаль содержит четыре оператора ввода/вывода: read, readln, write, writeln. Каждый оператор ввода (read, readln) должен содержать список ввода – имена переменных, значения которых задаются с клавиатуры. Каждый оператор вывода (write, writeln) может содержать список выводимых элементов. Элементами вывода могут быть: константы, переменные, выражения. Константа выводиться на экран без изменения. Для переменной – выводиться ее значение. Выражение сначала исполняется, а результат выполнения выводиться на экран.

Для форматирования вывода в языке Паскаль можно использовать запись:

Write(a:число1:число2);

где число1 –минимальная ширина поля вывода – число экранных знаков под вывод значения;

число2 – применяется только для дробных чисел, и означает число знаков после запятой.

Функции ввода и вывода в языке Си собраны в библиотеку stdio.h. Основными функциями являются: для ввода – scanf(); для вывода – printf();

Аргументами функций являются: управляющая строка и список ввода или вывода.

Управляющая строка заключается в кавычки и содержит указания на тип и формат вводимых/выводимых значений. Список ввода/вывода содержит переменные для ввода/вывода.

В управляющей строке могут использоваться следующие спецификаторы.

|  |  |
| --- | --- |
| Спецификаторы | Действия |
| %a, %A | возвращают число с плавающей точкой и 16-е цифры |
| %c | возвращает одиночный символ |
| %d, %i | возвращают целое число в десятичной форме |
| %e, %E | возвращают число с плавающей точкой в экспоненциальной форме |
| %f | возвращает число с плавающей точкой в десятичной форме |
| %g | использует спецификатор %f или %e в зависимости от значения (выбирается автоматически) |
| %o | возвращает восьмеричное целое число без знака |
| %p | вывод указателя в шестнадцатеричном формате |
| %s | возвращает строку символов |
| %u | возвращает десятичное целое число без знака |
| %x, %X | возвращают шестнадцатеричное целое число без знака |
| %% | вывод на печать символа % |

В управляющей строке могут использоваться следующие модификаторы и флаги.

|  |  |
| --- | --- |
| Модификаторы | Действия |
| цифра/цифры | минимальная ширина поля вывода (количество экранных символов) |
| .цифра | точность |
| h | используется при целочисленных преобразованиях для кодирования значений short int и unsigned short int |
| hh | используется при целочисленных преобразованиях для кодирования значений signed char и unsigned char |
| j | используется при целочисленных преобразованиях для получения типов int\_max\_t и uint\_max\_t |
| l | используется при целочисленных преобразованиях для получения типа long int и unsigned long int |
| ll | используется при целочисленных преобразованиях для получения типа long long int и unsigned long long int |
| t | используется с целью кодирования значений типа ptroliff\_t, который является разностью двух указателей |
| z | используется с целью кодирования значений size\_t |
| L | используется при преобразовании чисел с плавающей точкой типа long double |
| Флаг + | значения со знаком печатаются со знаком "+" или "-" |
| флаг - | элемент выравнивается по левому краю |
| Флаг пробел | значение со знаком "+" печатается с пробелом, но без знака |
| флаг # | для спецификаторов %o, %x, %X выводит не значащие нули. Для форм с плавающей точкой гарантировано печатается точка, даже если десятичных знаков нет |
| Флаг 0 | для числовых форм вместо пробелов на свободные экранные места выводятся нули |

* **ЗАДАНИЯ И ИНСТРУКЦИИ ПО ВЫПОЛНЕНИЮ**

***Задание 1***

Наберите текст программы в компиляторе С. Выполните программу. Проанализировав результат, сделайте выводы о работе модификаторов.

В отчет поместите распечатанный текст программы. Для результатов разлинуйте (или используйте клетчатый) лист. Перепишите результаты с экрана на этот лист, сохраняя размещение символов.

**1**

# include <stdio.h>

int main (void)

{

int p;

p=732;

printf (“\***%**d**\*\**n”, p);

printf (“\*%2d\*\n”, p);

printf (“\*%10d\*\n”, p);

printf (“\*%-10d\*\n”, p);

}

**2**

# include <stdio.h>

int main (void)

{

float p;

p=2345.67;

printf (“\*%f\*\n”,p);

printf (“\*%e\*\n”,p);

printf (“\*%g\*\n”,p);

printf (“\*%10.3f\*\n”,p);

printf (“\*%3.1f\*\n”, p);

printf (“\*%14.3f\*\n”, p);

printf (“\*%14.3e\*\n”, p);

printf (“\*%-14.2f\*\n”, p);

printf (“\*%+14.2f\*\n”, p);

printf (“\*%014.2f\*\n”, p);

}

**3**

# include <stdio.h>

int main (void)

{

printf (“\*%d\*%x\*%X\*%#x\*\n”, 31, 31, 31, 31);

printf (“\*%5d\*%5.3d\*%05d\*%05.3d\*\n”, 6, 6, 6, 6);

}

**4**

# include <stdio.h>

int main (void)

{

char a=”Prosto stroka”;

printf (“\*%2s\*\n”, a);

printf (“\*%24s\*\n”, a);

printf (“\*%24.5s\*\n”, a);

printf (“\*%-24.5s\*\n”, a);

}

***Задание 2***

Наберите текст программы в компиляторе языка С. Сделайте вывод о значении возвращаемом функции printf().

В отчет поместите распечатанный текст программы. Для результатов разлинуйте (или используйте клетчатый) лист. Перепишите результаты с экрана на этот лист, сохраняя размещение символов.

#include <stdio.h>

int main(void)

{

int t=212;

int r;

r=printf (“%d F соответствует точке кипения воды.\n”, t);

printf (“Функция printf() выводит %d символов.\n”, r);

}

***Задание 3***

Наберите текст программы в компиляторе языка С. Сравните работу операторов ввода и вывода.

В отчет поместите распечатанный текст программы. Для результатов разлинуйте (или используйте клетчатый) лист. Перепишите результаты с экрана на этот лист, сохраняя размещение символов.

#include <stdio.h>

int main (void)

{

unsigned width, precision;

int number=256;

double weight=242.5;

printf(“какова ширина поля?\n”);

scanf (“%d”, &width);

printf (“Значение будет:%\*d:\n”, width, number);

printf (“Теперь укажите ширину и точность:\n”);

scanf (“%d %d”, &width, &precision);

printf (“Значение будет: %\*.\*f\n”, width, precision, weight);

}

***Задание 4***

Составите математическую модель задачи.

Составьте алгоритм решения.

Подготовьте два тестовых значения.

Запишите программы на языках Паскаль и Си, выполните их тестирование.

Все элементы решения и результаты занесите в отчет.

***Задача:***

Запросите у пользователя пройденный транспортным средством километраж и количество потребленного топлива. Полученные данные обработайте с последующим отображением количества миль, преодоленных при запасе топлива в один галлон, причем справа от запятой используйте одну цифру. Учитывая, что один галлон равен 3, 785 литра, а одна миля – 1,609 километра, преобразуйте полученный результат в количество литров бензина, потребленных на 100 км пути (европейский стандарт) и отобразите его в десятичном формат (одна цифра справа от запятой). (Учтите, что в Америке измеряется расстояние, пройденное с запасом топлива в один галлон, а в Европе – количество топлива, израсходованное на 100 км пути.)

***Контрольные вопросы***

1. Перечислите операторы вывода языка Паскаль и функции вывода языка С.
2. Перечислите спецификаторы функции вывода языка С.
3. Перечислите операторы ввода языка Паскаль и функции ввода языка С.
4. Перечислите спецификаторы функции ввода языка С.
5. Назовите возвращаемые значения функций ввода и вывода.
6. Перечислите модификаторы используемые при вводе и выводе в языке С.

**ПРАКТИЧЕСКАЯ РАБОТА №3**

**Составление и отладка программ с ветвлениями.**

* **ЦЕЛЬ РАБОТЫ**

*Формирование навыков работы в средах PascalABC.NET и TurboC.*

*Формирование навыков составления, отладки и тестирования программ с ветвлениями на языках Паскаль и Си.*

* **ОБЕСПЕЧЕННОСТЬ ЗАНЯТИЯ**

*- Среда программирования PascalABC.NET*

*- Среда программирования TurboC*

* **ЛИТЕРАТУРА**
* СОДЕРЖАНИЕ ЗАДАНИЙ

1. Составление программы определения значения функции для произвольных значений аргумента
2. Составление программы решения математической задачи.

* ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЙ

*Задания выполняются в любом порядке*

* **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

В языке программирования Паскаль существуют следующие формы реализации ветвлений.

1. В неполной форме с одним действием (оператором)

IF условие THEN оператор;

1. В полной форме с одним действием (оператором)

IF условие THEN оператор1 ELSE оператор2;

1. В неполной форме с несколькими операторами

IF условие THEN

BEGIN   
                оператор1;   
                оператор2;   
                …   
                операторN;   
END;

1. В полной форме с несколькими операторами

IF условие THEN

BEGIN   
                оператор1;   
                оператор2;   
                …   
                операторN;   
END

ELSE  BEGIN   
                оператор1;   
                оператор2;   
                …   
                операторN;   
            END;

В языке программирования Си существуют следующие формы реализации ветвлений.

1. В неполной форме с одним действием (оператором)

if (условие) оператор;

1. В полной форме с одним действием (оператором)

if (условие) оператор1 else оператор2;

1. В неполной форме с несколькими операторами

if (условие)

{  
                оператор1;   
                оператор2;   
                …   
                операторN;   
}

1. В полной форме с несколькими операторами

if (условие)

{   
                оператор1;   
                оператор2;   
                …   
                операторN;   
}

else

{   
                оператор1;   
                оператор2;   
                …   
                операторN;   
  }

* **ЗАДАНИЯ И ИНСТРУКЦИИ ПО ВЫПОЛНЕНИЮ**

***Задание 1***

Составьте программу вычисления значения функции при любых значениях аргументов.

***Задача:***

***(по заданию 1 ПР3 УД ПО.08 Теория алгоритмов)***

* 1. 
  2. 
  3. 
  4. 
  5. 
  6. 
  7. 

1. .
2. 
3. 
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13.  
14. 
15. 
16. 
17. 
18. 
19. .
20. 
21. 
22. 
23. 

***Задание 2***

Составьте программу, определяющую ответ на вопрос задачи, для любых входных значений переменных.

***(по заданию 2 ПР3 УД ПО.08 Теория алгоритмов)***

1. Можно ли квадрат со стороной А описать окружностью радиусом R.
2. Дан треугольник с углами А, В, С. Может ли данный треугольник быть равносторонним.
3. Может ли тело, массой М, иметь ускорение А, если на него действует сила F.
4. Можно ли окружность радиусом R вписать в квадрат со стороной А.
5. Даны углы А, В, С. Может ли существовать треугольник с данными углами.
6. Является ли многоугольник со сторонами A, B, C, D квадратом.
7. Дан треугольник с сторонами X, Y, Z. Является ли он равнобедренным.
8. Дан треугольник с углами А, В, С. Может ли он быть прямоугольным.
9. Дан треугольник с сторонами А, В, С. Является ли он равносторонним.
10. Дан треугольник с сторонами А, В, С. Может ли он быть прямоугольным.
11. Сколько решений имеет квадратное уравнение ах2+вх+с=0.
12. Дан многоугольник со сторонами А, В, С, D. Может ли он быть ромбом.
13. Дан треугольник с углами А, В, С. Может ли он быть равнобедренным.
14. Определить, находиться ли точка М(а, в) внутри верхней части единичного круга с центром в начале координат.
15. Определить, имеется ли среди трех чисел а, в и с хотя бы одна пара равных между собой чисел.
16. Определить, имеется ли среди трех чисел а, в и с хотя бы одна пара взаимно противоположных чисел.
17. Числа а и в выражают длины катетов одного прямоугольного треугольника, а с и d – другого. Определить, являются ли эти треугольники подобными.
18. Даны два числа а и в. Определить, является ли одно из них квадратом другого.
19. Можно ли квадрат со стороной А описать окружностью радиусом R.
20. Дан треугольник с углами А, В, С. Может ли данный треугольник быть равносторонним.
21. Может ли тело, массой М, иметь ускорение А, если на него действует сила F.
22. Можно ли окружность радиусом R вписать в квадрат со стороной А.
23. Даны углы А, В, С. Может ли существовать треугольник с данными углами.
24. Является ли многоугольник со сторонами A, B, C, D квадратом.
25. Дан треугольник с сторонами X, Y, Z. Является ли он равнобедренным.
26. Дан треугольник с углами А, В, С. Может ли он быть прямоугольным.
27. Дан треугольник с сторонами А, В, С. Является ли он равносторонним.
28. Дан треугольник с сторонами А, В, С. Может ли он быть прямоугольным.
29. Сколько решений имеет линейное уравнение ах = в.
30. Определить, находиться ли точка М(а, в) внутри нижней части единичного круга с центром в начале координат.

* **МЕТОДИКА АНАЛИЗА РЕЗУЛЬТАТОВ, ОБРАЗЕЦ ОТЧЕТА**

***Задание 1***

***Вариант 0***

Составьте программу вычисления значения функции при любых значениях аргументов.



*Текст программы на языке Паскаль.*

**program** pr\_3\_1;

**var** x,y:real;

**BEGIN**

writeln('Введите значение агрумента х');

readln(x);

**if** x>5 **then**

**if** x>=8 **then**

**begin**

y:=sqrt(x-8);

writeln(y);

**end**

**else** writeln('Нет решения')

**else**

**if** x<-5 **then**

**if** x<>-7 **then**

**begin**

y:=1/(7+x)+5;

writeln(y);

**end**

**else** writeln('Нет решения')

**else**

**if** x<>3 **then**

**begin**

y:=1/(x-3);

writeln(y);

**end**

**else** writeln('Нет решения');

**END**.

*Тестирование программы на языке Паскаль.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | x>5 | | x8 | | x<-5 | | x-7 | | x3 | | Ожидается | Получено |
| Д | Н | Д | Н | Д | Н | Д | Н | Д | Н |
| 9 | + |  | + |  |  |  |  |  |  |  | 1 | 1 |
| 4 | + |  |  | + |  |  |  |  |  |  | Нет реш | Нет реш |
| -6 |  | + |  |  | + |  | + |  |  |  | 6 | 6 |
| -7 |  | + |  |  | + |  |  | + |  |  | Нет реш | Нет реш |
| 4 |  | + |  |  |  | + |  |  | + |  | 1 | 1 |
| 3 |  | + |  |  |  | + |  |  |  | + | Нет реш | Нет реш |

*Текст программы на языке Си.*

#include<stdio.h>

#include <math.h>

int main(void)

{

float x,y;

printf("vvedite x ");

scanf("%f",&x);

if (x>5)

if (x>=8)

{

y=sqrt(x-8);

printf("%7.2f",y);

}

else printf("Net resheniya");

else

if (x<-5)

if (x!=-7)

{

y=1/(7+x)+5;

printf("%7.2f",y);

}

else printf("Net resheniya");

else

if (x!=3)

{

y=1/(x-3);

printf("%7.2f",y);

}

else printf("Net resheniya");

}

*Тестирование программы на языке Си.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | x>5 | | x8 | | x<-5 | | x-7 | | x3 | | Ожидается | Получено |
| Д | Н | Д | Н | Д | Н | Д | Н | Д | Н |
| 9 | + |  | + |  |  |  |  |  |  |  | 1 | 1.00 |
| 4 | + |  |  | + |  |  |  |  |  |  | Нет реш | Net resheniya |
| -6 |  | + |  |  | + |  | + |  |  |  | 6 | 6.00 |
| -7 |  | + |  |  | + |  |  | + |  |  | Нет реш | Net resheniya |
| 4 |  | + |  |  |  | + |  |  | + |  | 1 | 1.00 |
| 3 |  | + |  |  |  | + |  |  |  | + | Нет реш | Net resheniya |

***Задание 2.***

Оформляется аналогично заданию 1.

* **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Перечислите операторы языка Паскаль используемые для записи ветвлений.
2. Перечислите операторы языка Си используемые для записи ветвлений.
3. Опишите правила составления и проведения тестирования для задач с ветвлениями.

**ПРАКТИЧЕСКАЯ РАБОТА №4**

**Составление и отладка программ с циклами.**

* **ЦЕЛЬ РАБОТЫ**

*Формирование навыков работы в средах PascalABC.NET и TurboC.*

*Формирование навыков составления, отладки и тестирования программ с циклами на языках Паскаль и Си.*

* **ОБЕСПЕЧЕННОСТЬ ЗАНЯТИЯ**

*- Среда программирования PascalABC.NET*

*- Среда программирования TurboC*

* **ЛИТЕРАТУРА**
* СОДЕРЖАНИЕ ЗАДАНИЙ

1. Составление (с использование циклов всех видов) программы нахождения суммы десяти первых элементов ряда
2. Составление (с циклом с предусловием) программы нахождения суммы ряда с заданной точностью и определения количества элементов ряда вошедших в найденную сумму
3. Составление (с циклом с постусловием) программы обработки числовой последовательности.

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЙ

*Задания выполняются в любом порядке*

* **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Оператор цикла в языке программирования Паскаль.

While условие do

Begin

{тело цикла}

End;

Оператор цикла с постусловием в языке программирования Паскаль.

Repeat

{тело цикла}

Until условие;

Оператор цикла с параметром в языке программирования Паскаль.

For i:=начальное значение to(c шагом 1) или dounto(с шагом -1) do

Begin

{тело цикла}

End;

Оператор цикла с предусловием в языке программирования Си.

while (условие)

{

/\*тело цикла\*/

}

Оператор цикла с постусловием в языке программирования Си.

ЦИКЛ ВЫПОЛНЯЕТСЯ ПРИ ИСТИННОМ УСЛОВИИ.

do

{

/\*тело цикла\*/

}

while (условие);

Оператор цикла с параметром в языке программирования Си.

for (выражения 1 типа; выражение 2 типа; выражения 3 типа)

{

/\*тело цикла\*/

}

где выражения 1 типа – задают начальные значения параметров;

выражение 2 типа – задает условие входа в цикл;

выражения 3 типа – задают законы изменения параметров.

* **ЗАДАНИЯ И ИНСТРУКЦИИ ПО ВЫПОЛНЕНИЮ**

***Задание 1***

Составьте программу для нахождения суммы десяти первых элементов ряда.

Задачу решите три раза: используя цикл с предусловием, цикл с постусловием и цикл с параметром.

***Задача:***

***(по заданию2 ПР3,4,5 УД ОП.08 Теория алгоритмов)***

**1**  **2**  **3** 

**4**   **5**  **6** 

**7**  **8**  **9** 

**10**  **11**  **12**. 

**13**  **14**  **15** 

**16**  **17**  **18** 

**19**  **20**  **21** 

**22**  **23**  **24** 

**25**  **26**  **27** 

**28**  **29**  **30** 

***Задание 2***

Составьте программу для нахождения суммы ряда с точностью 10-3 и определения количества элементов ряда вошедших в найденную сумму. Задачу решите используя цикл с предусловием.

***Задача.***

***(по заданию 1 ПР3 УД ОП.08 Теория алгоритмов)***

**1**  **2**  **3** 

**4**  **5**  **6** 

**7**  **8**  **9** 

**10**  **11** **12** 

**13**  **14**  **15** 

**16**  **17**  **18** 

**19**  **20**  **21** 

**22**  **23**  **24** 

**25**  **26**  **27** 

**28**  **29**  **30** 

***Задание 3***

Составьте программу для решения задачи. В решении задачи используйте цикл с постусловием.

***Задача.***

***(По заданию 1 ПР4 УД ОП08.Теория алгоритмов)***

Введена последовательность целых чисел. Признаком конца ввода является введение числа 0. Вычислите сумму элементов этой последовательности, найдите минимальный элемент в этой последовательности и количество элементов последовательности, удовлетворяющих правилу вашего варианта.

***Варианты:***

1. Число начинается с цифры 2.
2. В числе чередуются четные и нечетные цифры.
3. Количество цифр 2 в числе равно 1.
4. Число состоит из нечетного числа цифр.
5. Сумма цифр числа делиться на пять.
6. Количество цифр 3 в числе не менее 2.
7. Суммы четных и нечетных цифр числа равны.
8. В числе имеется хотя бы одна цифра шесть.
9. Произведение первой и последней цифр числа меньше 10.
10. Первая и последняя цифры числа четные.
11. Число состоит из четного количества цифр.
12. В числе более двух нулей.
13. Первая и последняя цифры в числе равны.
14. Четных цифр в числе большее, чем нечетных.
15. Число состоит только из нечетных цифр.
16. Количество цифр 1 в числе равно 2.
17. Сумма четных цифр числа меньше суммы нечетных цифр числа.
18. В числе нет ни одной цифры 4.
19. Число симметрично.
20. Первая цифра числа делиться на последнюю без остатка.
21. Число состоит только из цифр 3 и 4.
22. Число не содержит цифр 5 и 8.
23. Последняя цифра делиться на первую без остатка.
24. Первая и последняя цифры четные.
25. Хотя бы две цифры в числе нечетные.
26. Количество нечетных цифр числа больше количества четных.
27. Число заканчивается на 3.
28. Число начинается с цифры 9.
29. Все цифры числа упорядочены по возрастанию.
30. Число простое.

* **МЕТОДИКА АНАЛИЗА РЕЗУЛЬТАТОВ, ОБРАЗЕЦ ОТЧЕТА**

***Задание 1***

Составьте программу для нахождения суммы десяти первых элементов ряда.

Задачу решите три раза: используя цикл с предусловием, цикл с постусловием и цикл с параметром.

***Вариант 0***

Вычислите сумму первых десяти элементов ряда an=.

*Текст программы в языке Паскаль.*

*Цикл с предусловием.*

**program** pr4\_1\_1;

**var** a,f,s:real;

i,n:integer;

**BEGIN**

n:=1;

s:=0;

**while** n<=10 **do**

**begin**

i:=1;

f:=1;

**while** i<=n **do**

**begin**

f:=f\*i;

i:=i+1;

**end**;

a:=1/f;

s:=s+a;

n:=n+1;

**end**;

writeln('s=',s:7:2);

**END**.

*Результат исполнения программы.*

s= 1.72

*Текст программы в языке Паскаль.*

*Цикл с постусловием.*

**program** pr4\_1\_2;

**var** a,f,s:real;

i,n:integer;

**BEGIN**

n:=1;

s:=0;

**repeat**

i:=1;

f:=1;

**repeat**

f:=f\*i;

i:=i+1;

**until** i>n;

a:=1/f;

s:=s+a;

n:=n+1;

**until** n>10;

writeln('s=',s:7:2);

**END**.

*Результат исполнения программы.*

s= 1.72

*Текст программы в языке Паскаль.*

*Цикл с параметром.*

**program** pr4\_1\_3;

**var** a,f,s:real;

i,n:integer;

**BEGIN**

s:=0;

**for** n:=1 **to** 10 **do**

**begin**

f:=1;

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

**begin**

f:=f\*i;

**end**;

a:=1/f;

s:=s+a;

**end**;

writeln('s=',s:7:2);

**END**.

*Результат исполнения программы.*

s= 1.72

*Текст программы в языке Си.*

*Цикл с предусловием.*

include <stdio.h>

int main (void)

{

int n,i;

double a,s,f;

n=1;

s=0;

while (n<=10)

{

f=1;

i=1;

while (i<=n)

{

f=f\*i;

i++;

}

a=(double)1/f;

s+=a;

n++;

}

printf(“s=%7.2lf\n”,s);}

*Результат исполнения программы.*

S= 1.72

*Текст программы в языке Си.*

*Цикл с постусловием.*

#include <stdio.h>

int main (void)

{

int n,I;

double a,s,f;

n=1;

s=0;

do

{

f=1;

i=1;

do

{

f=f\*I;

i++;

}

while(i<=n);

a=(double)1/f;

s+=a;

n++;

}

while (n<=10);

printf(“s=%7.2lf\n”,s);

}

*Результат исполнения программы.*

S= 1.72

*Текст программы в языке Си.*

*Цикл с параметром.*

#include <stdio.h>

int main (void)

{

int n,I;

double a,s,f;

s=0;

for(n=1; n<=10; n++)

{

f=1;

for (i=1;i<=n;i++)

{

f=f\*I;

}

a=(double)1/f;

s+=a;

}

printf(“s=%7.2lf\n”,s);

}

*Результат исполнения программы.*

S= 1.72

***Задания 2 и 3 оформляются аналогично.***

* **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Перечислите операторы циклов с предусловием в языках Паскаль и Си.
2. Перечислите операторы циклов с постусловием в языках Паскаль и Си.
3. Перечислите операторы циклов с параметром в языках Паскаль и Си.

**ПРАКТИЧЕСКАЯ РАБОТА №5**

**Составление и отладка программ с массивами.**

* **ЦЕЛЬ РАБОТЫ**

*Формирование навыков работы в средах PascalABC.NET и TurboC.*

*Формирование навыков составления, отладки и тестирования программ с массивами на языках Паскаль и Си.*

* **ОБЕСПЕЧЕННОСТЬ ЗАНЯТИЯ**

*- Среда программирования PascalABC.NET*

*- Среда программирования TurboC*

* **ЛИТЕРАТУРА**
* СОДЕРЖАНИЕ ЗАДАНИЙ

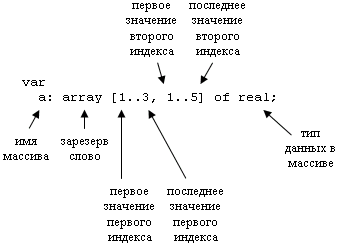
1. Составление программы использующей структуру данных массив.

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЙ

*Задания выполняются в любом порядке*

* **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Массив – это упорядоченная совокупность однотипных данных, порядок в которой задается номерами элементов.

Пример объявления массива в языке Паскаль.   


Пример объявления массива в языке Си.

float A [3][5];

количество строк и количество столбцов

имя массива

тип данных в массиве

* **ЗАДАНИЯ И ИНСТРУКЦИИ ПО ВЫПОЛНЕНИЮ**

***Задание 1***

Составьте программу для решения задачи.

***Задача.***

1. Дано число k (0 < k < 11) и матрица размера 4 x 10. Найти сумму и произведение элементов k - го столбца данной матрицы.
2. Дана матрица размера 5 x 9. Найти суммы элементов всех ее четных строк и столбцов.
3. Дана матрица размера 5 x 10. Найти минимальное и максимальное значение в каждой строке.
4. Дана матрица размера 5 x 10. В каждой строке найти количество элементов, больших среднего арифметического всех элементов этой строки.
5. Дана матрица размера 5 x 10. Преобразовать матрицу, поменяв местами минимальный и максимальный элемент в каждой строке.
6. Дана матрица размера 5 x 10. Найти минимальное значение среди сумм элементов всех ее строк и столбцов и номер строки столбца с этим минимальным значением.
7. Дана матрица размера 5 x 10. Найти минимальный и максимальный среди максимальных и минимальных элементов каждой строки и столбца.
8. Дана целочисленная матрица размера 5 x 10. Вывести номер ее первой и последней строки, содержащей равное количество положительных и отрицательных элементов (нулевые элементы не учитываются). Если таких строк нет, то вывести 0.
9. Дана матрица размера 5 x 10. Вывести номер ее первой и последней строки, содержащего только положительные элементы. Если таких строк нет, то вывести 0.
10. Дана целочисленная матрица размера M x N. Различные строки (столбцы) матрицы назовем похожими, если совпадают множества чисел, встречающихся в этих строках (столбцах). Найти количество строк, похожих на первую строку.
11. Дана целочисленная матрица размера M x N. Найти количество ее строк и столбцов, все элементы которых различны.
12. Дана целочисленная матрица размера M x N. Вывести номер ее первой строки, содержащей максимальное количество одинаковых элементов.
13. Дана квадратная матрица порядка M. Найти сумму элементов ее главной и побочной диагонали.
14. Дана квадратная матрица порядка M. Найти суммы элементов ее диагоналей, параллельных главной и побочной (начиная с одноэлементной диагонали A [1, M] и A [1, 1] ).
15. Дана квадратная матрица порядка M. Вывести минимальные из элементов каждой ее диагонали, параллельной главной (начиная с одноэлементной диагонали A [1, M]).
16. Дана квадратная матрица порядка M. Заменить нулями элементы матрицы, лежащие ниже главной и выше побочной диагонали.
17. Дана квадратная матрица порядка M. Заменить нулями элементы, лежащие одновременно выше главной диагонали (включая эту диагональ) и выше побочной диагонали (также включая эту диагональ).
18. Дана квадратная матрица порядка M. Зеркально отразить ее элементы относительно [горизонтальной оси симметрии] матрицы.
19. Дана квадратная матрица порядка M. Повернуть ее на 90 градусов в положительном направлении.
20. Дана матрица размера 5 x 10. Вывести количество строк, элементы которых монотонно возрастают.
21. Дана матрица размера 5 x 10. Найти минимальный среди элементов тех строк , которые упорядочены либо по возрастанию, либо по убыванию. Если такие строки отсутствуют, то вывести 0.
22. Даны два числа k1 и k2 и матрица размера 4 x 10. Поменять местами строки матрицы с номерами k1 и k2.
23. Дана матрица размера 5 x 10. Поменять местами строки, содержащие минимальный и максимальный элементы матрицы.
24. Дана матрица размера 5 x 10. Поменять местами столбец с номером 1 и первый из столбцов, содержащих только положительные элементы.
25. Дано число k и матрица размера 4 x 10. Удалить строку матрицы с номером k.
26. Дана матрица размера 5 x 10. Удалить столбец, содержащий минимальный элемент матрицы.
27. Дана матрица размера 5 x 10. Удалить первый столбец, содержащие только положительные элементы.
28. Дано число k и матрица размера 4 x 9. Перед строкой матрицы с номером k вставить строку из нулей.
29. Дана матрица размера 4 x 9. Продублировать строку матрицы, содержащий ее минимальный элемент.
30. Дана матрица размера 5 x 9. Перед последним столбцом, содержащим только положительные элементы, добавить столбец, состоящий из единиц.

* **МЕТОДИКА АНАЛИЗА РЕЗУЛЬТАТОВ, ОБРАЗЕЦ ОТЧЕТА**

Вариант 0.

Организовать заполнение двумерного массива размера 5\*10 случайными целыми числами. Организовать вывод массива на экран в виде матрицы.

*Алгоритм решения задачи на языке блок-схем.*



*Текст программы на языке Паскаль.*

**program** pr5\_1;

**var** A:**array**[1..5,1..10] **of** integer;

i,j:integer;

**BEGIN**

randomize;

**for** i:=1 **to** 5 **do**

**for** j:=1 **to** 10 **do**

A[i,j]:=random(100)-50;

**for** i:=1 **to** 5 **do**

**begin**

**for** j:=1 **to** 10 **do**

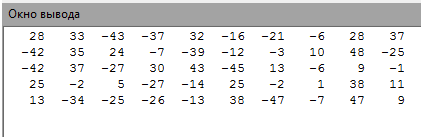
write(A[i,j]:5);

writeln;

**end**;

**END**.

*Результат работы программы на языке Паскаль.*



*Текст программы на языке Си.*

#include <stdio.h>

#include <stdlib.h>

int main(void)

{

int A[5][10];

int i,j;

randomize();

for (i=0; i<5; i++)

for (j=0; j<10; j++)

A[i][j]=random(100)-50;

for (i=0; i<5; i++)

{

for (j=0; j<10; j++)

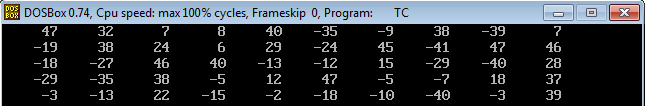
printf("%7i",A[i][j]);

printf("\n");

}

}

*Результат работы программы на языке Си.*



* **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Дайте определение структуры массив.
2. Перечислите свойства массива.
3. Назовите правила объявления массивов в языке Си.

**ПРАКТИЧЕСКАЯ РАБОТА №6**

**Составление и отладка программ со строками.**

* **ЦЕЛЬ РАБОТЫ**

*Формирование навыков работы в средах PascalABC.NET и TurboC.*

*Формирование навыков составления, отладки и тестирования программ со строками на языках Паскаль и Си.*

* **ОБЕСПЕЧЕННОСТЬ ЗАНЯТИЯ**
* *Среда программирования PascalABC.NET*
* *Среда программирования TurboC*
* **ЛИТЕРАТУРА**
* СОДЕРЖАНИЕ ЗАДАНИЙ

1. Составление программы использующей структуру данных строка.

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЙ

*Задания выполняются в любом порядке*

* **КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

Переменная типа строка предназначена для обработки цепочек символов. Каждый символ является элементом типа char.

Пример объявления строк в языке Паскаль:

var

S1: string; {объявлена строка с именем S1 максимально возможной длины 255 символов}

S2: string [255] {объявлена строка с именем S2 максимально возможной длины 255 символов}

S3: string [10] {объявлена строка с именем S3 максимально возможной длины 10 символов}

Пример объявления строк в языке Си:

char S1[50]; /\*объявлена строка с именем S1 и максимальной длины 49 символов\*/

В языке Паскаль для обработки строки используются:

1. Операции:
   1. Конкатенации (сцепления строк) +
2. Функции:
   1. Копирования подстроки copy(str, n1, n2)
   2. Поиска подстроки pos(str1,str)
   3. Преобразования символа в заглавный upcase(ch)