




МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДФУ)

**ИНЖЕНЕРНАЯ ШКОЛА**

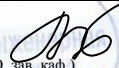
«СОГЛАСОВАНО»

Руководитель ОП  
«Прикладная механика»

 Озерова Г.П.  
(подпись) (Ф.И.О. рук.ОП)  
«24»июня 2017г.

«УТВЕРЖДАЮ»

Заведующая кафедрой  
Механики и математического моделирования  
(название кафедры)

 Бочарова А.А.  
(подпись) (Ф.И.О. зав. каф.)  
«23» июня 2017г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ (РПУД)**  
**ИНФОРМАЦИОННЫЕ И КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ В ПРИКЛАДНОЙ МЕХАНИКЕ**

**Направление подготовки: 15.03.03 Прикладная механика**

Профиль подготовки:

«Математическое и компьютерное моделирование механических систем и процессов»

**Форма подготовки (очная)**

курс 1,2 семестр 2,3  
лекции 18 час.  
практические занятия 72 час.  
лабораторные работы - час.  
в том числе с использованием МАО лек. - час. /пр.36 час.  
всего часов аудиторной нагрузки 90 час.  
в том числе с использованием МАО 36 час.  
самостоятельная работа 54 час.  
в том числе на подготовку к экзамену - час.  
контрольные работы -  
курсовая работа / курсовой проект -  
зачет -2,3 семестр  
экзамен - семестр

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования Дальневосточного федерального университета, принятого решением Ученого совета ДВФУ, протокол от 25.02.2016 № 02-16, введенного в действие приказом ректора ДВФУ от 10.03.2016 № 12-13-391

Рабочая программа обсуждена на заседании кафедры Механики и математического моделирования, протокол № 9 от «23» июня 2017 г.

Заведующая кафедрой: к.ф.-м.н., проф. Бочарова А.А.  
Составитель: к.т.н., доцент Озерова Г.П.

**Оборотная сторона титульного листа РПУД****I. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)**II. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от «\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

## ABSTRACT

### **Bachelor's degree in 15.03.03 Applied Mechanics**

### **Study profile “Mathematical and Computer Modeling Mechanical Systems and Processes”**

**Course title:** Computer Technology in Apply Mechanics

**Basic part of Block 1, 3 credits**

**Instructor:** Ozerova Galina Pavlovna

**At the beginning of the course a student should be able to:**

- should know how to use different information sources: books, textbooks, references, internet;
- should have an algorithmic mind, that was formed by high school classes “Algebra”, “Geometry”, “Informatics”
- can use computational machines;
- can use basic textual and graphical editors, office applications.

**Learning outcomes:**

GPC 7 – the ability to use the advanced design software and technological documentation

GPC 10 – the ability to solve common tasks of professional activities on the basis of bibliographic information and culture using information and communication technologies, taking into account the main requirements for information security;

PC 2 – the ability to apply physical and mathematical tools, theoretical, computational and experimental research methods, methods of mathematical and computer modeling in the professional activities.

**Course description:** The content of the course covers the following range of issues: the design and development of reliable, high-quality software for engineering calculations using modern programming technologies. All stages of development programs discussed in detail in the course. Theoretical and practical knowledge of programming will allow students more confident to navigate the

complex and diverse phenomena of the modern development of the information society, as well as some practical skills in the use of computer technology in teaching, research and experimental activities.

**Main course literature:**

1. Trofimov V.V. Informatics. Textbook for high schools. - M: Yurayt 2010. -911 p. <http://lib.dvfu.ru:8080/lib/item?id=chamo:356824&theme=FEFU>
2. Pavlovskaya T.A .Pascal. Programming in high-level programming language: textbook for high schools. - St. Petersburg: Peter, 2008. - 393 p. <http://lib.dvfu.ru:8080/lib/item?id=chamo:384229&theme=FEFU>
3. Algorithmic and programming: Textbook / SA Kantsedal. - M .: FORUM: SIC INFRA-M, 2013. - 352 p. <http://znanium.com/go.php?id=391351>
4. Nemtsov T.I. Programming in high-level programming language. Programming in Object Pascal: Tutorial / TI Nemtsov; Ed. LG Gagarina. - M .: FORUM: INFRA-M, 2015. - 496 p. <http://znanium.com/bookread.php?book=472870>
5. Zelenyak O.P. Programming on Turbo Pascal. Tasks, algorithms & solutions [electronic resource]. - Moscow: DMK Press, 2009. - 311 p. [http://e.lanbook.com/books/element.php?pl1\\_id=1249](http://e.lanbook.com/books/element.php?pl1_id=1249)

**Form of final knowledge control:** exam.

## Аннотация дисциплины

### «Информационные технологии в прикладной механике»

Дисциплина «Информационные технологии в прикладной механике» разработана для студентов, обучающихся по направлению подготовки 15.03.03 «Прикладная механика», профиль «Математическое и компьютерное моделирование механических систем и процессов» и является обязательной дисциплиной базовой части Блока 1 «Дисциплины (модули)» учебного плана (Б1.Б.29).

Трудоемкость дисциплины составляет 108 часов (3 зачетные единицы). Учебным планом предусмотрены лекционные занятия (18 часов), практические занятия (36 часов), и самостоятельная работа студентов (54 часа, из них 36 часов на экзамен). Дисциплина реализуется на 1 курсе во 2 семестре. Форма промежуточной аттестации – экзамен.

Дисциплина «Информационные технологии в прикладной механике» логически связана с дисциплинами «Современные информационные технологии», «Программирование в инженерных задачах».

**Цель дисциплины:** формирование теоретических и практических навыков по разработке надежного, качественного программного обеспечения для инженерных расчетов с применением современных технологий программирования, методов и средств коллективной разработки.

**Задачи дисциплины:**

- Дать целостное представление о возможностях вычислительной техники, современном ее состоянии и тенденциях развития.
- Сформировать умение ставить задачу для решения ее на компьютере, а также реализовать ее средствами имеющейся вычислительной техники.
- Изучить основы структурного программирования, типы данных и конструкции языка высокого уровня
- Сформировать умение реализовывать инженерные вычислительные задачи средствами языка программирования.

- Дать методику, позволяющую свободно изучать и применять новые программные системы.

- Развить логическое и алгоритмическое мышление.

Для успешного изучения дисциплины «Информационные технологии в прикладной механике» у обучающихся должны быть сформированы следующие предварительные компетенции:

- владение навыками работы с различными источниками информации: книгами, учебниками, справочниками, Интернет;
- способность к алгоритмическому мышлению, сформированное в рамках предметов общеобразовательной школы «Алгебра», «Геометрия», «Информатика и ИТК»;
- владение навыками работы с вычислительной техникой;
- способность использовать простейшие текстовые и графические редакторы, офисные программы.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют этапы формирования следующих компетенций:

Код и формулировка компетенции	Этапы формирования компетенции	
<b>ОПК-7</b> умением использовать современные программные средства подготовки конструкторско-технологической документации	Знает	методы описания графической информации и реализации алгоритмов компьютерной графики
	Умеет	программно реализовывать основные алгоритмы растровой графики
	Владеет	системами разработки программного обеспечения и графическими редакторами.
<b>ОПК-10</b> способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-	Знает	алгоритмы поиска, сортировки и представления данных
	Умеет	использовать технологическую и справочную документацию при разработке программ
	Владеет	способностью использовать современные технические средства и информационные технологии для

Код и формулировка компетенции	Этапы формирования компетенции	
коммуникационных технологий и с учетом основных требований информационной безопасности		решения аналитических, исследовательских и коммуникативных задач;
ПК-2 способностью применять физико-математический аппарат, теоретические, расчетные и экспериментальные методы исследований, методы математического и компьютерного моделирования в процессе профессиональной деятельности	Знает	основы структурного программирования
	Умеет	разрабатывать алгоритмы и программы для решения задач профессиональной деятельности средствами языков программирования
	Владеет	методологией проектирования и реализации программных продуктов в современных системах программирования

Для формирования вышеуказанных компетенций в рамках дисциплины «Информационные технологии в прикладной механике» применяются следующие методы активного/ интерактивного обучения: лекция пресс-конференция; лекция «вдвоем»; деловая игра; лекция пресс-конференция; групповая консультация.

## **I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА**

### **Раздел I. Основы структурного программирования (10 час.)**

#### **Тема 1. Классификация языков программирования (2 час.)**

Алгоритм и программа. Поколения языков программирования. Классификация языков программирования. Процедурное, объектно-ориентированное, функциональное и логическое программирование. Системы программирования. Языки программирования высокого уровня; Назначение языков программирования. Этапы разработки программы.

#### **Тема 2. Основы программирования на языке высокого уровня (4 час.)**

Синтаксис и семантика программы. Структура программы на языке Паскаль. Понятие синтаксиса и семантики. Представление данных в алгоритмических языках. Простейшие типы данных: INTEGER, REAL, BOOLEAN. Конструкции структурного программирования. Операторы

присваивания, ввода, вывода, условные операторы. Этапы разработки программы: проектирование и реализация. Простейшие алгоритмы обработки данных: геометрические и алгебраические вычисления. Особенности обработки целых и вещественных чисел.

### **Тема 3. Операторы повторения (4 час.)**

Назначение операторов цикла. Выделение повторяющихся действий. Различные типы циклов. Цикл со счетчиком. Алгоритмы нахождения минимальных и максимальных значений, сумм и произведений. Цикл с предусловием. Алгоритм разбиения числа на цифры. Цикл с постусловием. Алгоритмы нахождения наибольшего общего делителя и определения простоты числа. Алгоритмы решения вычислительных задач, использующих операторы повторения.

## **Раздел II. Структурированные типы данных (4 час.)**

### **Тема 1. Одномерные массивы (4 час.)**

Назначение и структура массивов. Описание одномерного массива. Типовые операции над массивами: ввод, вывод, заполнение случайным образом, простые переборные алгоритмы (поиск максимума/минимума, суммы, среднего арифметического и др.), вставка нового элемента в массив, удаление элемента из массива. Структура программ поиска и сортировки. Поиск «со сторожем», двоичный поиск. Методы сортировки данных – алгоритмы «пузырька», обмена и вставки.

Описание многомерных массивов. Типовые операции над двумерными массивами: ввод, вывод, заполнение, переборные алгоритмы, удаление и вставка строк и столбцов. Арифметические операции над матрицами и векторами. Сложение матриц и векторов, умножение матриц, умножение матрицы на вектор. Скалярное произведение векторов

## **Раздел III. Графика и визуализация (4 час.)**

### **Тема 1. Основы растровой графики(2 час.)**

Работа в графическом режиме. Графическое окно. Рисование точек. Функции пера. Функции кисти. Функции шрифта. Графические примитивы.



Функции канваса. Основы программного рисования. Построение графика функции.

## **Тема 2. Методы изображения и преобразования графических изображений (2 час.)**

Рисование в относительных координатах. Масштабирование. Поворот. Реализация движение «стиранием». Реализация движения через дополнительный буфер. Алгоритмы рисования.

## **II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА**

### **Практические занятия (18 час.)**

#### **Занятие 1. Классификация языков программирования (2 час.)**

1. Алгоритм и программа
2. Поколения языков программирования
3. Классификация языков программирования
4. Системы программирования
5. Этапы разработки программы
6. Свойства алгоритма

#### **Занятие 2. Структура программы, простейшие типы данных, оператор присваивания(2 час.)**

1. Общий вид программы на языке паскаль
2. Раздел описания переменных
3. Целые и вещественные типы данных
4. Взаимное преобразование целых и вещественных чисел
5. Тело программы
6. Оператор присваивания
7. Реализация программ в системе программирования Pascal ABC

#### **Занятие 3. Условный оператор, операторы ввода/вывода (2 час.)**

1. Условные операторы
2. Операторы вывода
3. Операторы ввода

4. Тип данных BOOLEAN
5. Описание констант
6. Реализация программ в системе программирования Pascal ABC

#### **Занятие 4. Решение задач на тему «Выражения и ветвления» (2 час.)**

1. Вычислительные геометрические задачи.
2. Алгоритм разбиения числа на цифры.
3. Логические и текстовые задачи.

1. Реализация программ в системе программирования Pascal ABC

#### **Занятие 5. Этапы разработки программы (2 час.)**

1. Неформальная постановка задачи.
2. Формальная постановка задачи.
3. Проектирование задачи, разработка алгоритма.
4. Реализация задачи.
5. Отладка и тестирование

#### **Занятие 6. Операторы повторения (2 час.)**

1. Семантика операторов повторения.
2. Цикл с параметрами.
3. Цикл с предусловием.
4. Цикл с постусловием.
5. Реализация программ в системе программирования Pascal ABC

#### **Занятие 7. Решение задач на тему «Операторы повторения»(2 час.)**

1. Поиск максимума и минимума последовательности чисел.
2. Вычисления суммы, произведения, среднего арифметического последовательности чисел.
3. Вычисления суммы ряда для заданного числа членов и с заданной точностью.
4. Реализация программ в системе программирования Pascal ABC

#### **Занятие 8. Решение задач на тему «Операторы повторения, вложенные циклы» (2 час.)**

1. Понятие вложенных циклов.

2. Решение переборных задач.
3. Решение вычислительных задач со вложенными циклами.
4. Реализация программ в системе программирования Pascal ABC

**Занятие 9. Проектирование и реализация сложной задачи с использованием циклов (2 час.)**

1. Неформальная постановка задачи.
2. Формальная постановка задачи.
3. Проектирование задачи, разработка алгоритма.
4. Реализация задачи в системе программирования Pascal ABC.
5. Отладка и тестирование

**Занятие 10. Одномерные массивы (2 час.)**

1. Определение структурированных скалярных типов данных
2. Описание одномерного массива
3. Типовые операции над массивами: ввод, вывод, заполнение случайным образом, простые переборные алгоритмы (поиск максимума/минимума, суммы, среднего арифметического и пр.)
4. Реализация программ в системе программирования Pascal ABC

**Занятие 11. Одномерные массивы, продолжение (2 час.)**

1. Типовые операции над массивами: вставка нового элемента в массив, удаление элемента из массива
2. Алгоритмы поиска
3. Реализация программ в системе программирования Pascal ABC

**Занятие 12. Сортировка одномерных массивов (2 час.)**

1. Определение сортировки
2. Сортировка «пузырьком»
3. Сортировка вставками
4. Сортировка перестановками

**Занятие 13. Решение задач на тему «Одномерные массивы» (2 час.)**

1. Задачи, сводящиеся к вычислению минимума, максимума
2. Вычислительные задачи.

3. Задачи на комбинированные алгоритмы.
4. Реализация программ в системе программирования Pascal ABC

**Занятие 14-15. Проектирование и реализация сложной задачи с использованием одномерных массивов (2 час.)**

1. Неформальная постановка задачи.
2. Формальная постановка задачи.
3. Проектирование задачи, разработка алгоритма.
4. Реализация задачи в системе программирования Pascal ABC.
5. Отладка и тестирование.

**Занятие 16. Программное рисование (2 час.)**

1. Работа в графическом режиме.
2. Графическое окно.
3. Рисование точек.
4. Графические примитивы.
5. Реализация программ в системе программирования Pascal ABC

**Занятие 17. Программное рисование: масштабирование, поворот (2 час.)**

1. Функции пера.
2. Функции кисти.
3. Шрифт.

**Занятие 18. Программное рисование: реализация движения. (2 час.)**

1. Рисование в относительных координатах.
2. Масштабирование.
3. Поворот.
4. Реализация движение «стиранием».
5. Реализация движения через дополнительный буфер.
6. Реализация программ в системе программирования Pascal ABC

**III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ  
САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Информационные технологии в прикладной механике» представлено в Приложении 1 и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
- характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- требования к представлению и оформлению результатов самостоятельной работы;
- критерии оценки выполнения самостоятельной работы.

#### IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
1	Основы структурного программирования	ОПК-10	Знает алгоритмы поиска, сортировки и представления данных	ПР-1	Вопросы 1-8
			Умеет использовать технологическую и справочную документацию при разработке программ	ПР-12 ПР-6	Задачи 1-7
			Владеет способностью использовать современные технические средства и информационные технологии для решения аналитических, исследовательских и коммуникативных задач;	ПР-12	Задачи 18-25
2	Структурированные типы данных	ПК-2	Знает основы структурного программирования	УО-1	Вопросы 18-25
			Умеет разрабатывать алгоритмы и программы для решения задач профессиональной деятельности средствами языков программирования	ПР-12 ПР-6	Задачи 7-14
			Владеет методологией проектирования и реализации программных продуктов в современных системах программирования	ПР-12	Задачи 26-30

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
3	Графика и визуализация	ОПК- 7	Знает методы описания графической информации и реализации алгоритмов компьютерной графики	УО-2	Вопросы 27- 33
			Умеет программно реализовывать основные алгоритмы растровой графики	ПР-6	Задачи 15-17
			Владеет системами разработки программного обеспечения и графическими редакторами.	ПР-12	Задачи 31-33

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

## V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### Основная литература

1. Трофимов В. В. Информатика. Учебник для вузов. – М: Юрайт. – 2010. –911 с <http://lib.dvfu.ru:8080/lib/item?id=chamo:356824&theme=FEFU>
2. Павловская Т. А. Паскаль. Программирование на языке высокого уровня : учебник для вузов. – Санкт-Петербург : Питер , 2008. – 393 с. <http://lib.dvfu.ru:8080/lib/item?id=chamo:384229&theme=FEFU>
3. Алексеев, Е.Р. Free Pascal и Lazarus: Учебник по программированию [Электронный ресурс] : учебник / Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер. — Электрон. дан. — М. : ДМК Пресс, 2010. — 438 с. [http://e.lanbook.com/books/element.php?pl1\\_id=1267](http://e.lanbook.com/books/element.php?pl1_id=1267)

4. Озерова, Г.П.Pascal в примерах и задачах [Электронный ресурс] : учебное пособие для вузов / Г. П. Озерова ; Дальневосточный федеральный университет, Инженерная школа. – Владивосток: Изд-во Дальневосточного федерального университета, 2014 , 123с

<http://elib.dvfu.ru/vital/access/manager/Repository/fefu:2537>

5. Немцова Т. И. Программирование на языке высокого уровня.

Программирование на языке Object Pascal: Учебное пособие / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с.

<http://znanium.com/bookread.php?book=472870>

### **Дополнительная литература**

1. Парфилова Н. И. Программирование. Основы алгоритмизации и программирования : учебник для вузов / Н. И. Парфилова, А. Н. Пылькин, Б. Г. Трусов – Москва : Академия , 2014. - 240 с.

<http://lib.dvfu.ru:8080/lib/item?id=chamo:790355&theme=FEFU>

2. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона [Электронный ресурс] : учебное пособие. — Электрон. дан. — М. : ДМК Пресс, 2010. — 272 с. [http://e.lanbook.com/books/element.php?pl1\\_id=1261](http://e.lanbook.com/books/element.php?pl1_id=1261)

### **Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. Сайт свободно распространяемой системы программирования PascalABC. – Режим доступа: <http://pascalabc.net/>

2. Примеры программ на языке Паскаль. – Режим доступа: <http://pascalabc.net/wiki/>

3. Справочник по Паскалю. - <http://tpdn.ru/>

### **Перечень информационных технологий и программного обеспечения**

Необходимое программное обеспечение: свободно распространяемая система программирования PascalABC (<http://pascalabc.net/>).

Материалы курса размещены в LMS BlackBoard, идентификатор: FU50219-15.03.03-ITvPM-01.

## **VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

## **1. Описание последовательности действий студента («сценарий изучения дисциплины»).**

Для успешного изучения дисциплины «Информационные технологии в прикладной механике» студенту необходимо:

1. При подготовке к лекции изучить конспект предыдущей лекции, а также при необходимости теоретический материал, представленный в системе BlackBoard. Далее ознакомиться с материалом следующей лекции (также через систему BlackBoard), вникнуть в суть изучаемой проблемы, подготовить вопросы.

2. На лекционном занятии тщательно конспектировать теоретический материал, участвовать в обсуждении, задавать вопросы.

3. При подготовке к практическим занятиям на основе материалов, представленных в системе BlackBoard, сначала необходимо понять задания, которые предложены на практическом занятии, затем найти необходимый теоретический материал, изучить алгоритм реализации задания, сформулировать вопросы преподавателю.

4. На практическом занятии сначала выполнить задание «по образцу». Отправить преподавателю через систему BlackBoard выполненное задание на проверку, ответить на вопросы преподавателя. Прежде, чем приступить к выполнению самостоятельных заданий, обдумать алгоритм их реализации, задать вопросы преподавателю по сути заданий, спланировать их выполнение.

5. Расчетно-графические задания выполняются обучающимся самостоятельно во внеаудиторное время. При этом результат их реализации необходимо проверять через систему BlackBoard и только после этого отправлять преподавателю на окончательную проверку и оценивание.

6. В течение недели выбрать время для работы со специальной литературой в библиотеке и для занятий на компьютере.

7. Самостоятельную работу организовывать в соответствии с графиком выполнения самостоятельной работы, приведенном в приложении 1.



**2. Рекомендации по использованию материалов учебно-методического комплекса.** Рекомендуется использовать методические указания и материалы по курсу «Информационные технологии в прикладной механике», размещенные в системе BlackBoard, идентификатор курса FU50219-15.03.03-ITvPM-01.

**3. Рекомендации по работе с литературой.** Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекций, использованию учебно-методического комплекса, представленного в системе BlackBoard, изучаются и книги из списка основной и дополнительной литературы. Литературу по курсу можно изучать в библиотеке, брать книгу на дом или читать ее на компьютере (если это электронный ресурс). Полезно использовать несколько учебников, однако желательно придерживаться рекомендации преподавателя по выбору книг по каждому разделу. Не рекомендуется «заучивать» материал, желательно добиться понимания изучаемой темы дисциплины, а затем использовать изученный материал для реализации программ. Кроме того, очень полезно выявить тенденции развития той или иной технологии разработки, выделить для себя направления дальнейшего изучения материала, для достижения более продвинутого уровня изучения дисциплины.

**4. Рекомендации по подготовке к экзамену.** Успешная подготовка к экзамену включает, с одной стороны, добросовестную работу в течение семестра, выполнение всех заданий преподавателя, а с другой – правильную организацию процесса непосредственной подготовки. При подготовке к экзамену необходимо освоить теорию: разобрать определения всех понятий, повторить синтаксис и семантику изучаемых языков программирования, типовые алгоритмы решения задач. Затем рассмотреть примеры и самостоятельно реализовать задания из каждой темы. При этом, если студент формулирует и реализует алгоритм самостоятельно, – достигается более продвинутый уровень изучения дисциплины.

## **VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Аудиторные занятия по дисциплине включают лекции, практические занятия и лабораторные работы.

Для проведения лекционных и практических занятий необходима аудитория со следующим оборудованием:

- мультимедийный проектор Optima EX542I – 1 шт;
- аудио усилитель QVC RMX 850 – 1 шт;
- колонки – 1 шт;
- ноутбук;
- ИБП – 1 шт;
- настенный экран;
- микрофон – 1 шт.

Лабораторные работы проводятся в компьютерном классе, в котором должно быть установлено:

– 15 моноблоков Lenovo C360G-i34164G500UDK с установленным стандартным программным обеспечением, свободно-распространяемой системой PascalABC, доступом к сети Интернет.

- мультимедийный проектор OptimaEX542I – 1 шт;
- настенный экран;
- аудио усилитель QVC RMX 850 – 1 шт;
- колонки – 1 шт;
- ИБП – 1 шт;
- микрофон – 1 шт.
- документ-камера.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Дальневосточный федеральный университет»**  
(ДФУ)

---

---

**ИНЖЕНЕРНАЯ ШКОЛА**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ  
РАБОТЫ ОБУЧАЮЩИХСЯ**

**по дисциплине «Информационные технологии в прикладной механике»**

**Направление подготовки: 15.03.03 Прикладная механика**

**Профиль подготовки: «Математическое и компьютерное моделирование  
механических систем и процессов»**

**Форма подготовки очная**

**Владивосток**

**2016**

## План-график выполнения самостоятельной работы по дисциплине

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1	4 неделя семестра	Решение задач по теме «Выражения и ветвления»	3 час.	ПР-12
2	8 неделя семестра	Решение задач по теме «Операторы повторения»	3 час.	ПР-12
3	10 неделя семестра	Подготовка к тесту по разделу «Основы структурного программирования»	3 час.	ПР-1
4	11 неделя семестра	Решение задач по теме «Одномерные массивы»	3 час.	ПР-12
	13 неделя семестра	Решение задач по теме «Графика и визуализация»	3 час.	ПР-12
12	17 неделя семестра	Подготовка к устному опросу по разделу «Графика и визуализация»	3 час.	УО-2
5	Экзаменационная сессия	Подготовка к экзамену	36 час.	Экзамен
<b>Итого</b>			<b>54 час.</b>	

### Характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению

#### Расчетно-графические задания

Каждая работа состоит из нескольких заданий, в каждом задании требуется решить задачу определенного типа. Варианты заданий приведены в Приложении 2. Все задачи реализуются на языке Паскаль в системе PascalABC. Перед сдачей преподавателю задания, его необходимо проверить с помощью специального инструмента, реализованного в виде web-приложения и размещенного в курсе «Информационные технологии в прикладной механике» в LMS BlackBoard. Далее приводятся образцы решения каждого расчетно-графического задания.

#### Расчетно-графическая работа по разделу «Основы структурного программирования», тема «Выражение и ветвление», базовый уровень

Задача 1. Дан треугольник, заданный координатами трех его вершин. Определить, лежат ли хотя бы две его вершины в одной четверти координатной плоскости.

Решение:

```

VAR
  x1, y1, x2, y2, x3, y3, r1, r2, r3 : REAL;
BEGIN
  WRITELN('введите координаты трех точек');
  READLN (x1, y1, x2, y2, x3, y3);
  { проверить, может ли треугольник быть задан этими точками, по
  правилу треугольника }
  r1 :=sqrt( sqr(x1 - x2) + sqr(y1 - y2));
  r2:=sqrt( sqr(x2 - x3) + sqr(y2 - y3));
  r3:=sqrt( sqr(x3 - x1) + sqr(y3 - y1));
  IF (r1 >= r2 + r3) OR (r2 >= r1 + r3) OR (r3 >= r2 + r1)
  THEN
    WRITELN('точки не могут быть вершинами треугольника')
  ELSE
    BEGIN
      { две точки лежат в одной четверти, если произведение их
      первых и вторых координат положительно }
      IF ((x1*x2 >= 0) AND (y1*y2 >= 0)) OR
      ((x1*x3 >= 0) AND (y1*y3 >= 0)) OR
      ((x3*x2 >= 0) AND (y3*y2 >= 0))
      THEN
        WRITELN('две вершины лежат в одной четверти')
      ELSE
        WRITELN('все вершины лежат в разных плоскостях')
      END
    END.

```

Задача 2. Определить, равна ли сумма двух первых цифр заданного четырехзначного числа сумме двух его последних цифр.

Решение:

```

VAR
  num, a1, a2, a3, a4: INTEGER;
BEGIN
  WRITELN('введите четырехзначное число');
  READLN(num);
  IF (num < 1000) OR (num > 9999) THEN
    WRITELN('введено неверное значение')
  ELSE
    BEGIN
      a1 := num MOD 10; {выделение последней цифры}
      num := num DIV 10; {уменьшение разряда числа}
      a2 := num MOD 10;
      num := num DIV 10;
      a3 := num MOD 10;
      num := num DIV 10;
      a4 := num MOD 10;
      IF (a1 + a2)=(a3 + a4) THEN WRITELN('сумма цифр равна')
      ELSE WRITELN('сумма цифр не равна');
    END
  END.

```

Задача 3. Выяснить, являются ли поля (k,l) и (m,n) шахматной доски полями одного цвета.

Решение:

```

VAR
  k, l, m, n : INTEGER;
BEGIN
  WRITELN('введите координаты двух шахматных полей');
  READ(k, l, m, n);
  IF (k<1) OR (k>8) OR (l<1) OR (l>8) OR (m<1)
    OR (m>8) OR (n<1) OR (n>8)
  THEN WRITE ('введены неправильные координаты')
  ELSE
    { поля имеют одинаковый цвет, если сумма их координат
      является для обоих полей либо четной, либо нечетной }
    IF ODD(k+l) = ODD(n+m)
    THEN WRITELN('поля имеют одинаковый цвет')
    ELSE WRITELN('поля имеют разный цвет')
  END.

```

**Расчетно-графическая работа по разделу «Основы структурного программирования», тема «Операторы повторения», продвинутый уровень**

Задача 1. Дана последовательность из  $n$  элементов. Найти максимальный элемент последовательности и его номер.

Решение:

```

VAR
  num, max: REAL;
  n, i, nMax: INTEGER;
BEGIN
  WRITE('n='); READ(n);
  IF n>0 THEN
    BEGIN
      WRITE('num='); READ(num);
      max:=num; {считаем максимальным первое введенное число}
      nMax:= 1; {запоминаем его номер}
      FOR i:= 2 TO n DO
        BEGIN
          WRITE('num='); READ(num);
          IF num > max THEN {найдено число, больше максимального}
            BEGIN
              {запоминаем новый максимум и его номер}
              max:= num;
              nMax:= i;
            END;
        END;
      END;
      WRITELN('max=', max, ' nMax=', nMax);
    END;
  END;

```

```

END
ELSE WRITELN('ошибочный ввод');
END.

```

Задача 2. Вычислить сумму ряда для заданного числа членов **n**:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

Решение:

```

VAR
  n, i:INTEGER;
  sum, r, x: REAL;
BEGIN
  WRITE('n='); READ(n);
  WRITE('x='); READ(x);
  IF n > 0 THEN
  BEGIN
    r:= x;
    sum:= r;
    FOR i:=1 TO n-1 DO
    BEGIN
      { используем предыдущее значение r }
      r:=r*(-1)*sqr(x)/(2*i*(2*i+1));
      sum:=sum + r;
    END;
    WRITELN('sum=',sum, 'sin(x)= ',sin(x));
  END
  ELSE WRITE('ошибочный ввод');
END.

```

Задача 3. Подобрать всевозможные варианты размена произвольной >7 руб. суммы с помощью 3-х и 5-ти рублевых купюр.

Решение:

```

VAR
  x, y, sum :INTEGER;
BEGIN
  WRITE('sum='); READ(sum);
  IF sum > 7 THEN
  BEGIN
    FOR x :=0 TO sum div 3 DO
      FOR y :=0 TO sum div 5 DO
        IF x*3 + y*5 = sum
        THEN WRITELN('3*',x, '+5*',y, '=',sum);
      END
    END
  ELSE WRITE('ошибочный ввод');
END.

```

## Расчетно-графическая работа по разделу «Структурированные типы данных», тема «Одномерные массивы»

Задача 1. Найти максимальный элемент в массиве А и его место в массиве.

Решение:

```

const
  n1=20;
var
  a:array [1..n1] of real;
  n, i, k:integer;
  max:real;
begin
  write('n='); readln(n);
  if (n<=0) or (n>n1) then writeln('error ', n)
  else
    begin
      for i:=1 to n do
        begin
          write('a[' ,i, ']= '); readln(a[i]);
        end;
      {предполагаем, что первый элемент - максимальный}
      max:= a[1]; k:=1;
      for i:=2 to n do
        if max<a[i] then
          begin
            {запоминаем новый максимум и его номер}
            max:=a[i]; k:=i;
          end;
      writeln('max=',max, ' k=',k);
    end;
  readln;
end.

```

Задача 2. Удалить из массива А элемент с номером k.

Решение:

```

const
  n1=20;
var
  a:array [1..n1] of real;
  n, i, k, j:integer;
begin
  write('n='); readln(n);
  {проверку n выполните самостоятельно}
  for i:=1 to n do
    begin
      write('a[' ,i, ']= '); readln(a[i]);
    end;
  write('k='); readln(k);

```



```

if (k<=0) or (k>n) then writeln('error ',k)
else
  begin
    {сдвигаем элементы массива на одну позицию влево,
    начиная с k-го элемента}
    for j:=k to n-1 do
      a[j]:=a[j+1];
    n:=n-1;
    {Вывод элементов массива}
    writeln('new A');
    for i:=1 to n do
      write(a[i], ' ');
    end;
    readln;
end.

```

**Задача 3.** Определить количество различных элементов массива.

**Решение:**

```

const n1=20;
var
k, j, i: integer;
a, b: array [1..n1] of integer;
begin
  write('n= '); readln(n);
  for i:=1 to n do
    begin
      write('a[' ,i, ']= '); readln(a[i]);
    end;

  k:=0;
  for i:=1 to n do
    begin
      {заносим очередной элемент в конец массива B}
      b[k+1]:=a[i];
      j:=1;
      while b[j]<>a[i] do j:=j+1;
      if j=(k+1) {такого элемента в массиве не было}
      then k:=k+1;{сдвигаем указатель конца массива B}
    end;

  writeln('count= ',m);
  write('different ');
  for j:=1 to k do
    write(a[j], ' ');
  end;
  readln;
end.

```

## Расчетно-графическая работа по разделу «Графика и визуализация»

Задача 1. Нарисовать фигуру «кораблик» и реализовать ее движение по горизонтали, так, чтобы фигура не выходила за границу окна.

Решение:

```

uses GraphABC;
var
  i, x, y, h, kx, ky: integer;
procedure drawShip(x,y: integer; kx,ky: real);
begin
  { корпус корабля}
  setPenColor(clBlack);
  MoveTo(round(x+1*kx),round(y +18*ky));
  LineTo(round(x+21*kx),round(y+14*ky));
  LineTo(round(x+18*kx),round(y+20*ky));
  LineTo(round(x+1*kx),round(y+20*ky));
  LineTo(round(x+1*kx),round(y +18*ky));
  //закрашиваем область
  FloodFill(round(x+5*kx),round(y +18*ky),clBlue);
  { парус}
  MoveTo(round(x+5*kx),round(y+17*ky));
  LineTo(round(x+10*kx),round(y+3*ky));
  LineTo(round(x+20*kx),round(y+14*ky));
  LineTo(round(x+5*kx),round(y+17*ky));
  FloodFill(round(x+7*kx),round(y +15*ky),clRed);
  { флаг }
  setBrushStyle(bsHatch);
  setBrushColor(clGreen);
  rectangle( round(x+10*kx),round(y+3*ky),
             round(x+16*kx),round(y+1*ky));
  {иллюминаторы}
  setBrushStyle(bsSolid);
  setBrushColor(clWhite);
  circle(round(x+16*kx),round(y+17*ky),round(kx*1));
  circle(round(x+12*kx),round(y+17.5*ky),round(kx*1));
  circle(round(x+8*kx),round(y+18*ky),round(kx*1));
end; { DrawShip }

begin
  {SetWindowSize(700,500);}
  x:=20; y:=20; kx:=5; ky:=5; h:=2;
  LockDrawing;
  While x<= (windowWidth - 22*kx) do
  begin
    ClearWindow;
    {рисует кораблик}
    drawShip(x,y,kx,ky);
    Redraw;
    Sleep(1);
    x := x+h;
  end;
end.

```

Задача 2. Построить график функции  $f(x) = x \cdot \sin(x) \cdot \cos(x)$  на интервале  $[a, b]$ .

Решение:

```

uses GraphABC;
var x, h, MinF, MaxF, y, a, b: real;
    i, w, hp: integer;

function F(x: real):real;
BEGIN
    f:=x*sin(x)*cos(x);
END; {F}

function P_x(x: real; {переводимая координата}
    a,b:real; {границы исходного интервала}
    MaxX:integer ):integer; {ширина графического поля}
BEGIN
    P_x:=round(MaxX/(b-a)*(x-a));
END; {P_x}

{Для преобразования y в координату экрана }
function P_y(y: real;
    MinF,MaxF:real; {границы исходного интервала}
    MaxY:integer):integer; {высота графического поля}
BEGIN
    P_y:=round((MaxY-3)-(MaxY-3)/(MaxF-MinF)*(y-MinF));
END;
begin
    SetWindowSize(700,500); //создание графического окна
    w:=WindowWidth;
    hp:=WindowHeight;
    a:=-5; b:=5;
    x:=a; minF:=f(x); MaxF:=F(x); h:=(b-a)/500;
    for i:=1 to 500 do
        begin
            if maxF <= f(x) then maxF:=f(x);
            if minF >= f(x) then minF:=f(x);
            x:=x+h;
        end;
        {Вывод осей координат:
        ось OX (y=0, x=a..b) }
        Line( P_x(a,a,b,w), P_y(0,MinF,MaxF, hp) ,
            P_x(b,a,b,w) , P_y(0,MinF,MaxF, hp) );
        {ось OY (x=0, y=MinF..MaxF) }
        Line(P_x(0,a,b,w), P_y(MinF,MinF,MaxF, hp) ,
            P_x(0,a,b,w), P_y(MaxF,MinF,MaxF, hp) );
        SetPenColor(clBlue);
        SetPenWidth(3);
        {Построение графика функции}
        x:=a; y:=F(a); { начальная точка}
        {курсор в начальную точку на экране}
        MoveTo(P_x(x,a,b,w) , P_y(y,MinF,MaxF, hp));

```

```

For i:=1 to 500 do
begin
  {вычисляем X и значение функции в этой точке}
  x:=x+h;  y:=f(x);
  {Проводим линию от текущей позиции в точку с координатами:}
  LineTo (P_x(x,a,b,w), P_y(y,MinF,MaxF,hp))
end;
end.

```

### Устные опросы и коллоквиум

Устные опросы и коллоквиум осуществляется преподавателем по завершению изучения каждого раздела. Вопросы и задания приведены в приложении 2. Для подготовки используется основная и дополнительная литература по дисциплине «Информационные технологии в прикладной механике», а также информация, размещенная в LMS BlackBoard.

Вопросы, возникающие в процессе подготовки, студент может задать преподавателю либо на консультациях, либо через специальное средство LMS BlackBoard.

### Тестирование

Тестирование осуществляется на занятии через систему BlackBoard. Для подготовки тестов используются пробные тесты, размещенные в системе BlackBoard. Эти тесты не оцениваются преподавателем и служат элементом самоконтроля. Оба типа теста формируются на основе одной базы вопросов.

### Требования к представлению и оформлению результатов самостоятельной работы

Результатом самостоятельной работы студентов являются:

1. Программы для решения заданий расчетно-графической работы, реализованные в системе PascalABC и оформленные в соответствии с правилами оформления программ. Эти программы обязательно отсылаются преподавателю через систему BlackBoard. Защита расчетно-графического задания осуществляется только после того, как задача проверена и оценена преподавателем в системе BlackBoard.

2. Тесты самоконтроля, выполненные в системе BlackBoard. Все попытки. Выполненные пользователем сохраняются. К основному тесту студент допускается только после прохождения на зачетном уровне данного вида тестов.

### **Правила оформления программ**

#### **I. Блоки**

1.1. Каждый оператор языка программирования записывается на отдельной строке.

1.2. Содержимое конструкций `begin-end`, `repeat-until`, `case-end`, `record-end` выделяется отступом в виде некоторого количества дополнительных пробелов в начале строки или табуляции.

1.3. Содержимое конструкций `if-then`, `if-then-else`, `while-do`, `with-do` допускается писать в одну строку, если суммарная длина строки остается приемлемой. В случае, если содержимое перенести необходимо, то его можно вынести в отдельную строку, дополнив отступом.

1.4. Конструкцию `if-then-else` можно писать в две части, выделяя действие в случае невыполнения условия (`else`) на отдельную строку, но тогда `else` должен выделяться дополнительным отступом. Также по предыдущему пункту можно писать всю конструкцию в одну строку или в четыре, но никак иначе.

1.5. Если в любую конструкцию (кроме `if-then-else`) вложен `begin-end`, то `begin-end` можно писать на том же уровне вложенности, что и конструкцию, в которую он вложен. Более того, допускается не переносить `begin` на отдельную строку.

1.6. Содержимое секций `uses`, `type`, `const`, `var` должно выделяться отступом. Сами названия секций при этом должны идти без отступа в отдельной строке.

1.7. Выбранный вид отступа должен быть единым для всей программы: не допускается менять ширину отступа в пробелах в середине программы, а также менять табуляцию на пробелы и наоборот.

## II. Пробелы внутри строк

2.1. Любая бинарная операция (<, >, <=, >=, <>, =, +, -, \*, div, mod, /, and, or, xor, и т.д.) должна выделяться пробелом как слева, так и справа от себя. В том числе это относится и к знаку присваивания :=.

2.2. Однострочный комментарий может идти в той же строке, что и код, но при этом он обязан

отделяться от кода двумя пробелами. Во всех иных случаях комментарий не может идти в той же строке, что и код.

2.3. Перед символами «:», «,» и «;» никогда не должен идти пробел, и всегда должен идти после (если это не конец строки).

2.4. Запрещается использовать в строке более одного пробела подряд, если эти пробелы не составляют отступ и не служат для отделения комментария от кода.

2.5. Ключевые слова array, case, do, downto, else, file, if, in, of, then, to, while, with, until всегда должны содержать по обе стороны от себя хотя бы один пробельный символ (допускается перевод строки).

2.6. Запрещено отделять знак «..» (две точки) от соседних лексем слева и справа при объявлении диапазонов.

2.7. При обращении к элементу массива пробелы между квадратными скобками и их содержимым не ставятся. Также не ставится пробел между именем переменной и открывающейся квадратной скобкой.

## III. Идентификаторы

3.1. В программе запрещается использование кириллицы (даже в комментариях).

3.2. В программе запрещается использование транслита.

3.3. В качестве имен простых переменных/констант, смысл которых достаточно очевиден, разрешается использовать короткие название из одной-двух букв. При этом, по сложившейся традиции, имена счетчиков в циклах обычно называют i и j. Для нетривиальных понятий лучше использовать английские слова.

3.4. Не стоит злоупотреблять длинными именами переменных и констант. 10-15 символов должно быть достаточно для передачи смысла переменной.

3.5. Имена переменных и констант должны содержать только строчные латинские буквы, цифры и нижние подчеркивания для разделения слов в названии.

3.6. Имена функций и процедур именуются с использованием CamelCase, то есть содержат только латинские буквы и цифры, каждое слово, кроме первого, пишется с заглавной буквы.

3.7. Все ключевые слова языка должны записываться только строчными латинскими буквами.

4.2. Каждая процедура и функция должна содержать перед и после себя ровно одну пустую строку.

4.3. В случае, если в функцию или процедуру передается переменная сложного типа (массив, множество и т.п.), она обязательно должна выделяться одним из модификаторов `const` или `var`.

4.4. Между списком аргументов и именем функции/процедуры никогда не ставится пробел.

## V. Другие правила

5.1. Запрещается использование конструкции `goto`.

5.2. В программе запрещается использование нетривиальных констант без их явного описания в разделе `const`. Обычно нетривиальными считаются любые константы, кроме, быть может, `-1`, `0`, `1`, `2`, `10`.

5.3. Каждая переменная перед использованием обязана быть явно инициализирована присваиванием.

5.4. Запрещается сравнивать переменные типа `boolean` с `true` или `false`.

5.5. Внутри цикла `for` запрещается явное изменение значения счетчика цикла.

5.6. Запрещено использовать строки длиннее 120 символов. Строки можно разбивать на части в длинных выражениях и списках аргументов, при

этом если выражение разбивается на бинарной операции, то саму операцию лучше переносить на отдельную строку. Строки 2, 3, 4, ... одного оператора считается вложенными в первую строку и выделяются отступом.

### **Критерии оценки выполнения самостоятельной работы**

Самостоятельная работа студентов включает расчетно-графические задания, подготовку к устным опросам, тестирование. Критерии оценки каждого вида работы приведены в приложении 2.





МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
**«Дальневосточный федеральный университет»**  
(ДФУ)

---

**ИНЖЕНЕРНАЯ ШКОЛА**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**по дисциплине «Информационные технологии в прикладной механике»**  
**Направление подготовки: 15.03.03 Прикладная механика**  
**Профиль подготовки: «Математическое и компьютерное моделирование**  
**механических систем и процессов»**  
**Форма подготовки очная**

**Владивосток**  
**2016**

## Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
ОПК-7 – умением использовать современные программные средства подготовки конструкторско-технологической документации	Знает	методы описания графической информации и реализации алгоритмов компьютерной графики
	Умеет	программно реализовывать основные алгоритмы растровой графики
	Владеет	системами разработки программного обеспечения и графическими редакторами.
ОПК-10 – способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Знает	алгоритмы поиска, сортировки и представления данных
	Умеет	использовать технологическую и справочную документацию при разработке программ
	Владеет	способностью использовать современные технические средства и информационные технологии для решения аналитических, исследовательских и коммуникативных задач;
ПК-2 – способностью применять физико-математический аппарат, теоретические, расчетные и экспериментальные методы исследований, методы математического и компьютерного моделирования в процессе профессиональной деятельности	Знает	основы структурного программирования
	Умеет	разрабатывать алгоритмы и программы для решения задач профессиональной деятельности средствами языков программирования
	Владеет	методологией проектирования и реализации программных продуктов в современных системах программирования

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций		Оценочные средства	
				текущий контроль	промежуточная аттестация
1	Основы структурного программирования	ОПК-10	Знает алгоритмы поиска, сортировки и представления данных	ПР-1	Вопросы 1-8
			Умеет использовать технологическую и справочную документацию при разработке программ	ПР-12 ПР-6	Задачи 1-7

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства		
			текущий контроль	промежуточная аттестация	
			Владеет способностью использовать современные технические средства и информационные технологии для решения аналитических, исследовательских и коммуникативных задач;	ПР-12	Задачи 18-25
2	Структурированные типы данных	ПК-2	Знает основы структурного программирования	УО-1	Вопросы 18-25
			Умеет разрабатывать алгоритмы и программы для решения задач профессиональной деятельности средствами языков программирования	ПР-12 ПР-6	Задачи 7-14
			Владеет методологией проектирования и реализации программных продуктов в современных системах программирования	ПР-12	Задачи 26-30
3	Графика и визуализация	ОПК-7	Знает методы описания графической информации и реализации алгоритмов компьютерной графики	УО-2	Вопросы 27-33
			Умеет программно реализовывать основные алгоритмы растровой графики	ПР-6	Задачи 15-17
			Владеет системами разработки программного обеспечения и графическими редакторами.	ПР-12	Задачи 31-33

### Шкала оценивания уровня сформированности компетенций

Код и формулировка компетенции	Этапы формирования компетенции		критерии	показатели
ОПК-7 – умение использовать современные программные средства подготовки конструкторско-технологической документации	знает	Методы описания графической информации и реализации алгоритмов компьютерной графики	Воспроизводит и объясняет с требуемой степенью научной точности и полноты методы описания графической информации и реализации алгоритмов компьютерной графики	1. Дает определение основных понятий, используемых в компьютерной графике 2. Знает способы изображения и трансформации растровых изображений 3. Знает возможности графических редакторов и других систем,

Код и формулир. компетенции	Этапы формирования компетенции		критерии	показатели
				используемых для визуализации данных
	умеет	программно реализовывать основные алгоритмы растровой графики	Реализует стандартные алгоритмы растровой графики.	1. Разрабатывает программы на основе стандартных графических алгоритмов 2. Использует пакеты графики и визуализации
	владеет	системами разработки программного обеспечения и графических редакторов.	Успешно и систематически применяет системы разработки программного обеспечения и графические редакторы, в том числе и в нестандартных ситуациях.	1. Проектирует и реализует программы для визуализации изображений и данных различной природы 2. Свободно владеет пакетами графики и визуализации
ОПК-10 – способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	знает	алгоритмы поиска, сортировки и представления данных	Воспроизводит и объясняет с требуемой степенью научной точности и полноты алгоритмы сортировки и поиска данных, использует типы различные данных для описания информации	1. Формально описывает и объясняет основные алгоритмы поиска и сортировки 2. Дает определения различным типам данных, знает способы описания информации различными типами данных
	умеет	использовать технологическую и справочную документацию при разработке программ	Грамотно использует технологическую и справочную информацию при разработке и реализации программ	1. Использует различные типы справочной и технологической документации при разработке программ
	владеет	способностью использовать современные технические средства и информационные технологии для решения аналитических, исследовательских и коммуникативных задач;	Успешно систематически применяет современные технические средства и информационные технологии для решения аналитических, исследовательских и коммуникативных задач, в том числе и в нестандартных ситуациях	1. Проектирует и реализует нестандартные задачи в области профессиональной деятельности 2. Работает в проектной команде по разработке программных продуктов
ПК-2 – способностью применять физико-математический аппарат, теоретические, расчетные и экспериментальные методы исследований, методы математического и компьютерного моделирования в	знает	основы структурного программирования	Воспроизводит и объясняет с требуемой степенью научной точности и полноты основы структурного и модульного программирования	1. Дает определения основных понятий модульного и структурного программирования 2. Знает основные способы представления реальных данных с помощью типов данных языков высокого уровня 3. Знает приемы разработки программ
	умеет	разрабатывать алгоритмы и программы для	Разрабатывает алгоритмы и программы для	1. Разрабатывает собственные программные продукты для решения

Код и формулир. компетенции	Этапы формирования компетенции		критерии	показатели
процессе профессиональной деятельности		решения задач профессиональной деятельности средствами языков программирования	решения задач профессиональной деятельности средствами языков программирования, используя стандартные методы и эталонные образцы	стандартных задач в области профессиональной деятельности
	владеет	методологией проектирования и реализации программных продуктов в современных системах программирования	Успешно и систематически применяет современные системы программирования в процессе разработки программного продукта	1. Современными системами программирования как средством разработки программ 2. Методологией разработки программных систем

### Зачетно-экзаменационные материалы

#### Перечень типовых экзаменационных вопросов

1. Основные парадигмы программирования.
2. Классификация языков программирования.
3. Понятие алгоритма и программы.
4. Структура программы на языке Паскаль.
5. Целые и вещественные типы данных. Операции над ними. Правила преобразования и совместимости типов.
6. Операторы присваивания, ввода и вывода.
7. Логический тип данных.
8. Условный оператор.
9. Операторы повторения в языке Паскаль.
10. Цикл с параметрами.
11. Цикл с предусловием.
12. Цикл с постусловием.
13. Разбиение числа на цифры.
14. Алгоритм Эвклида нахождения наибольшего общего делителя двух чисел.
15. Алгоритмы вычисления сумм и произведений.
16. Алгоритмы статистической обработки данных.

17. Вычисление суммы ряда для заданного числа членов и с заданной точностью.

18. Структурированные типы данных: одномерные массивы. Операции ввода и вывода. Заполнения элементов массива случайным образом.

19. Операции над массивами: перебор, перестановка элементов, копирование массива целиком.

20. Операции над массивами: удаление и вставка элементов массива.

21. Поиск первого встретившегося и последнего встретившегося элемента в неупорядоченном массиве.

22. Сортировка пузырьком.

23. Сортировка обменом.

24. Сортировка вставками.

25. Поиск элемента в упорядоченном массиве (двоичный поиск).

26. Типы графических изображений

27. Графические возможности языка Паскаль.

28. Основные понятия графического режима: перо, холст, канвас.

29. Процедуры реализации графических примитивов

30. Алгоритм построения графика функции

31. Относительные и абсолютные координаты для рисования изображения

32. Преобразование изображений: масштабирование и поворот.

33. Программная реализация движения.

### **Перечень типовых экзаменационных задач**

1. Дано натуральное число. Составить алгоритм, который позволяет получить самую большую цифру этого числа.

2. Составить алгоритм решения системы двух уравнений.

3. Составить алгоритм вычисления суммы ряда:

$$(1+x)^p = 1 + \frac{p}{1!}x + \frac{p(p-1)}{2!}x^2 + \dots + \frac{p(p-1)(p-2)\dots(p-n+1)}{n!}x^n + \dots$$

4. Составить алгоритм решения уравнения  $\mathbf{a} \cdot |\mathbf{x}| = \mathbf{b}$ , где  $\mathbf{a}$  и  $\mathbf{b}$  - заданные действительные числа.

5. Составить алгоритм вычисления суммы ряда:

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + \frac{(-1)^{n-1} x^n}{n} + \dots$$

6. Дано натуральное число. Составить алгоритм формирования числа, полученного выписыванием в обратном порядке цифр исходного числа.

7. Составить алгоритм нахождения наибольшего общего делителя двух чисел

8. Составить алгоритм поиска максимального числа среди отрицательных чисел некоторой последовательности

9. Написать алгоритм, который в последовательности  $a_1, a_2, \dots, a_n$  меняет местами наибольший и наименьший элементы.

10. Дан массив из  $n$  элементов. Написать алгоритм удаления из него всех элементов, меньших среднеарифметического значения этой последовательности.

11. Даны две последовательности по  $n$  целых чисел в каждой. Найти наименьшее среди тех чисел первой последовательности, которые не входят во вторую.

12. Даны две последовательности чисел. Написать алгоритм построения новой последовательности, являющейся объединением двух данных (в нее входят те элементы, которые принадлежат обоим последовательностям).

13. Написать алгоритм, который переставляет в обратном порядке элементы, расположенные между минимальным и максимальным элементами массива.

14. Вставить по одному элементу перед всеми отрицательными числами. Составить алгоритм построения графической области, ограниченной неравенствами:

$$\begin{cases} y \geq x^2 \\ y \leq 2 - x \end{cases}$$

15. Составить алгоритм построения графиков, заданных в параметрическом виде. Построить график следующей функции:

$$\begin{cases} x(t) = \sin^3(t) \\ y(t) = \cos^3(t) \end{cases}$$

16. Составить алгоритм построения графиков, заданных в полярной системе координат. Построить график следующей функции:

$$r(t) = 1 + 3 \cdot t$$

17. Дана равнобочная трапеция, заданная длинами оснований и высотой. Написать программу, которая вычисляет площадь и периметр этой трапеции.

18. Составить программу для решения уравнения  $x^2 + b \cdot x + c = 0$ , где  $c$  и  $b$  - заданные действительные числа. Определить сколько корней имеет уравнение.

19. Даны координаты центров двух окружностей  $(x_1, y_1)$  и  $(x_2, y_2)$ , а также их радиусы  $r_1$  и  $r_2$  ( $r_1 > r_2$ ). Написать программу, которая определяет взаимное расположение этих окружностей.

20. Даны координаты трех вершин треугольника найти. Написать программу, которая определяет тип этого треугольника (прямоугольный, остроугольный или тупоугольный).

21. Написать программу, которая вводит с клавиатуры последовательность из  $N$  целых чисел и найти сумму положительных и произведение отрицательных чисел этой последовательности.

22. Написать программу, которая получает все числа Армстронга, состоящие из трех и четырех цифр. Натуральное число из  $n$  цифр является числом Армстронга, если сумма его цифр, возведенных в  $n$ -ю степень, равна самому числу (например  $153 = 1^3 + 5^3 + 3^3$ ).

23. Написать программу для решения уравнения  $n^2 + m^2 = k^2$  в целых числах на интервале  $[1..p]$ .

24. Написать программу для вычисления следующего выражения:

$$S = \sum_{k=1}^{10} \frac{\sum_{i=1}^k \cos(ki)}{k!}$$

25. Даны вещественные числа  $a_0, a_1, \dots, a_{15}$ . Найти коэффициенты многочлена  $(x-a_0)(x-a_1) \dots (x-a_{15})$ .



26. Даны целые числа  $a_1, a_2, \dots, a_n$ . Преобразовать последовательность, выбросив из нее все члены, равные  $\max(a_1, a_2, \dots, a_n)$ .

27. В последовательности  $a_1, a_2, \dots, a_n$  поменять местами наибольший и наименьший элементы.

28. Удалить все повторяющиеся элементы массива, то есть получить массив, состоящий из различных элементов.

29. Нарисовать произвольную картинку, масштабируемую по вертикали и горизонтали и реализовать ее движение по параболе, так чтобы картинка не выходила за границы графического окна.

30. Нарисовать произвольную картинку, масштабируемую по вертикали и горизонтали и реализовать ее движение по прямой, заданной уравнением  $y=kx+b$ , так чтобы картинка не выходила за границы графического окна.

31. Нарисовать произвольную картинку, масштабируемую по вертикали и горизонтали и реализовать ее движение по периметру окна.

### Образец экзаменационного билета

1. Классификация языков программирования.

2. Операции над массивами: перебор, перестановка элементов, копирование массива целиком

3. Составить алгоритм вычисления суммы ряда:

$$(1+x)^p = 1 + \frac{p}{1!}x + \frac{p(p-1)}{2!}x^2 + \dots + \frac{p(p-1)(p-2)\dots(p-n+1)}{n!}x^n + \dots$$

4. Даны координаты центров двух окружностей  $(x_1, y_1)$  и  $(x_2, y_2)$ , а также их радиусы  $r_1$  и  $r_2$  ( $r_1 > r_2$ ). Написать программу, которая определяет взаимное расположение этих окружностей.

### Принцип составления экзаменационного билета

Первые два вопроса являются теоретическими и предназначены для оценивания порогового уровня освоения дисциплины. Третий вопрос, на составление алгоритма, предназначен для оценки продвинутого уровня.

Последний вопрос – написание программы на компьютере – для высокого уровня освоения.

Таблица для составления экзаменационных билетов по фонду оценочных средств:

Номер вопроса в билете	Номера вопросов и задач из перечня
1	вопросы 1 – 9
2	вопросы 10 – 27
3	задачи 1 – 17
4	задачи 18 – 31

### Образец экзаменационного билета

1. Классификация языков программирования.

2. Операции над массивами: перебор, перестановка элементов, копирование массива целиком

3. Составить алгоритм вычисления суммы ряда:

$$(1+x)^p = 1 + \frac{p}{1!}x + \frac{p(p-1)}{2!}x^2 + \dots + \frac{p(p-1)(p-2)\dots(p-n+1)}{n!}x^n + \dots$$

4. Даны координаты центров двух окружностей  $(x_1, y_1)$  и  $(x_2, y_2)$ , а также их радиусы  $r_1$  и  $r_2$  ( $r_1 > r_2$ ). Написать программу, которая определяет взаимное расположение этих окружностей.

### Критерии выставления оценки студенту на экзамене по дисциплине

#### «Информационные технологии в прикладной механике»

Баллы (рейтинговой оценки)	Оценка экзамена (стандартная)	Требования к сформированным компетенциям
86-100	«отлично»	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал по основам программирования, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет составлять алгоритм и программу, а затем выполнять ее документирование, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, свободно использует компьютер для сбора и анализа данных, выбирает эффективный алгоритм обработки информации, правильно обосновывает принятое решение,

		владеет разносторонними навыками и приемами выполнения практических задач, связанных с проектированием и реализацией программ в области профессиональной деятельности.
76-85	<i>«хорошо»</i>	Оценка «хорошо» выставляется студенту, если он твердо знает материал по основам программирования, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, связанных с проектированием и реализацией программ в области профессиональной деятельности, владеет необходимыми навыками и приемами их выполнения с использованием информационно-коммуникационных средств.
61-75	<i>«удовлетворительно»</i>	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала в области программирования, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, связанным с написанием программ и применением стандартных пакетов в области своей профессиональной деятельности.
0-60	<i>«неудовлетворительно»</i>	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала по основам программирования, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, связанные с написанием программ и алгоритмов. Оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине

## .Оценочные средства для текущей аттестации

### Вопросы для собеседований и коллоквиума по дисциплине «Информационные технологии в прикладной механике»

#### Раздел «Структурированные типы данных»

1. Дать определение массива.
2. Чем отличаются структурированные типы данных от простых?
3. Какие типы данных не допустимы для компонентов массива?

Почему?

4. В каких разделах программы допустимо описание массива?
5. Как определяется общее число элементов массива?
6. Можно ли использовать при описании массива тип диапазон?
7. Дайте определение индекса. Какие ограничения накладываются при описании границ изменения индекса?
8. Что называется базовым типом?
9. Приведите описание двух способов объявления массива.
10. Можно ли в качестве границ индексов массива использовать переменные? Объяснить почему.
11. Как осуществляется доступ к каждому элементу массива?
12. Объявить массив, элементами которого являются вещественные числа, размером 20. Для этого массива реализовать:
  - ввод, заполнение случайным образом (числами в интервале от 5.5 до 6.7), вывод массива;
  - поиск максимального элемента в массиве
  - вставку 3-го и удаление 5-го элемента
13. Как осуществляется доступ к каждому элементу массива?
14. Какие математические объекты можно описать средствами массивов?
15. Объявить вектор, размерностью  $n$ , матрицу, размерностью  $n$  на  $n$ .  
Для них реализовать:

- заполнение случайным образом числами в интервале от 5.5 до 6.7;
- сложение двух векторов, сложение двух матриц, умножение матрицы на вектор, умножение двух матриц.

16. Реализовать алгоритм двоичного поиска

17. Реализовать алгоритм поиска со сторожем

18. Реализовать алгоритм сортировки массива «пузырьком»

19. Реализовать алгоритм сортировки массива перестановками

20. Реализовать алгоритм сортировки массива вставками

21. Реализовать алгоритм сортировки упорядоченных массивов слиянием.

### **Раздел «Графика и визуализация»**

1. Какие графические примитивы содержит модуль GraphABC?

2. Какими процедурами задают цвет и толщину пера?

3. Какими процедурами задают цвет кисти?

4. Какие процедуры используют при выводе и форматировании текста

в графическом окне

5. Какие размеры имеет экран в графическом режиме Паскаль?

6. Где находится точка начала координат?

7. Как изменяются координаты x и y в Паскале?

8. Какого типа должны быть координаты объектов?

9. Какие команды рисования объектов вы знаете?

10. Как установить графический режим?

11. Двигается ли что-то физически на экране при изображении движения?

12. Как организовать изменение координат при движении по вертикали?

13. Написать программу, которая демонстрирует движение прямоугольника по диагоналям: из левого верхнего угла – в правый нижний, а затем из правого верхнего угла – в левый нижний.

14. Написать программу, которая спрашивает у пользователя какую фигуру нужно нарисовать: 1 – эллипс, 2 – круг, 3 – дугу и какого цвета; и выводит на экран соответствующую фигуру

15. Изобразить на экране прямоугольник, вращающийся в плоскости экрана вокруг своей середины.

16. Составить программу, изображающую бегущую строку с каким-нибудь текстом.

17. Составить программу, которая демонстрирует движение увеличивающегося прямоугольника по диагонали.

18. Написать программу, которая спрашивает пользователя, как должен вести себя шар: 1 центра до границ экрана или 2 экрана к центру и его цвет; и показывать соответствующее изменение шарика.

19. Изобразить на экране отрезок, вращающийся в плоскости экрана вокруг точки, делящей отрезок в отношении 1:3.

20. Составить программу, изображающую бегущую строку (снизу вверх) с каким

21. Составить программу, которая показывает упругий удар двух шаров, которые движутся навстречу друг другу, сталкиваются и продолжают движение в разные стороны. Шары должны быть разного цвета.

22. Написать программу, которая спрашивает у пользователя какую фигуру нужно нарисовать: 1 – прямоугольник, 2 – круг, 3 – линию и какого цвета; и выводит на экран соответствующую фигуру.

23. Изобразить на экране отрезок, вращающийся в плоскости экрана вокруг своей середины.

24. Составить программу, изображающую бегущую строку (справа налево) с каким-нибудь текстом.

#### **Критерии оценки устного ответа на коллоквиуме:**

✓ 100-85 баллов выставляется студенту, если его ответ показывает прочные знания основных положений изучаемого раздела

программирования, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять синтаксис и семантику основных конструкций программирования и структур данных, делать выводы и обобщения, давать аргументированные ответы, приводить примеры алгоритмов; свободное владение монологической речью, логичность и последовательность ответа; умение алгоритмически описывать проблему из выбранной предметной области.

✓ 85-76 баллов выставляется студенту, если его ответ, обнаруживающий прочные знания основных положений изучаемого раздела программирования, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять синтаксис и семантику основных конструкций программирования и структур данных, делать выводы и обобщения, давать аргументированные ответы, приводить примеры алгоритмов; свободное владение монологической речью, логичность и последовательность ответа; умение алгоритмически описывать проблему из выбранной предметной области. Однако допускается одна - две неточности в ответе.

✓ 75-61 балл выставляется студенту, если его ответ, свидетельствующий в основном о знании основных положений изучаемого раздела программирования, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками объяснения синтаксис и семантику основных конструкций программирования и структур данных, недостаточным умением давать аргументированные ответы и приводить примеры алгоритмов; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение алгоритмически описывать проблему из выбранной предметной области.

✓ 60-50 баллов выставляется студенту, если его ответ, обнаруживающий незнание процессов основных положений изучаемого

раздела программирования, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа синтаксиса и семантики основных конструкций программирования и структур данных; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; неумение алгоритмически описывать проблему из выбранной предметной области

**Тестовые задания по дисциплине «Информационные технологии в  
прикладной механике»**

Каждому студенту формируется индивидуальный тест, в который входит 20 вопросов, выбранных случайным образом из списка заданий.

**Список типовых заданий теста по разделу «Основы структурного  
программирования»**

1. Какие из перечисленных идентификаторов являются допустимыми в языке Pascal?

- a) Count\_Num
- b) Num\_1
- c) F\_and\_S
- d) 1\_Num
- e) First N

2. Какие числовые значения допустимы в языке Паскаль?

- a) 456734.45+
- b) -17890999+
- c) E+36+
- d) 32.0 E-10+
- e) 8\*\*2
- f) 34,2345



3. Переменная P в результате выполнения команд

P:=1;

FOR k:=1 TO 4 DO

P:=P\*k;

получит значение...

4. Установите в правильной последовательности команды, позволяющие вычислить факториал введенного натурального числа N

a) Readln(n);

b) fn:=1;

c) FOR i:=1 TO n TO

d) fn := fn \* i;

e) Writeln(N, '!=', FN);

#### **Критерии оценки теста, состоящего из 20 вопросов**

✓ 15-20 баллов – считается, что тест пройден.

✓ 0-14 баллов – тест не засчитывается

**Комплекты заданий для выполнения расчетно-графических работ по дисциплине «Информационные технологии в прикладной механике»**

**Расчетно-графическая работа по разделу «Основы структурного программирования», тема «Выражение и ветвление» базовый уровень**

Расчетно-графическая работа состоит из трех заданий. В каждом задании требуется реализовать программу на языке Паскаль для некоторой задачи. При этом необходимо использовать следующие операторы: ввод, вывод, присваивание и условный.

Первое задание – вычислительная геометрическая задача.

Второе задание – задача на разбиение числа на цифры и выполнения некоторых операций с полученными цифрами

Третье задание – текстовая задача.

#### **Типовое задание**

1. Найти уравнение прямой, проходящей через две различные точки, заданные своими координатами.
2. Определить, равна ли сумма двух первых цифр заданного четырехзначного натурального числа сумме двух его последних цифр.
3. На поле  $(k, p)$  шахматной доски расположен ферзь. Угрожает ли он полю  $(m, n)$ ?

**Расчетно-графическая работа по разделу «Основы структурного программирования», тема «Операторы повторения», продвинутый уровень**

Расчетно-графическая работа состоит из трех заданий. В каждом задании требуется реализовать программу на языке Паскаль для некоторой задачи. При этом необходимо использовать следующие операторы: ввод, вывод, присваивание, условный, циклы.

Первое задание – переборная задача для последовательности вводимых чисел (поиск максимума, суммирование и пр.).

Второе задание – вычисление суммы ряда либо для заданного числа членов, либо с заданной точностью.

Третье задание – вычислительная задача, в которой используются простые или вложенные циклы.

**Типовое задание**

1. Ввести с клавиатуры последовательность из  $n$  целых чисел и найти максимальное число среди элементов, стоящих на четных местах.
2. Вычислить сумму ряда для заданного числа членов  $n$  ( $n$  и  $x$  вводятся с клавиатуры):

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$$

3. Определить, является ли натуральное число  $k$  степенью 3.

### **Расчетно-графическая работа по разделу «Структурированные типы данных», тема «Одномерные массивы»**

Расчетно-графическая работа состоит из трех заданий. В каждом задании требуется реализовать программу на языке Паскаль для некоторой задачи, в которой обрабатывается одномерный массив.

Первое задание – переборная задача для одномерного массива (поиск максимума, суммирование и пр.).

Второе задание – удаление или вставка элементов массива.

Третье задание – вычислительная задача, в которой используются одномерные массивы.

#### **Типовое задание**

1. Дана последовательность из  $n$  целых чисел. Найти количество чисел этой последовательности, расположенных между первым максимальным и последним минимальным элементами.
2. Дана последовательность из  $n$  целых чисел. Если сумма ее элементов отрицательна, то вставить перед первым элементов новый, равный по модулю этой сумме.
3. Даны две последовательности по  $n$  целых чисел в каждой. Найти наименьшее среди тех чисел первой последовательности, которые не входят во вторую.

### **Расчетно-графическая работа по разделу «Графика и визуализация»**

Расчетно-графическая работа состоит из двух заданий. В каждом задании требуется реализовать программу на языке Паскаль для некоторой задачи, в которой используются графический режим работы.

Первое задание – масштабирование и движение изображения.

Второе задание – визуализация графика функции или области.

#### **Типовое задание**

1. Нарисовать произвольную картинку, масштабируемую по вертикали и горизонтали и реализовать ее движение по горизонтали с ускорением, так чтобы картинка не выходила за границы окна.
2. Реализовать построение графиков, заданных в полярной системе координат. Построить график следующей функции:

$$r(t) = 1 + 3 \cdot t$$

### **Критерии оценки расчетно-графической работы**

✓ 10-8 баллов выставляется студенту, если студент выполнил все пункты расчетно-графического задания. Фактических ошибок, связанных с пониманием проблемы, нет; семантических и синтаксических ошибок в программах нет. При защите студент отвечает на все вопросы преподавателя.

✓ 7-6 баллов – работа выполнена полностью; одна задача реализована не для всех исходных данных или одна программа оформлена не в соответствии со стандартами. При защите студент отвечает на все вопросы преподавателя.

✓ 5-4 балла – работа выполнена полностью. Две задачи реализованы не для всех входных данных или две программы оформлены не в соответствии со стандартами. При защите студент не отвечает на 1-2 вопроса преподавателя.

✓ 1-3 балла – работа выполнена не полностью. Три задачи реализованы не для всех входных данных или три программы оформлены не в соответствии со стандартами. При защите студент не отвечает на более, чем на 2 вопроса преподавателя.