

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан ФЭСУ Бурковский А.В.
«31» августа 2021 г.



РАБОЧАЯ ПРОГРАММА
дисциплины
«Прикладная информатика»

Направление подготовки 13.03.02 Электроэнергетика и электротехника

Профиль Электромеханика

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2018

Автор программы

И.В. Зубарев /Зубарев И.В./

Заведующий кафедрой
Электропривода,
автоматики и управления в
технических системах

В.Л. Бурковский /Бурковский В.Л./

Руководитель ОПОП

А.В. Тикунов /Тикунов А.В./

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

приобретение знаний, умений и навыков по основам прикладной информатики, алгоритмизации, программировании на языках высокого уровня, выработка умений при работе с системами программирования и написание программ приложений, применительно для электроэнергетики, электротехники и современных информационных технологий.

1.2. Задачи освоения дисциплины

- усвоение понятия алгоритма, его свойств, способов описания и формирование представлений об основных алгоритмических конструкциях, выработка умений применять их для построения алгоритмов решения учебных задач;
- формирование знаний об одном из языков высокого уровня и умения использовать его для записи алгоритмов решения простых задач;
- выработать умения разработки алгоритмов линейной, разветвляющейся и циклической структуры.
- выработать умения построения блок-схем алгоритмов и их оформления с помощью Microsoft Vizio.
- выработать умения использовать инструменты создания и редактирования графических объектов, построения графических изображений;
- формирование знаний об определениях, понятиях и основных алгоритмах создания Web-документов в формате HTML
- выработать умения владеть компьютерными методами создания и редактирования Web-документов, применять полученные знания в сфере его профессиональной деятельности.
- формирование и развитие у студентов научного мышления; умения активно участвовать в творческой дискуссии, делать правильные выводы;
- формирование и развитие у студентов профессиональных качеств, необходимых обучающимся, как будущим специалистам: дисциплинированность, исполнительность; трудолюбие, добросовестное отношение к учебе, стремление в совершенстве овладеть избранной специальностью.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Прикладная информатика» относится к дисциплинам обязательной части блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Прикладная информатика» направлен

на формирование следующих компетенций:

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-2	<p data-bbox="708 329 810 360">Знать:</p> <ul data-bbox="740 376 1481 1021" style="list-style-type: none"><li data-bbox="740 376 1481 501">– об основах алгоритмизации инженерных задач, основных видах алгоритмов и их свойствах<li data-bbox="740 510 1481 591">– об основах программирования на алгоритмическом языке высокого уровня.<li data-bbox="740 600 1481 770">– об назначении и возможностях «Microsoft Visio», порядке создания, форматирования, и редактирования графических объектов в Microsoft Visio;<li data-bbox="740 779 1481 904">– об порядке создания простейших графических карт и схем в системах деловой графики.<li data-bbox="740 913 1481 1021">– об определениях, понятиях и основных алгоритмах создания Web-документов в формате HTML. <p data-bbox="692 1032 804 1064">Уметь:</p> <ul data-bbox="740 1079 1481 1424" style="list-style-type: none"><li data-bbox="740 1079 1481 1205">– разрабатывать алгоритмы линейной разветвляющейся, циклической структуры,<li data-bbox="740 1214 1481 1294">– разработки алгоритмов с помощью счетного оператора цикла.<li data-bbox="740 1303 1481 1335">– построить графические изображения.<li data-bbox="740 1344 1481 1424">– применять знания о создании Web-документов на практике. <p data-bbox="692 1473 836 1505">Владеть:</p> <ul data-bbox="740 1520 1481 1989" style="list-style-type: none"><li data-bbox="740 1520 1481 1691">– методами разработки основных видов алгоритмов с помощью блок-схем на персональном компьютере в системе Microsoft Visio.<li data-bbox="740 1700 1481 1825">– алгоритмами решаемых прикладных задач и осуществлять их реализацию на персональном компьютере.<li data-bbox="740 1834 1481 1989">– компьютерными методами создания и редактирования Web-документов, применять полученные знания в профессиональной сфере.

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Прикладная информатика» составляет 1 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		8
Аудиторные занятия (всего)	12	12
В том числе:		
Практические занятия (ПЗ)	12	12
Самостоятельная работа	24	24
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость: академические часы	36	36
зач.ед.	1	1

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Прак зан.	СРС	Всего, час
1	ОСНОВЫ АЛГОРИТМИЗАЦИИ	1. Основные типы алгоритмических структур, 2. Разработка блок-схем алгоритмов, 3. Построение блок-схем алгоритмов в Microsoft Visio.	2	4	6
2	СИСТЕМА РАЗРАБОТКИ ДЕЛОВОЙ ГРАФИКИ	1. Основы работы в Microsoft Visio. 2. Разработка схем визуального моделирования. 3. Конструирование планов и схем.	2	4	6
3	РАЗРАБОТКА ПРОГРАММ ЛИНЕЙНОЙ И РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ	1. Разработка алгоритмов линейной структуры, 2. Разработка алгоритмов разветвляющейся структуры	2	4	6
4	РАЗРАБОТКА АЛГОРИТМОВ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ	1. Реализация алгоритмов с помощью счетного оператора цикла	2	4	6
5	ГРАФИЧЕСКИЕ СРЕДСТВА АЛГОРИТМИЧЕСКОГО ЯЗЫКА	1. Построение графических изображений	2	4	6
6	СОЗДАНИЕ WEB-ДОКУМЕНТА	1. Создание и сохранение Web-документов в формате HTML. 2. Связь между несколькими HTML-документами с помощью гипертекстовых ссылок.	2	4	6
Итого			12	24	36

5.2 Перечень лабораторных работ

Не предусмотрено учебным планом

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-2	Знать: об основах алгоритмизации инженерных задач, основных видах алгоритмов и их свойствах об основах программирования на алгоритмическом языке высокого уровня. об назначении и возможностях «Microsoft Visio», порядке создания, форматирования, и редактирования графических объектов в Microsoft Visio; об порядке создания простейших графических карт и схем в системах деловой графики. об определениях, понятиях и основных алгоритмах создания Web-документов в формате HTML	Решение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь: разрабатывать алгоритмы линейной разветвляющейся, циклической структуры, разработки алгоритмов с помощью счетного оператора цикла. построить графические изображения. применять знания о создании Web-документов на практике.	Решение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть: методами разработки основных видов алгоритмов с помощью блок-схем на персональном компьютере в системе Microsoft Visio. алгоритмами решаемых прикладных задач и осуществлять их реализацию на персональном	Решение прикладных задач в конкретной предметной области	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	компьютере. компьютерными методами создания и редактирования Web-документов, применять полученные знания в профессиональной сфере.			
--	---	--	--	--

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 8 семестре для очной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ОПК-2	Знать: об основах алгоритмизации инженерных задач, основных видах алгоритмов и их свойствах об основах программирования на алгоритмическом языке высокого уровня. об назначении и возможностях «Microsoft Visio», порядке создания, форматирования, и редактирования графических объектов в Microsoft Visio; об порядке создания простейших графических карт и схем в системах деловой графики. об определениях, понятиях и основных алгоритмах создания Web-документов в формате HTML	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь: разрабатывать алгоритмы линейной разветвляющейся, циклической структуры, разработки алгоритмов с помощью счетного оператора цикла. построить графические изображения. применять знания о создании Web-документов на практике.	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть: методами разработки основных видов алгоритмов с помощью блок-схем на персональном компьютере в системе Microsoft Visio. алгоритмами решаемых прикладных задач и осуществлять их реализацию на персональном компьютере. компьютерными методами создания и редактирования Web-документов, применять полученные знания в профессиональной сфере.	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

ЗАДАНИЕ 1 (по вариантам). Составить блок-схему вычислительного процесса и реализовать ее в Microsoft Vizio.

1. $Z = \max(A, \min(B, C, D))$
2. $Z = \min(A + 1, \max(B, C + D))$
3. $Y = \min(A, \max(B + C, D + 1))$

ЗАДАНИЕ 2 (по вариантам). Составить блок-схему вычислительного процесса и реализовать ее в Microsoft Vizio.

$$Z = \begin{cases} 0.25 \min(A, B) + |C|, & \text{если } A + B + C \geq 0 \\ 0.1(1 - A^2) + \frac{B}{C + 1}, & \text{если } A + B + C < 0, A \geq 0 \\ \min(A, B, C), & \text{если } A + B + C < 0, A < 0 \end{cases}$$

ЗАДАНИЕ 3 (по вариантам). Составить блок-схему вычислительного процесса и реализовать ее в Microsoft Vizio.

1. Дан массив $X_i, i = 1, \dots, 15$. Найти среднее арифметическое значение положительных и отрицательных элементов массива R1 и R2. Определить, какое из найденных значений больше по абсолютной величине.
2. Дан массив $A_i, i = 1, 2, \dots, 15$. Найти сумму положительных элементов и сумму отрицательных элементов массива S1 и S2. Определить, что больше по абсолютной величине – S1 или S2.
3. Дан массив $X_i, i = 1, 2, \dots, 15$. Найти произведения положительных и отрицательных элементов массива P1 и P2. Определить, что больше по абсолютной величине – P1 или P2.

ЗАДАНИЕ 4. Средствами программы Microsoft Visio создайте следующие схемы:

- Построить организационную структуру управления факультетом.
- Построить организационную структуру управления цеха.
- Построить организационную структуру управления учебной группой.
- Построить схему локальной вычислительной сети.

ЗАДАНИЕ 5 При заданных значениях x и параметра a вычислить значение функции F :

$$1.15. F = \begin{cases} (ax)^2 - \sqrt{\frac{x}{2}} + \sin(x + 3), & \text{при } x < 10 \\ -\cos 4x + \operatorname{arctg} \frac{\pi x}{5} - \ln(6 + x), & \text{при } x = 10 \\ e^{7x} - \operatorname{ctg} \frac{x}{8} + |a + x|, & \text{при } x > 10 \end{cases}$$

ЗАДАНИЕ 6

- Составить программу, которая вводит время (целое число от 0 до 24) и определяет, какому времени суток это время соответствует (полночь, ночь, утро, полдень, день, вечер). Если число лежит вне диапазона 0-24, выводится надпись «Неверное число».
- Составить программу, которая вводит номер месяца (число от 1 до 12) и определяет, какому времени года этот месяц соответствует. Если число лежит вне диапазона 1-12, выводится надпись «Неверное число».
- Составить программу, которая вводит номер месяца (число от 1 до 12) и определяет, какому кварталу этот месяц соответствует.

ЗАДАНИЕ 7

- Написать программу, которая вычисляет факториал числа, введенного в клавиатуры.
- Написать программу, которая вычисляет сумму ряда $1, 2, 3, \dots, n$ (n вводится с клавиатуры).
- Написать программу, которая вычисляет произведение ряда $1, 1/2^2, 1/3^2, \dots, 1/n^2$ (n вводится с клавиатуры).
- Написать программу, которая вычисляет сумму четных членов ряда $1, 2, 3, \dots, n$ (n вводится с клавиатуры).
- Написать программу, которая вычисляет количество нечетных членов ряда $1, 2, 3, \dots, n$ (n вводится с клавиатуры).

ЗАДАНИЕ 8. Изобразить рисунок

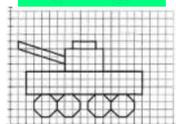
1.



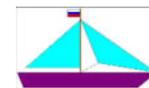
2



3.



4.



ЗАДАНИЕ 9 Построить график функции $y(x)$.

1. $y = \ln(1+x^2)$;

2. $y = \sin(x^2) + 4x$.

3. $y = \frac{x^3}{3} + x^2$;

4. $y = \cos(0.1+x) - 2x$;

5. $y = x + 2 \operatorname{ctg} x$;

6. $y = (1+x^2) \cdot e^x$;

ЗАДАНИЕ 10 Создание Web-сайта на основе программных приложений MICROSOFT Office (Word и Excel)

- Мои увлечения
- Мои друзья

7.2.2 Примерный перечень заданий для решения стандартных задач

Задача 1. Составить алгоритм вычисления площади треугольника со

сторонами A, B, C по формуле Герона: $S = \sqrt{p(p-A)(p-b)(p-C)}$, где $p = (A+B+C)/2$.

Решение. Словесное описание алгоритма будет иметь вид:

1. Ввести A, B, C .
2. Вычислить $p = (A+B+C)/2$.
3. Вычислить $S = \sqrt{p(p-A)(p-b)(p-C)}$.
4. Вывести S .
5. Конец.

Алгоритм имеет линейную структуру при любых исходных данных. И каждое последующее действие следует из предыдущего. Схема алгоритма представлена на рис. 1.6.

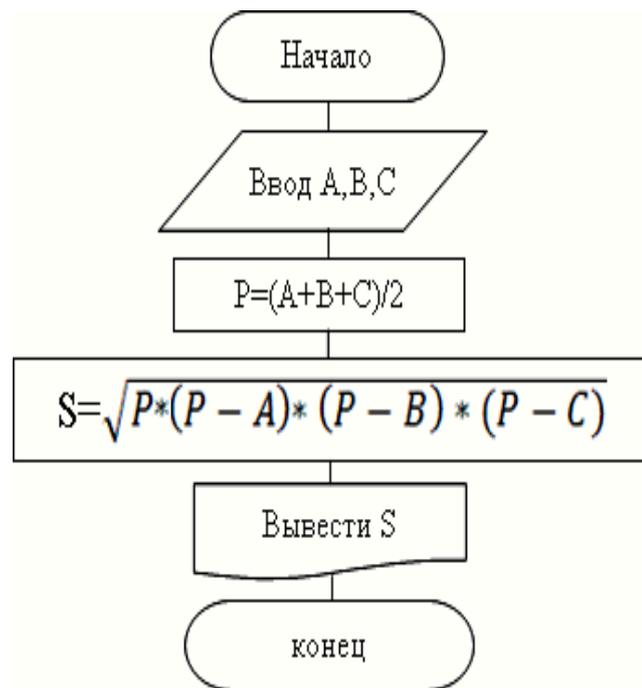


Рисунок 1.6 – Схема линейного алгоритма

Задача 2. Вычислить сумму сходящегося ряда:

$$S = \sum_{n=1}^{\infty} \frac{n}{n^2 + A}$$

Условие окончания вычисления $\left| \frac{n}{n^2 + A} \right| \leq EPS$, где EPS – очень малая величина, которая определяет точность решения задачи.

Решение. Схема алгоритма представлена на рис. 1.7.

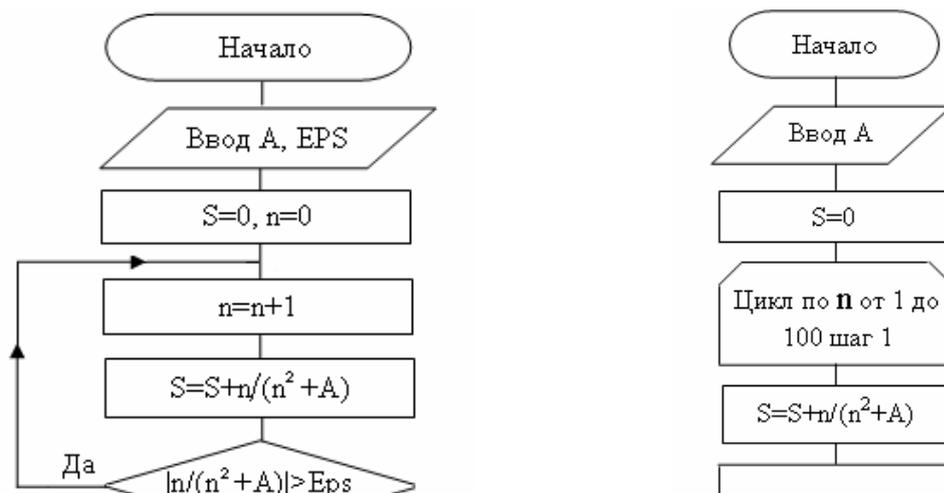


Рисунок 1.7 – Схема циклического алгоритма с параметром

числом повторений

Если же в задаче 2 нужно вычислить сумму 100 элементов ряда, то используют циклический алгоритм параметром. Схема алгоритма представлена на рис. 1.8.

Рисунок 1.8 – Циклический алгоритм с неизвестным

Задача 3. Разработать блок-схему алгоритма Евклида, определяющего наибольший общий делитель (НОД) двух натуральных чисел A и B.

Решение. В основе алгоритма Евклида лежит правило:

$\text{НОД}(A,B) = \text{НОД}(\min(A,B), |A-B|)$, где $\text{НОД}(A,B)$ — наибольший общий делитель двух натуральных чисел A и B.

Основной идеей решения задачи является многократное применение указанного выше правила, после которого большее из чисел очередной пары уменьшается. Решение получено, когда числа оказываются равны друг другу. Поскольку количество повторений заранее неизвестно, в алгоритме следует применить цикл "пока" с предусловием (рис. 1.9).

Задача 4. Даны действительные числа A, B, C. Составить схему алгоритма вычисления $Z = \max(A^2+B, \min(B+1,C))$.

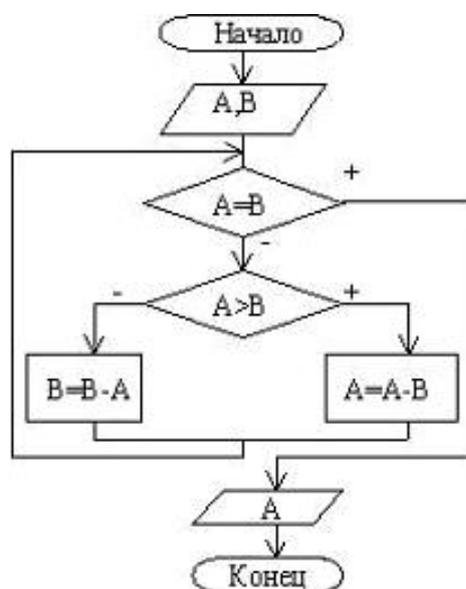


Рисунок 1.9 – Алгоритм Евклида

Решение. В данном примере возможно три варианта ответа: или $A^2 + B$, или $B + 1$, или C . И выбор будет выполнен только по 2 результатам проверки условий. Вычисление величины Z выполняется в два этапа. На первом этапе выбираем \min из двух величин $B+1$ и C . Результат выбора обозначим через промежуточную переменную R , т.е. $R = \min(B+1, C)$. Тогда формула для вычисления Z будет иметь вид $Z = \max(A^2 + B, R)$.

Алгоритм вычисления минимума и максимума из двух величин будет иметь вид:

$$R = \begin{cases} C, & \text{если } B+1 \geq C, \\ B+1, & \text{если } B+1 < C, \end{cases} \quad Z = \begin{cases} A^2 + B, & \text{если } A^2 + B \geq R, \\ R, & \text{если } A^2 + B < R. \end{cases}$$

Схема представлена на рис. 2.1.

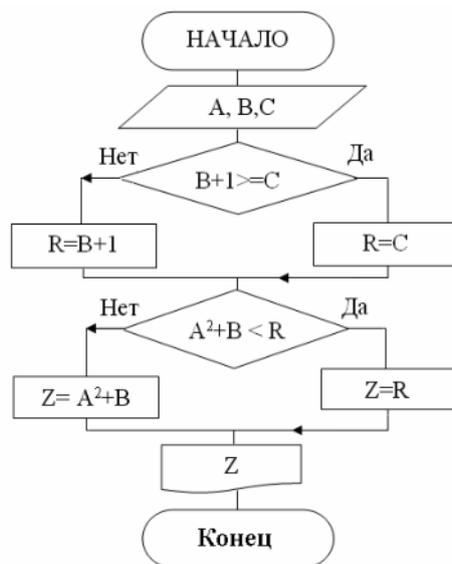


Рис. 2.1 – Схема алгоритма вычисления $Z = \max(A^2 + B, \min(B+1, C))$

Задача 5. Даны действительные числа A, B, C . Составить схему алгоритма вычисления Z .

$$Z = \begin{cases} \max(A, B) + C^3 & , \text{ если } A + B + C \geq 0, \\ 0.5(A - B^2) + e^{0.1C} & , \text{ если } A + B + C < 0, A \geq 0, \\ A^3 + \min(B, C) & , \text{ если } A + B + C < 0, A < 0. \end{cases}$$

Решение. Схема алгоритма данной задачи имеет разветвленную структуру. Переменная Z вычисляется по одной из формул в зависимости от условий. Первым должно проверяться условие $A+B+C \geq 0$. Если условие выполняется, т.е. логическое выражение имеет значение “Да” (истина), то Z вычисляется по первой ветке $Z = \max(A, B) + C^3$. Схема алгоритма представлена на рис. 2.2.

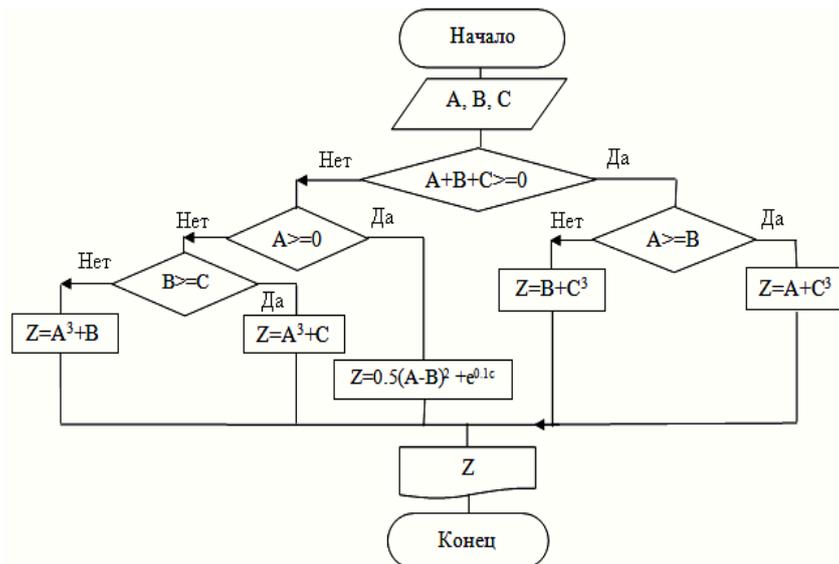


Рисунок 2.2 – Схема алгоритма вычисления функции

Задача 6. Задан массив по имени А, состоящий из 20 элементов A_i , $i = 1, \dots, 20$. Составить схему алгоритма вычисления суммы и произведения элементов этого массива.

Решение. В соответствии со смыслом описываемых величин выбираем имя переменных: для суммы – S, произведения – P. Алгоритм вычисления будет состоять из следующих шагов, представленных на рис. 2.3.

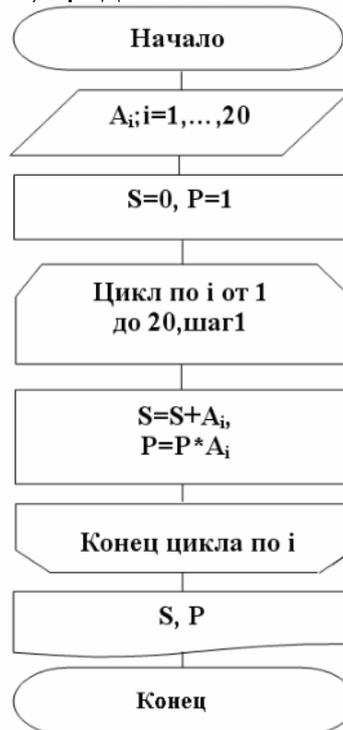


Рисунок 2.3 – Схема алгоритма вычисления суммы и произведения элементов массива

Задача 7. Задан массив X, состоящий из 20 элементов, X_i , $i = 1, \dots, 20$. Составить схему алгоритма вычисления суммы и количества положительных элементов массива.

Решение. В этом примере вычисляется сумма и количество только положительных элементов. Поэтому внутри цикла нужно сделать проверку

очередного элемента на знак. Алгоритм вычисления представлена на рис. 2.4.

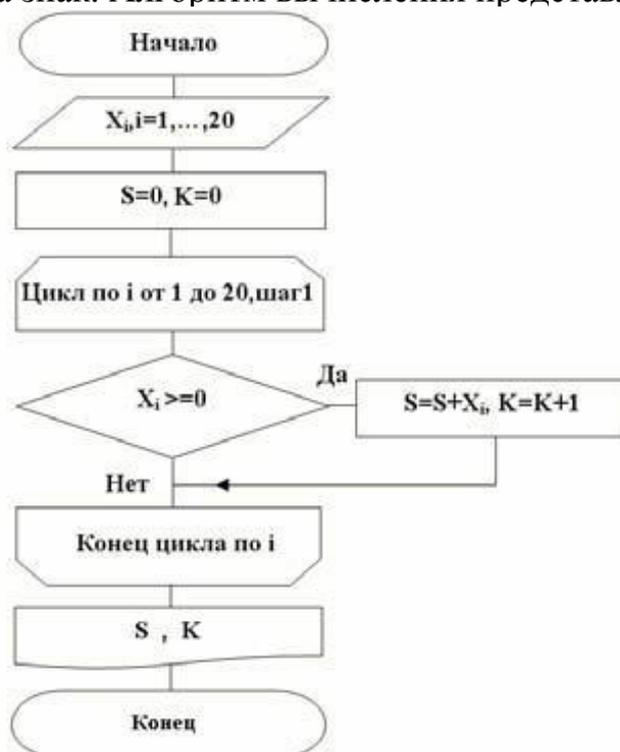


Рисунок 2.4 – Схема алгоритма вычисления суммы и количества положительных элементов массива

Задача 8. Требуется создать схему алгоритма вычисления суммы факториалов.

Решение:

1. Запустить программу Microsoft Visio 2013.
2. В окне «Приступая к работе» выбрать в «Категориях шаблонов», расположенных в левой части, категорию «Блок-схема» (в версии 2003 Flowchart).
3. Выбрать шаблон «Простая блок-схема» (Basic Flowchart Shapes).
4. На панели инструментов выбрать масштаб 75%.
5. Передвинуть границу между окном «Фигуры» (Shapes) и окном «Область вставки» так, чтобы «Область вставки» занимала не менее двух третей экрана.
6. Щелчок по кнопке «Текст» (Text Tool), в верхней части страницы области вставки нарисовать прямоугольник для будущего текста заголовка.
7. На панели инструментов установить шрифт Arial 18 пт жирный.
8. Ввести текст «СУММА ФАКТОРИАЛОВ», для окончания ввода щелкнуть вне прямоугольника текста на пустом месте (рис. 3.1). Сохранить файл в своей рабочей папке.

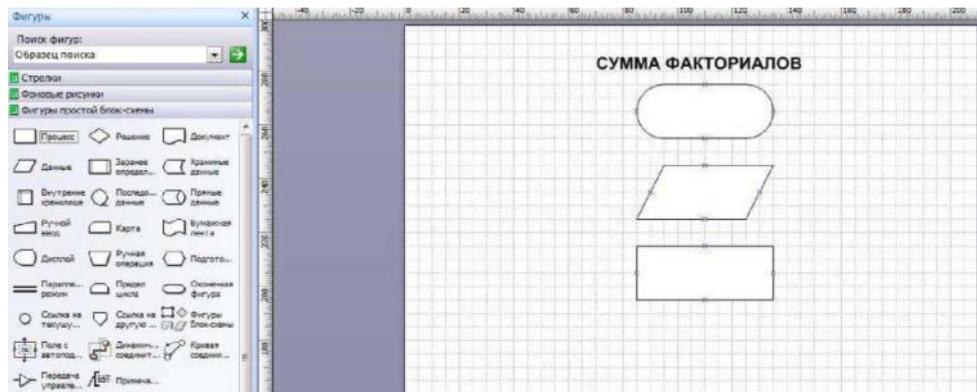


Рисунок 3.1 – Фигуры алгоритма

9. Щелчок по кнопке «Указатель» (Pointer Tool), щелчок по введенном тексту, вокруг текста появится поле выделения с манипуляторами выделения.

10. Передвинуть мышью поле выделения текста в верхней части страницы, расположив его так, как это указано на рис. 3.1. Для снятия выделения - щелчок вне прямоугольника выделения.

11. Перетащить с трафарета элемент «Оконечная фигура (Terminator)».

12. Передвинуть манипулятор в виде зеленого квадрата в нижнем левом углу поля выделения фигуры так, чтобы горизонтальный размер фигуры установился в десять клеток страницы, а вертикальный - в четыре.

13. Передвинуть полученную фигуру примерно под середин заголовка.

14. Аналогично перетащить элементы: «Данные» (Data), «Процесс» (Process), «Подготовка» (Preparation), «Заранее определенный процесс (Predefined Process), «Документ» (Document) и, передвигая манипуляторы поле выделения (зеленые квадраты), установить одинаковые габаритные размеры фигур.

15. Скопировать элементы «Оконечная фигура» и «Процесс» верхней части страницы в нижнюю с помощью перетаскивания при нажатой клавише Ctrl. Результат представлен на рис. 3.2.

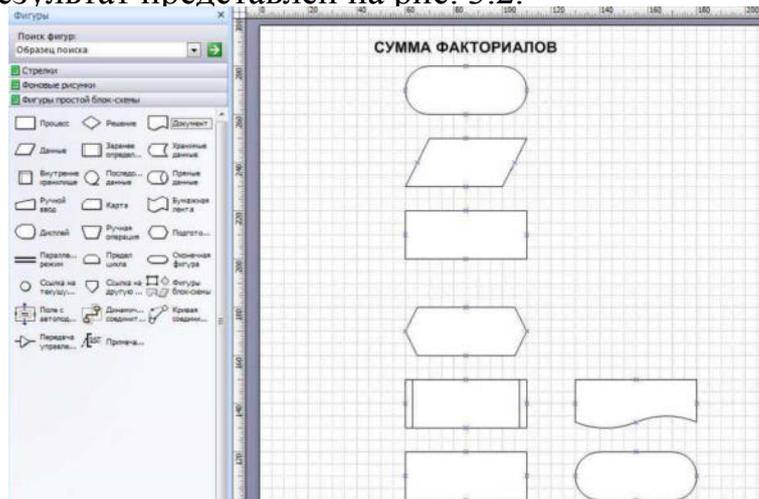


Рисунок 3.2 – Блоки

16. Соединить две верхние фигуры. Для этого выполнить следующие действия:

- Щелчок по кнопке «Соединительная линия» (Connector Tool) на панели инструментов.
- Установить курсор на синий крест, расположенный в центре нижней

края фигуры «Оконечная фигура»; появление красного квадрата в этом месте означает наличие соединения.

- Нажав левую клавишу мыши и удерживая ее, переместить курсор на синий крест, расположенный в центре верхнего края фигуры «Данные»; появление красного квадрата в этом месте означает наличие соединения. Отпустить левую клавишу мыши.

17. Аналогично создать остальные соединения (рис. 3.3). Изгибы установятся автоматически.

18. Перенести соединительную линию, исходящую от последней фигуры «Процесс» и направленную вверх, на левую сторону схемы. Для этого выполнить следующие действия:

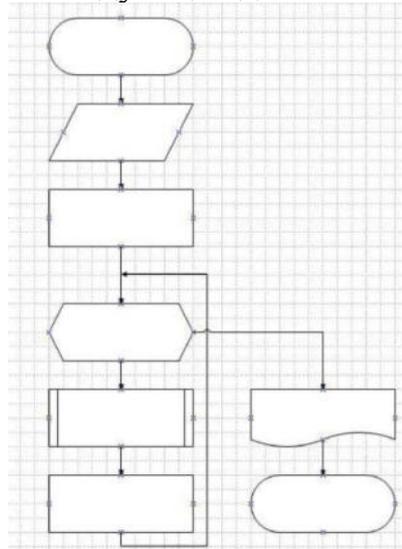


Рисунок 3.3 – Соединения

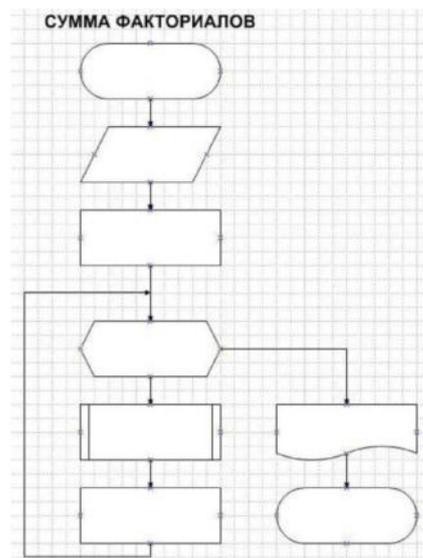


Рисунок 3.4 – Перенос соединения

- Щелкнуть по кнопке «Указатель».
- Щелкнуть по линии - появятся манипуляторы зеленого цвета.
- Установить курсор на манипулятор, расположенный в середине вертикального сегмента линии, при этом курсор преобразуется в двунаправленную стрелку.
- Нажав левую клавишу мыши и удерживая ее, перетащить сегмент влево (рис. 3.4). Отмена выделения - клавиша Esc.

19. Ввести текст в элементы. Для этого выполнить следующие действия:

- Двойной щелчок внутри первой фигуры «Оконечная фигура» (рис. 3.5).
- На панели инструментов установить шрифт Times New Roman 14 пт.
- Ввести текст во все фигуры с шрифтом Times New Roman 14 п (рис. 3.5).

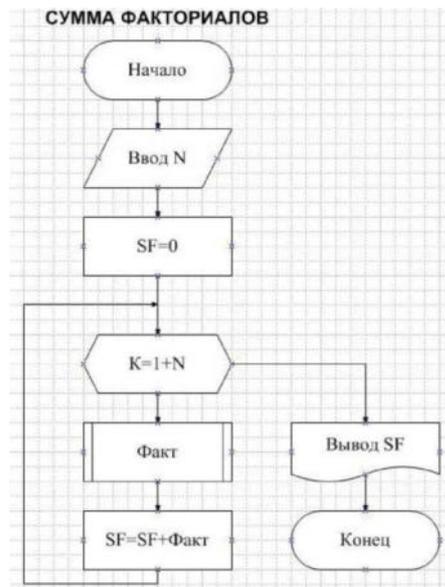


Рисунок 3.5 – Текст в блоках

20. Изменить высоту шрифта во всех фигурах с помощью копирования формата. Для этого выполнить следующие действия:

- Изменить высоту шрифта в тексте «Начало» первой фигуры до 1 пт.
- Двойной щелчок по тексту «Начало» в первой фигуре - появится синее выделение текста.
- Двойной щелчок по кнопке «Формат по образцу» (Format Painter) на панели инструментов.
- Последовательно щелкая по текстам внутри элементов, перенесет формат текста первой фигуры на все остальные.
- Отжать кнопку «Формат по образцу».
- Снять выделение - кнопка Esc.

21. Изменить размер стрелок. Для этого выполнить следующие действия:

- На панели инструментов в списке кнопки «Указатель» выбрать «Выбор нескольких объектов» (Multiple Select).
- Для выделения всех соединительных линий, удерживая нажато клавишу Ctrl, последовательно щелкать по всем соединительным линиям.
- Выполнить команду «Формат / Линия» (Format / Line).
- В окне «Линия» (Line) в строке «Конец» (End) выбрать стрелку 1; а в строке «Начальный размер» (Begin size) - «Очень крупный» (Extra Large «Применить» (Apply), ОК.
- Снять выделение клавишей Esc.

22. Ввести текст для соединительных линий фигуры с текстом «K=1+N»
Порядок действий:

- Двойной щелчок по соединительной линии между элементом с текстом «K=1+N» и элементом с текстом «Вывод SF».
- В прямоугольнике с зеленой рамкой ввести текст «Нет».
- Щелчок вне прямоугольника на пустом месте.
- Щелчок по слову «Нет». Внутри текста появится манипулятор в виде желтого ромба.
- Перетащить этот манипулятор так, чтобы слово «Нет» расположилось над линией (рис. 3.6). Снять выделение клавишей Esc.
- Выделить текст в любом блоке (фигуре). Появится синий прямоугольник.

- Щелчок по кнопке «Формат по образцу».
- Щелчок по введенному слову «Нет». Снять выделение клавише Esc.

- Аналогично ввести слово «Да» (рис. 3.6).

23. Выполнить контекстную команду «Переименовать» (Rename Page) выполненную на ярлыке «Страница 1» (Page 1). Ввести название страниц «Сумма факториалов» (рис. 3.6).

24. Создать блок-схему алгоритма «Поиск чисел по условию» (рис. 3.7). Для этого выполнить следующие действия:

- В раскрывающемся списке кнопки «Указатель» выбрать «Выбор лассо» (Lasso Select).

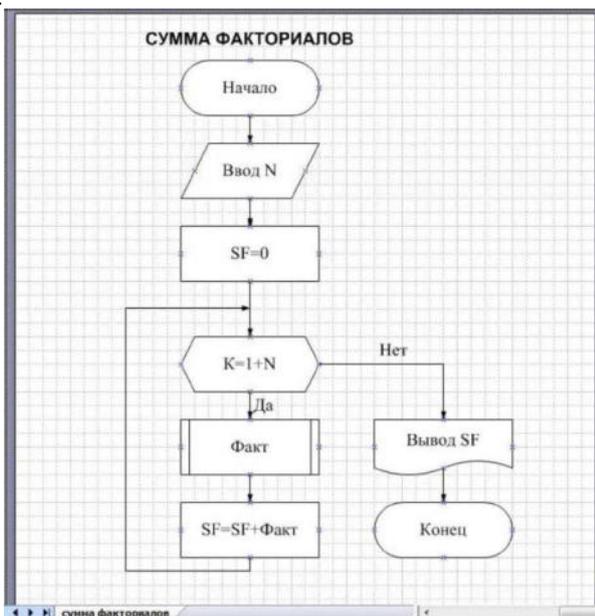


Рисунок 3.6 – Размер стрелок

- Обвести всю схему криволинейным окном.
 - В этом окне выполнить контекстную команду «Копировать» (Copy).
 - Выполнить контекстную команду «Добавить страницу» (Insert Page), выполненную на ярлыке «Сумма факториалов».
 - В окне «Параметры страницы» (Page Setup) ввести имя «Поиск чисел по условию».
 - На странице «Поиск чисел по условию» выполнить контекстную команду «Вставить» (Paste).
 - Изменить схему так, чтобы получилась блок-схема, указанная на рис. 3.7.
25. Сохранить файл с результатами работы.

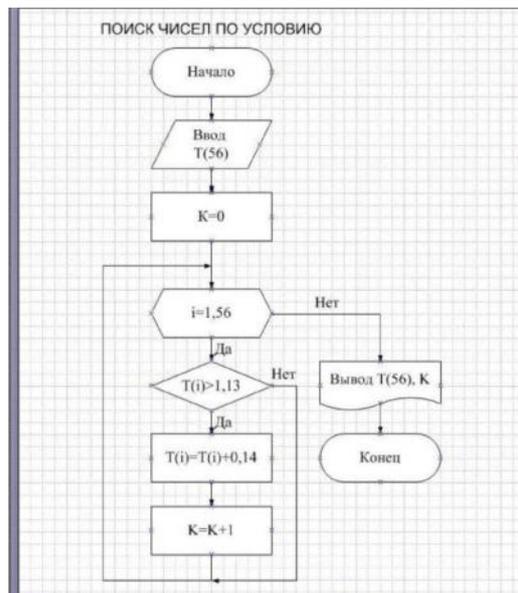


Рисунок 3.7 – Алгоритм «Поиск чисел по условию»

7.2.3 Примерный перечень заданий для решения прикладных задач

Задача 1. Составить программу, вычисляющую площадь сферы и объём шара по заданному радиусу.

Решение. Листинг программы и экран пользователя при работе программы представлены на рис.1.3.

```

program shar;
{ r-радиус шара,
  s-площадь шара}
  var r,s:real;
Begin
  writeln('Вычисляем объём шара и площадь сферы. ');
  write('Введите радиус: ');
  readln(r);
  s:=4*pi*sqr(r);
  writeln('Площадь: ',s:8:3);
  writeln('Объём: ', s/3*r);
  readln;
End.
  
```



Рис.1.3. Программная реализация алгоритма линейной структуры

Задача 2. Составить программу, вычисляющую корни квадратного уравнения по заданным коэффициентам a, b, c .

Решение. Листинг программы и экран пользователя при работе программы представлены на рис.1.4.

```

program kvur;
{ a,b,c-коэффициенты квадратного уравнения,
  D-дискриминант,
  x1,x2-корни}
  var a,b,c,D,x1,x2:real;
Begin
  writeln('Вычисляем корни квадратного уравнения. ');
  write('Введите коэффициенты a b c: ');
  readln(a,b,c);
  D:=sqr(b)-4*a*c;
  if D<0 then writeln('Уравнение корней не имеет. ');
  if D=0 then writeln('x=',-b/(2*a));
  if D>0 then
    begin
      x1:=(-b-sqrt(D))/(2*a);
      x2:=(-b+sqrt(D))/(2*a);
      writeln('x1=',x1,' x2=',x2);
    end;
  readln;
End.

```

```

Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Вычисляем корни квадратного уравнения.
Введите коэффициенты a b c: 1 2 3
Уравнение корней не имеет.

Вычисляем корни квадратного уравнения.
Введите коэффициенты a b c: -1 2.2 3
x1= 3.15182845286836E+0000 x2=-9.51828452867630E-0001

Вычисляем корни квадратного уравнения.
Введите коэффициенты a b c: 1 -2 1
x= 1.00000000000000E+0000

```

Рис.1.4. Программная реализация разветвляющегося алгоритма с использованием оператора if

Задача 3. Написать программу, которая после введенного с клавиатуры числа (в диапазоне 1-99) дописывает слово «копейка» с правильным окончанием

Решение. Листинг программы и экран пользователя при работе программы представлены на рис.1.5.

```

program kop;
{a-количество копеек}
  var a:integer;
Begin
  writeln('Дописываем копейки. ');
  write('Введите число: ');
  readln(a);
  case a mod 10 of
    0,5..9: writeln(a:2,' копеек');
    1: writeln(a:2,' копейка');
    2,3,4: writeln(a:2,' копейки');
  end;
  readln;
End.

```

```

Turbo Pascal
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Дописываем копейки.
Введите число: 51
51 копейка

Дописываем копейки.
Введите число: 20
20 копеек

Дописываем копейки.
Введите число: 25
25 копеек

Дописываем копейки.
Введите число: 73
73 копейки

```

Рис.1.5. Программная реализация разветвляющегося алгоритма с использованием оператора case

Задача 4. Написать программу, которая выводит степени числа 2 от нулево до десятой.

Решение. Листинг программы и экран пользователя при работе программы представлены на рис.1.2.

```

program kvadrat;
{i-переменная цикла,
 k-степень двойки}
var i,k:integer;
Begin
writeln('Таблица квадратов и степеней числа 2. ');
writeln(' N N^2 2^N');
writeln(' 0 0 1');
k:=1;
for i:=1 to 10 do
begin
k:=k*2;
writeln(i:2, ' ',i*i:3, ' ',k:4);
end;
readln;
End.

```

```

Turbo Pascal
Turbo Pascal Version 7.0 Copyright (c) 1983,92 Borland International
Таблица квадратов и степеней числа 2.
 N N^2 2^N
0 0 1
1 1 2
2 4 4
3 9 8
4 16 16
5 25 32
6 36 64
7 49 128
8 64 256
9 81 512
10 100 1024

```

Рис.1.2. Программная реализация циклического алгоритма с использованием счетного оператора цикла

Задача 5. Написать программу табулирования функции $F = (6x^2 - x^3)^{1/2}$ на заданном промежутке с заданным шагом.

Решение. Листинг программы и экран пользователя при работе программы представлены на рис.1.3.

```

program tabl_f;
{ a,b-начало и конец интервала,
  h-шаг,
  x,y-аргумент и значение функции,
  i-переменная цикла}
  var i:integer;
      a,b,h,x,y:real;
Begin
  writeln('Таблица значений функции  $y=(6x^2-x^3)^{(1/3)}$ .');
  write('Введите начало, конец интервала и шаг: ');
  readln(a,b,h);
  writeln('   x           F(x) ');
  if b>a then
    x:=a
  else
    x:=b;
  for i:=1 to trunc(abs((b-a)/h)+1) do
  begin
    if sqr(x)*(6-x)>0 then y:=exp(ln(sqr(x)*(6-x))/3);
    if sqr(x)*(6-x)=0 then y:=0;
    if sqr(x)*(6-x)<0 then y:=-exp(ln(-sqr(x)*(6-x))/3);
    writeln(x:8:3,' ',y:8:3);
    x:=x+h;
  end;
  readln;
End.

```



Рис.1.3. Программная реализация задачи табулирования функции с использованием счетного оператора цикла

Задача 6. Графическими средствами языка Паскаль создать рисунок.

Решение. Листинг программы и экран пользователя при работе программы представлены на рис.1.

```

program qwel;
  USES Graph;
  var d,m:integer;
Begin
  d:=detect;
  InitGraph (d,m,'D:\IN_OUT\206\LabTP\TP\BGI');
  setBKcolor(1);
  setfillstyle(1,11);
  bar(0,300,800,800);

```

```

setcolor(14);
setfillstyle(1,14);
circle(500,100,40);
floodfill(500,100,14);
  setcolor(3);
setfillstyle(1,3);
fillellipse(100,50,100,15);
fillellipse(300,75,125,25);
setcolor(15);
setfillstyle(1,15);
line(250,280,310,280);
  line(250,280,310,150);
  line(310,150,310,280);
    floodfill(260,279,15);
setcolor(8);
setfillstyle(1,8);
line(240,300,200,280);
line(200,280,380,280);
line(380,280,360,300);
line(360,300,240,300);
floodfill(240,283,8);
setcolor(12);
  setfillstyle(1,12);
  line(330,350,365,320);
  line(365,320,365,410);
  line(365,410,330,350);
  floodfill(335,350,12);
setcolor(15);
setfillstyle(1,15);
  line(330,350,350,330);
  line(350,330,430,350);
  line(430,350,350,380);
  line(350,380,330,350);
  line(430,350,445,330);
  line(430,350,445,370);
  line(445,370,445,330);
  floodfill(335,350,15);
setcolor(2);
circle(345,350,4);
readln;

```

End.

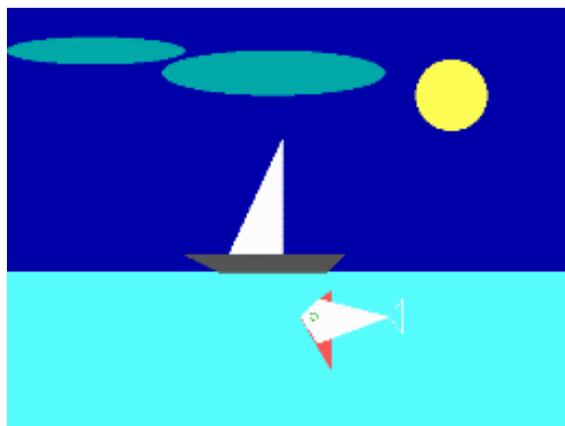


Рис.1. Программная реализация работы с графикой

Задача 7. Изобразить график функции $\sin(x)/x$.

Решение. Листинг программы и экран пользователя при работе программы представлены на рис.2.

```
Program Graphik_Avtomat;           {Построение графика
sin(x)/x}
  uses Graph;
  type Func = function(x: real) : Real;

  {$F+}
  function f(x: Real) : Real;       {Функция для построения}
    begin if x <> 0 then f:= sin(x)/x else f:= 1 end;
  {$F-}
  {Границы изменения аргумента и функции}
  procedure Graphic(xmin,xmax,ymin,ymax: Real; f: Func); {Строит
  график f}
    var GrDr,GrMd,Ox,Oy,i : Integer; {Ox, Oy - положения осей в
  пикселях}
    x,h : Real;                     {Аргумент и шаг его изменения}
    Txt : String;
    function Px(x: real) : Integer;  {Преобразование аргумента
  в пиксели}
      begin Px:= Round(640*(x-xmin)/(xmax-xmin)) end;
    function Py(y: real) : Integer;  {Преобразование функции
  в пиксели}
      begin Py:= Round(480*(ymax-y)/(ymax-ymin)) end;
    begin
      GrDr:= 0;
  InitGraph(GrDr,GrMd,'D:\IN_OUT\206\LabTP\TP\BGI');
      SetBkColor(15); SetColor(9); SetLineStyle(0,0,3);
      LineTo(0,480); LineTo(640,480); LineTo(640,0);
  LineTo(0,0); {Памка}
      SetColor(13); SetLineStyle(3,0,1);
      Ox:= Px(0); Line(Ox,0,Ox,480);           {Построение осей
  координат}
      Oy:= Py(0); Line(0,Oy,640,Oy);         {Не видны вне
  графического окна}
      SetLineStyle(0,0,1);
      for i:= 1 to 5 do                       {Построение делений шкал}
        begin
          Line( Px(i*xmax/5) , Oy - 3 , Px (i*xmax/5) , Oy + 3);
          Line( Px(-i*xmax/5), Oy - 3 , Px(-i*xmax/5) , Oy + 3);
          Line( Ox - 3 , Oy + Py(i*ymax/5), Ox +3, Oy +
  Py(i*ymax/5));
          Line( Ox - 3 , Oy - Py(i*ymax/5), Ox +3, Oy -
  Py(i*ymax/5))
        end;
      Str(xmin :2:2, Txt); OutTextXY(0,Oy-10, Txt);
      Str(xmax :2:2, Txt); OutTextXY(600,Oy-10, Txt);
      Str(ymin :2:2, Txt); OutTextXY(Ox+5,470, Txt);
      Str(ymax :2:2, Txt); OutTextXY(Ox+5,5, Txt);
      SetColor(2);
      x:= xmin; h:= (xmax-xmin)/10000;       {10000 - число
  точек графика}
```

```

repeat                                     {Построение графика по точкам}
    PutPixel( Px(x), Py(f(x)), 10);
    x:= x + h
until x > xmax;
Readln;
CloseGraph;
end;
BEGIN
Graphic(-25, 25, -0.3, 1.2, f);
END.

```

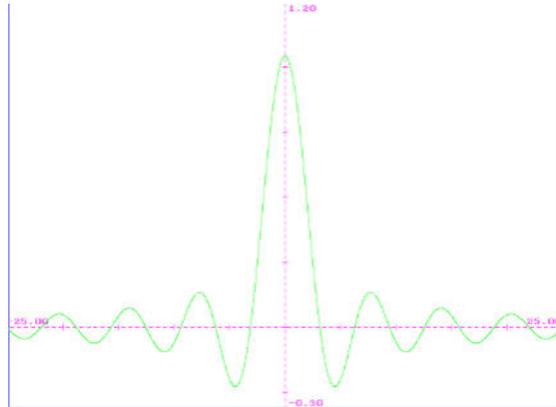


Рис.2. Программная реализация построения графика функции

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Назовите семь этапов решения задач.
2. Дайте определения понятиям алгоритм и программирование.
3. Перечислите основные свойства алгоритмов.
4. Назовите и изобразите основные алгоритмические конструкции.
5. Назовите способы представления алгоритмов. Изобразите основные блоки, используемые в блок-схемах.
6. К какому классу редакторов относится Visio 2007?
7. Что представляет собой файл с расширением .vsd?
8. Что представляет собой файл с расширением .vss?
9. Что представляет собой файл с расширением .vst?
10. Как улучшить просмотр мелких деталей рисунка?
11. Каково назначение и основные возможности редактора *Microsoft Visio*?
12. Опишите элементы окна программы *Microsoft Visio*.
13. Опишите правила и приемы работы с цветами и палитрой в *Microsoft Visio*.
14. Опишите набор инструментов панели инструментов *Microsoft Visio*.
15. Как вставить текст в рисунок *Microsoft Visio* и отформатировать его?
16. Что такое циклический алгоритм?
17. В каком случае используется счетный оператор цикла с параметром?
18. Опишите синтаксис оператора for.
19. Как происходит инициализация графического режима в Паскале?
20. Как происходит переход в графический режим и возврат в

текстовый?

21. Как установить тип графического адаптера по умолчанию?

22. Какие стандартные процедуры и функции работы с графикой вы знаете?

23. Какими способами можно создать Web-страницу в Word?

24. Как Word изменяет структуру папок на диске при сохранении новой Web-страницы?

25. Какие новые приемы оформления документа появляются при работе с Web-страницами? Какие становятся недоступны?

26. Как вставить гиперссылку на другой документ? Как вставить рисунок?

27. Как просмотреть, отредактировать тэги HTML-документа, созданного в Word?

7.2.5 Примерный перечень заданий для решения прикладных задач

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится с группой студентов, в аудитории для проведения практических работ, на индивидуальных персональных компьютерах с привлечением среды программирования, PascalABC.NET - бесплатной, простой и мощной среды разработки с подсказками по коду, авто форматированием и образцами кода для начинающих.

В методических указаниях к практической работе, студентами, выбирается вариант задания, в соответствии с порядковым номером в списке группы. В отчете отображается алгоритм выполнения задания в виде блок-схемы, выполненной в Microsoft Vizio и листинг программы с комментариями выполнения кода.

Зачет выставляется студенту по итогам защиты практических работ, выполненных самостоятельно согласно выбранному варианту.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	ОСНОВЫ АЛГОРИТМИЗАЦИИ	ОПК-2	защита практической работы,
2	СИСТЕМА РАЗРАБОТКИ ДЕЛОВОЙ ГРАФИКИ	ОПК-2	защита практической работы
3	РАЗРАБОТКА ПРОГРАММ ЛИНЕЙНОЙ И РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ	ОПК-2	защита практической работы
4	РАЗРАБОТКА АЛГОРИТМОВ ЦИКЛИЧЕСКОЙ СТРУКТУРЫ	ОПК-2	защита практической работы
5	ГРАФИЧЕСКИЕ СРЕДСТВА АЛГОРИТМИЧЕСКОГО ЯЗЫКА	ОПК-2	защита практической работы
6	СОЗДАНИЕ WEB-ДОКУМЕНТА	ОПК-2	защита практической работы

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Острейковский В.А. Информатика: Учебник / В. А. Острейковский. - 3-е изд. - М. : Высш. шк., 2005. с.325-340.
2. Каймин В.А. Информатика: Учебник / В.А. Каймин. - 4-е изд. - М. : Инфра-М, 2005. с.54-66.
3. Юркин, А.Г. Задачник по программированию: Учебник для вузов / А. Г. Юркин. - СПб. : Питер, 2002. с.33-45.

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

Лицензионное программное обеспечение

Windows Professional 10 Single Upgrade MVL A Each

Academic-операционная система компании Microsoft

Office Professional Plus 2013 Russian OLP NL Academic Edition -пакет офисных приложений компании Microsoft.

Microsoft Visio 2013 Professional- векторный графический редактор, редактор диаграмм и блок-схем для Windows.

RAD Studio XE8 среда быстрой разработки приложений фирмы Embarcadero Technologies, работающая под Windows. Разработка

приложения для Windows на Delphi/Object Pascal и C++

Свободное ПО

PascalABC.NET – это язык программирования Паскаль нового поколения, сочетающий простоту классического языка Паскаль, ряд современных расширений и огромные возможности платформы Microsoft .NET. PascalABC.NET разрабатывается под свободной лицензией LGPLv3 в первую очередь как язык программирования для сферы образования и научных исследований.

LibreOffice — кроссплатформенный, свободно распространяемый офисный пакет с открытым исходным кодом

Firebird (FirebirdSQL) — кроссплатформенная реляционная система управления базами данных, работающая на macOS, Linux, Microsoft Windows

Oracle VM VirtualBox - программный продукт виртуализации для операционных систем Microsoft Windows, Linux, FreeBSD[7], macOS, Solaris/OpenSolaris, ReactOS, DOS и других. Принадлежит корпорации Oracle.

Linux openSuse Leap 15.X- операционная система на базе ядра Linux, разработанная сообществом openSUSE Design System. Один дистрибутив объединяет в себе одновременно рабочую станцию и сервер.

Отечественное ПО

Astra Linux Common Edition - операционная система на базе ядра Linux, разработанная АО «НПО РусБИТех».

Альт Рабочая станция К 9 - операционная система на базе ядра Linux, разработанная ООО «Базальт СПО»

«Программная система для обнаружения текстовых заимствований в учебных и научных работах «Антиплагиат.ВУЗ»»

Модуль «Программный комплекс поиска текстовых заимствований в открытых источниках сети интернет «Антиплагиат-интернет»»

Модуль обеспечения поиска текстовых заимствований по коллекции диссертаций и авторефератов Российской государственной библиотеки (РГБ)

Модуль поиска текстовых заимствований по коллекции научной электронной библиотеки eLIBRARY.RU

Ресурс информационно-телекоммуникационной сети «Интернет»

<http://www.edu.ru/>

Образовательный портал ВГТУ

Информационная справочная система

<http://window.edu.ru>

<https://wiki.cchgeu.ru/>

Современные профессиональные базы данных

IEEE Xplore

Электронная библиотека Институт инженеров по электротехнике и электронике (IEEE) и его партнеров в сфере издательской деятельности.

Адрес ресурса: <https://ieeexplore.ieee.org/Xplore/home.jsp>

SQL

Сайт, посвященный SQL, программированию, базам данных, разработке информационных систем

Адрес ресурса: <https://www.sql.ru/>

OpenNet

На сайте проекта OpenNet размещается информация о Unix системах и открытых технологиях для администраторов, программистов и пользователей

Адрес ресурса: <http://www.opennet.ru/>

Проглаб

Адрес ресурса: <https://proglab.io>

ХабрХабр

Адрес ресурса: <https://habr.com/ru/>

Microsoft Developer Network

Адрес ресурса: <https://msdn.microsoft.com/ru-ru/>

ACMQUEUE

Адрес ресурса: <https://queue.acm.org/>

The Register

На сайте публикуются актуальные новости из области компьютерных технологий; информация о программном обеспечении, сетях, безопасности; интересные видео, форумы и др.

Адрес ресурса: <https://www.theregister.co.uk/>

Driver.ru

Адрес ресурса: <https://driver.ru/>

Хакер

Адрес ресурса: <https://хакер.ru/>

Исходники.ru

На сайте размещается информация по программированию, администрированию и дизайну

Адрес ресурса: <https://forum.sources.ru/>

Инструменты разработчика Firefox

Адрес ресурса: <https://developer.mozilla.org/ru/docs/Tools>

Codewars

Адрес ресурса: <https://www.codewars.com/>

Uikit

Адрес ресурса: <https://getuikit.com/>

Dribbble

Адрес ресурса: <https://dribbble.com/>

Frontender Magazine

Адрес ресурса: <https://frontender.info/>

PR-CY

Адрес ресурса: <https://pr-cy.ru/>

1stWebDesigner

Адрес ресурса: <https://1stwebdesigner.com/>

Weng Vox

Адрес ресурса: <https://medium.com/web-engineering-vox>

NOUPE

Адрес ресурса: <https://www.noupe.com/>

Codrops

Адрес ресурса: <https://tympanus.net/codrops/category/tutorials/>

Bento

Адрес ресурса: <https://bento.io/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Специализированная аудитория, оснащенная оборудованием для лекционных демонстраций, проекционной аппаратурой, компьютерами и компьютерными программами для проведения практических занятий.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Прикладная информатика» проводятся практические занятия.

Практические занятия направлены на приобретение практических навыков программирования на языке высокого уровня. Занятия проводятся путем решения конкретных задач в аудитории.

Вид учебных занятий	Деятельность студента
Практическое занятие	Конспектирование рекомендуемых источников. Работа с конспектом лекций, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы. Прослушивание аудио- и видеозаписей по заданной теме, выполнение расчетно-графических заданий, решение задач по алгоритму.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками,

	<p>дополнительной литературой, а также проработка конспектов лекций;</p> <ul style="list-style-type: none"> - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
<p>Подготовка к промежуточной аттестации</p>	<p>Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начинаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.</p>

6 Лист регистрации изменений

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
1	Актуализирован раздел 8.2 в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных и справочных информационных систем	31.08.2019	
2	Актуализирован раздел 8.2 в части состава используемого лицензионного программного обеспечения, современных профессиональных баз данных и справочных информационных систем	31.08.2020	
3	Проведена актуализация РПД согласно изменению ФГОС ВО (редакция с изменениями №1456 от 26.11.2020). Актуализирован разделы 8.1 и 8.2 в части состава используемого методического обеспечения, а также программного обеспечения, современных профессиональных баз данных и справочных информационных систем	31.08.2021	