

федеральное государственное бюджетное образовательное учреждение  
высшего образования  
РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ  
УНИВЕРСИТЕТ

Кафедра Информационных технологий и систем безопасности

Рабочая программа по дисциплине

**ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ**

Основная профессиональная образовательная программа  
высшего образования программы специалитета по специальности

**10.05.02 «Информационная безопасность телекоммуникационных систем»**

Специализация:

**Разработка защищенных телекоммуникационных систем**

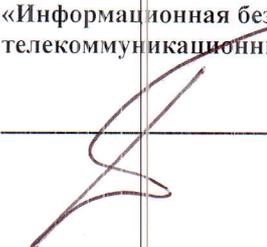
Квалификация:

**Специалист**

Форма обучения

**Очная**

Согласовано  
Руководитель ОПОП  
«Информационная безопасность  
телекоммуникационных систем»

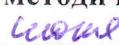
  
Бурлов В.Г.

Утверждаю

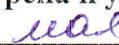
Председатель УМС  И.И. Палкин

Рекомендована решением

Учебно-методического совета

  2018 г., протокол № 4

Рассмотрена и утверждена на заседании кафедры

  2018 г., протокол № 5

Зав. кафедрой  Бурлов В.Г.

Авторы-разработчики:

 Шишкин А.Д.

## **Раздел I Информатика**

### **Цели:**

- формирование общей информационной культуры студентов, подготовка их к деятельности, связанной с использованием современных информационных технологий;
- подготовка специалистов к деятельности, связанной с разработкой программного обеспечения для решения профессиональных задач;
- формирование теоретических знаний, освоение методов и средств программирования практических инженерных задач на современных пакетах объектно-ориентированного языка, подготовка специалистов к деятельности, связанной с разработкой и эффективной реализацией программных средств для решения профессиональных задач

### **Основные задачи дисциплины:**

- изучение основных понятий информатики;
- изучение свойств и способов записи алгоритмов;
- изучение способов представления чисел, символов, графики, аудио- и видеоинформации в персональном компьютере;
- ознакомление с логическими основами устройства ЭВМ;
- ознакомление с составом и назначением функциональных узлов компьютера;
- изучение основ построения операционных систем (ОС) на примере ОС с открытым кодом;
- изучение основ программирования в командных оболочках;
- овладение навыками применения сервисных программных средств системного и прикладного назначения;
- изучение основ построения компьютерных сетей;
- овладение навыками поиска информации в глобальной информационной сети Интернет;
- ознакомление с теоретическими основами программирования;
- изучение основ алгоритмизации;
- изучение средств описания данных и средств описания действий языков программирования;
- овладение навыками и приемами программирования;
- освоение современных сред создания программных продуктов
- изучение базовых структур данных;
- изучение основных алгоритмов сортировки и поиска;
- изучение алгоритмов генерации перестановок, подмножеств;
- изучение основных алгоритмов на графах и паттернов программирования;
- освоение основных методов оценки вычислительной сложности алгоритмов.
- обучить технологии составления проектно-технической документации на разрабатываемый программный продукт;

### **В результате освоения дисциплин студент должен:**

#### Знать:

- основные понятия информатики;
- формы и способы представления данных в персональном компьютере;

- состав, назначение функциональных компонентов и программного обеспечения персонального компьютера;
- классификацию современных компьютерных систем;
- типовые структуры и принципы организации компьютерных сетей;
- основы и стандарты разработки алгоритмов вычислительных процессов;
- методы составления алгоритмов;
- методы формализации алгоритмов;
- основные приемы реализации алгоритмов на языках структурного программирования;
- основные приемы программирования, области и особенности применения языков объектно-ориентированного программирования (ООП).
- паттерны проектирования;
- алгоритмы сортировки и поиска данных;
- методы тестирования и отладки ПО.

Уметь:

- применять типовые программные средства сервисного назначения (средства восстановления системы после сбоев, дефрагментации и очистки диска);
- пользоваться сетевыми средствами для обмена данными, в том числе с использованием глобальной информационной сети Интернет;
- работать с интегрированной средой разработки программного обеспечения;
- реализовывать на языке высокого уровня алгоритмы решения профессиональных задач, в том числе задач обработки потоков;
- работать с интегрированной средой разработки программного обеспечения;
- реализовывать на языке высокого уровня алгоритмы решения профессиональных задач, в том числе задач обработки потоков

Владеть:

- навыками работы с офисными приложениями (текстовыми процессорами, электронными таблицами, средствами подготовки презентационных материалов);
- навыками обеспечения безопасности информации с помощью типовых программных средств (антивирусов, архиваторов, стандартных сетевых средств обмена информацией).
- навыками разработки, документирования, тестирования и отладки программ;

## Раздел II Языки программирования

### 1. Цели освоения дисциплины

**Целью** изучения дисциплины «Языки программирования» является подготовка специалистов к деятельности, связанной с разработкой программного обеспечения для решения профессиональных задач.

**Задачи** изучения дисциплины:

- ознакомление с теоретическими основами программирования;
- изучение основ алгоритмизации;
- изучение средств описания данных и средств описания действий языков программирования;
- овладение навыками и приемами программирования;
- освоение современных сред создания программных продуктов.

## 2. Место дисциплины в структуре ОП

Дисциплина читается студентам 2-го курса в рамках дисциплины «Организационное и правовое обеспечение информационной безопасности».

Дисциплина «Языки программирования» относится к числу дисциплин базовой части профессионального цикла. Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе изучения дисциплин:

- «Алгебра и геометрия» – владеть методами линейной алгебры;
- «Информатика» – уметь пользоваться сетевыми средствами для обмена данными, в том числе с использованием глобальной информационной сети Интернет, владеть навыками работы с офисными приложениями;
- «Языки программирования» - знать язык программирования высокого уровня (объектно-ориентированное программирование); уметь работать с интегрированной средой разработки программного обеспечения; владеть навыками разработки, документирования, тестирования и отладки программ;
- «Английский язык» – владеть в объеме, необходимом для работы с языками программирования.

Дисциплина «Языки программирования» является предшествующей для изучения следующих базовых дисциплин: «Информационные технологии», «Моделирование систем и сетей телекоммуникаций». Знания и практические навыки, полученные в результате освоения дисциплины «Языки программирования», используются студентами при разработке курсовых и дипломных работ, в научно-исследовательской работе.

### Компетенции обучающегося, формируемые в результате освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Код компетенции	Компетенция
ОПК-4	способностью понимать значение информации в развитии современного общества, применять достижения информационных технологий для поиска и обработки информации
ОПК-5	способностью применять программные средства системного и прикладного назначения, языки, методы и инструментальные средства программирования для решения профессиональных задач

### 3. Требования к результатам освоения дисциплины

В результате изучения дисциплины студент должен

Код компетенции	Результаты обучения
ОПК-4	<p>Знать:</p> <ul style="list-style-type: none"><li>– основы и стандарты разработки алгоритмов вычислительных процессов;</li><li>– методы составления алгоритмов;</li></ul> <p>Уметь:</p> <ul style="list-style-type: none"><li>– работать с интегрированной средой разработки программного</li></ul>

	<p>обеспечения;</p> <p>Владеть:</p> <ul style="list-style-type: none"> <li>– навыками разработки и документирования программ.</li> </ul>
ОПК-5	<p>Знать:</p> <ul style="list-style-type: none"> <li>– методы формализации алгоритмов;</li> <li>– основные приемы реализации алгоритмов на языках структурного программирования;</li> <li>– основные приемы программирования, области и особенности применения языков объектно-ориентированного программирования (ООП).</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>– реализовывать на языке высокого уровня алгоритмы решения профессиональных задач, в том числе задач обработки потоков;</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>– навыками тестирования и отладки программ.</li> </ul>

Основные признаки проявленности формируемых компетенций в результате освоения дисциплины «Языки программирования» сведены в таблице.

**Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания**

Этап (уровень) освоения компетенции	Основные признаки проявленности компетенции (дескрипторное описание уровня)				
	1.	2.	3.	4.	5.
минимальный	не владеет	слабо ориентируется в терминологии и содержании	Способен выделить основные идеи текста, работает с критической литературой	Владеет основными навыками работы с источниками и критической литературой	Способен дать собственную критическую оценку изучаемого материала
	не умеет	не выделяет основные идеи	Способен показать основную идею в развитии	Способен представить ключевую проблему в ее связи с другими процессами	Может соотнести основные идеи с современными проблемами
	не знает	допускает грубые ошибки	Знает основные рабочие категории, однако не ориентируется в их специфике	Понимает специфику основных рабочих категорий	Способен выделить характерный авторский подход
базовый	не владеет	плохо ориентируется в терминологии и содержании	Владеет приемами поиска и систематизации, но не способен свободно изложить материал	Свободно излагает материал, однако не демонстрирует навыков сравнения основных идей и концепций	Способен сравнивать концепции, аргументированно излагает материал
	не умеет	выделяет основные идеи, но не видит проблем	Выделяет конкретную проблему, однако излишне упрощает ее	Способен выделить и сравнить концепции, но испытывает сложности с их практической привязкой	Аргументированно проводит сравнение концепций по заданной проблематике
	не знает	допускает много ошибок	Может изложить основные рабочие категории	Знает основные отличия концепций в заданной проблемной области	Способен выделить специфику концепций в заданной проблемной области
продвинутый	не владеет	ориентируется в терминологии и содержании	В общих чертах понимает основную идею, однако плохо связывает ее с существующей проблематикой	Видит источники современных проблем в заданной области анализа, владеет подходами к их решению	Способен грамотно обосновать собственную позицию относительно решения современных проблем в заданной области
	не умеет	выделяет основные идеи, но не видит их в развитии	Может понять практическое назначение основной идеи, но затрудняется выявить ее	Выявляет основания заданной области анализа, понимает ее практическую ценность, однако испытывает	Свободно ориентируется в заданной области анализа. Понимает ее основания и умеет

			основания	затруднения в описании сложных объектов анализа	выделить практическое значение заданной области
	не знает	допускает ошибки при выделении рабочей области анализа	Способен изложить основное содержание современных научных идей в рабочей области анализа	Знает основное содержание современных научных идей в рабочей области анализа, способен их сопоставить	Может дать критический анализ современным проблемам в заданной области анализа

#### 4. Структура и содержание дисциплины

Общая трудоемкость (объем) дисциплины (модуля) составляет 7 зачетные единицы, 252 академических часа.

Общая трудоемкость (объем) дисциплины (модуля) составляет 4 зачетные единицы, 144 академических часа.

Объём дисциплины	Всего часов	Часов по семестрам			
		1 семестр	2 семестр	3 семестр	4 семестр
<b>Общая трудоёмкость дисциплины</b>	576	144	144	144	144
<b>Контактная работа обучающихся с преподавателям (по видам аудиторных учебных занятий) – всего:</b>	272	72	64	72	64
в том числе:					
лекции	136	36	32	36	32
Лабораторные занятия	136	36	32	36	32
<b>Самостоятельная работа (СРС) – всего:</b>	304	72	80	72	80
в том числе:					
курсовая работа			+		+
<b>Вид промежуточной аттестации (зачет/экзамен)</b>	Экзамен (1,2,4) Зачет 3	Экзамен	Экзамен	Зачет	Экзамен

#### Очная форма обучения

№ п/п	Раздел и тема дисциплины	Семестр	Виды учебной работы, в т.ч. самостоятельная работа студентов, час.			Формы текущего контроля успеваемости	Занятия в активной и интерактивной форме, час.	Формируемые компетенции
			Лекции	Лабораторные работы	Самост. работа			
<b>1</b>	<b>Информатика</b>	<b>1</b>	<b>36</b>	<b>36</b>	<b>72</b>			<b>ОПК-4 ОПК- 5</b>
<b>1.1</b>	Информация и ее свойства	<b>1</b>	4			Экзамен		ОПК-4 ОПК- 5

1.2	Измерение информации	1	4	36				ОПК-4 ОПК- 5
1.3	Системы счисления	1	4					ОПК-4 ОПК- 5
1.4	Аппаратное обеспечение средств вычислительной техники	1	24					ОПК-4 ОПК- 5
2	<b>Языки программирования</b>	2						ОПК-4 ОПК- 5
2.1	Введение	2	2					ОПК-4 ОПК- 5
2.2	Особенности языка Си	2	2	2	1	Защита лабораторных работ Экзамен	2	ОПК-4 ОПК- 5
2.3	Базовые понятия языка: Основные операции и конструкции	2	6	8	2	Ответ на зачёте. Ответы на экзаменах	8	ОПК-4 ОПК- 5
2.4	Массивы, функции пользователя	2	6	6	2	Ответ на зачёте. Ответы на экзаменах	6	ОПК-4 ОПК- 5
2.5	Типы данных, определяемые пользователем	2	4	4	6	Ответ на зачёте. Ответы на экзаменах	4	ОПК-4 ОПК- 5
2.6	Препроцессорные средства, Обработка файлов	2	6	6	2	Ответ на зачёте. Ответы на экзаменах.	6	ОПК-4 ОПК- 5

2.7	Классы памяти и области видимости данных	2	2	2	2	Ответ на зачёте. Ответы на экзаменах.	2	ОПК-4 ОПК- 5
2.8	Динамическое распределение памяти	2	4	4	2	Ответ на зачёте. Ответы на экзаменах.	5	ОПК-4 ОПК- 5
2.9	Расширенное представление данных	3	2	2	4	Ответы на зачетах	2	ОПК-4 ОПК- 5
2.10	Основные принципы объектно-ориентированного программирования	3	8	8	10	Ответы на зачетах Ответ на защите КР	8	ОПК-4 ОПК- 5
2.11	Технологии объектно-ориентированного программирования	3	10	10	20	Ответы на зачетах Ответ на защите КР	10	ОПК-4 ОПК- 5
2.12	Наследование	3	4	6	6	Ответы на зачетах Ответ на защите КР	6	ОПК-4 ОПК- 5
2.13	Полиморфизм	3	4	4	6	Ответы на зачетах Ответ на защите КР	4	ОПК-4 ОПК- 5
2.14	Объекты и классы	3	6	2	4	Ответы на зачетах Ответ на защите КР	2	ОПК-4 ОПК- 5
2.15	Потоки данных	3	2	4	4	Ответы на зачетах Ответ на защите КР	4	ОПК-4 ОПК- 5

3	<b>Методы программирования</b>	4						ОПК-4 ОПК- 5
3.1	Введение	4	4	4	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.2	Проектирование программного обеспечения при структурном подходе	4	4	4	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.3	Проектирование программного обеспечения при объектно-ориентированном подходе	4	4	4	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.4	Жизненный цикл и этапы разработки программного обеспечения	4	4	4	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.5	Паттерны проектирования	4	4	4	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.6	Базовые структуры данных. Алгоритмы	4	4	4	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.7	Алгоритмы сортировки и поиска	4	2	2	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.8	Алгоритмы на графах	4	2	2	8	Ответы на экзаменах.		ОПК-4 ОПК- 5

3.9	Тестирование программных продуктов	4	2	2	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
3.10	Отладка программного обеспечения	4	2	2	8	Ответы на экзаменах.		ОПК-4 ОПК- 5
Итого			136	136	304			

## 4.2. Содержание разделов дисциплины

### 4.2.1. Информатика

#### Информация и ее свойства

Информатика. Понятие информации

#### Измерение информации

Измерение информации. Количество информации. Обработка информации

#### Системы счисления

Основные понятия. Преобразование чисел из одной системы счисления в другую.

#### Аппаратное обеспечение средств вычислительной техники

История развития вычислительной техники. Поколения ЭВМ. Общая структура ЭВМ. Системный блок. Внутренняя память. Периферийные устройства персонального компьютера. Программное обеспечение ЭВМ: системное и инструментальное ПО.

### 4.2.2. Языки программирования

#### Введение

Начальные сведения о языках программирования. Роль языков программирования. Определение целей создания программы. Характеристики и свойства языков программирования. История развития языков программирования. Поколения языков.

#### Особенности языка Си

Этапы разработки программы. Структурные особенности языка Си.

Основные элементы языка программирования. Инструментальные средства разработки программ на языках высокого уровня в различных операционных системах. Ключевые слова и лексемы

#### Базовые понятия языка: Основные операции и конструкции

Алфавит, идентификаторы, служебные слова, константы, комментарии. Структура программы. Этапы создания исполняемой программы. Типизация языка. Определение типа. Контроль типов. Уровни типизации. Эквивалентность типов. Простые типы данных. Переменные. Классы памяти. Операции, выражения, операторы.

Операторы условия. Операторы организации циклов. Вложенные циклы. Операторы передачи управления.

#### Массивы, функции пользователя

Указатели. Ссылки. Одномерные и многомерные массивы. Строки. Связь указателей и

массивов. Способы инициализации числовых и текстовых массивов. Базовые алгоритмы и программы для обработки одномерных и двумерных массивов. Динамические массивы.

Шаблоны функций. Определение и вызов функций пользователей. Способы передачи параметров в функцию. Рекурсивные функции. Подставляемые функции. Перегрузка функций. Указатели на функции. Аргументы по умолчанию. Передача массивов в функции.

#### **Типы данных, определяемые пользователем**

Понятие структуры, примеры использования структур, операции над структурами. Объединения, перечислимый тип. Определение размера памяти в байтах (операция sizeof). Динамическое выделение памяти. Динамические структуры.

Объединения, поля битов, доступ к отдельному биту.

#### **Препроцессорные средства, Обработка файлов**

Стадии препроцессорной обработки. Директивы. Макроподстановки. Условная компиляция. Препроцессоры и директивы условной компиляции.

Понятие потока. Открытие и закрытие файлов. Операции, функции ввода-вывода результатов вычислений и текстов.

#### **Классы памяти и области видимости данных**

Классы памяти. Описатели классов памяти. Модели памяти. Распределение памяти при работе программы. Управление экраном и памятью в текстовом режиме.

#### **Динамическое распределение памяти**

Функции для динамического распределения памяти. Понятие очереди, Линейные списки, Двусвязные списки. Операции над списками. Методы хранения списков. Стеки и их обработка.

#### **Расширенное представление данных**

Абстрактные типы данных. Получение абстрактного типа. Создание интерфейса. Использование интерфейса. Реализация интерфейса представления данных. Имитация реальных ситуаций с помощью очереди. Деревья двоичного поиска

#### **Основные принципы объектно-ориентированного программирования**

Основные понятия объектно-ориентированного программирования (ООП). Отличия ООП от процедурного программирования. Понятие класса и объекта. Определение шаблона класса. Использование шаблона класса. Правила описания шаблонов. Специализация шаблона класса.

Понятия наследования и полиморфизма.

#### **Технологии объектно-ориентированного программирования**

Концепция и основные правила перегрузки операций. Перегрузка операций методами класса, дружественными и обычными функциями. Перегрузка различных операций.

#### **Наследование**

Механизм наследования. Базовые и производные классы. Типы наследования. Правила наследования различных методов. Множественное наследование.

#### **Полиморфизм**

Виртуальные и чистые виртуальные функции. Абстрактные классы. Статическое и динамическое связывание. Механизм динамического связывания. Виртуальные деструкторы.

#### **Объекты и классы**

Спецификаторы доступа к членам класса. Объявление или спецификация класса. Доступ к членам класса. Типы конструкторов. Вызов конструкторов. Статистические данные класса. Константные методы класса. Конструкторы и деструкторы. Дружественные классы и функции. Статические элементы класса. Константные методы и объекты. Указатель this. Указатели и ссылки на объекты, массивы объектов.

#### **Потоки данных**

Классификация способов ввода – вывода данных. Принципы работы с потоками и

файлами. Стандартные классы потоков. Форматированный ввод-вывод базовых типов. Манипуляторы ввода и вывода данных. Файловые потоки данных.

### **4.2.3 Методы программирования**

#### **Введение**

Задачи дисциплины, связь с другими курсами.

#### **Проектирование программного обеспечения при структурном подходе**

Разработка структурных и функциональных схем. Использование метода пошаговой детализации для проектирования структуры программного обеспечения (ПО). Структурные карты Консантайна. Проектирование структур данных. Проектирование ПО, основанное на декомпозиции данных. Case- технологии, основанные на методологиях анализа и проектирования. Программирование с защитой от ошибок.

#### **Проектирование программного обеспечения при объектно-ориентированном подходе**

Объектная модель. Основные принципы и элементы объектной модели. Виды связей между элементами объектной модели. Понятие и цель объектно-ориентированного анализа. Подходы к анализу объектно-ориентированных систем. Преимущества и риски, связанные с объектно-ориентированным проектированием (ООП). Универсальный язык ООП Java. Определение и история создания языка Java. Платформа Java. Java-технология. Пакеты- Java. Методы – Java. Объявление и вызов методов. Использование строк. Основные базовые конструкции. Коллекции. Создание конструкторов. Классы, методы класса. Сравнение и перегрузка объектов. Обработка исключений. Java-приложения.

#### **Жизненный цикл и этапы разработки программного обеспечения**

Понятие технологии программирования и основные этапы ее развития. Жизненный цикл программного обеспечения (ПО). Стандарты, регламентирующие жизненный цикл. Модели жизненного цикла ПО. Основные этапы разработки сложных программных систем. Планирование и организация разработки ПО. Проектирование ПО. Кодирование ПО.

#### **Паттерны проектирования**

Понятие и описание паттерна проектирования. Принципы применения паттернов. Классификация паттернов. Подходы к выбору паттернов.

#### **Базовые структуры данных. Алгоритмы**

Базовые структуры данных: множество, стек, очередь, дек, список, деревья, двоичные деревья поиска. Понятие алгоритма. Принципы анализа алгоритмов.

#### **Алгоритмы сортировки и поиска**

Основные понятия и классификация алгоритмов сортировки. Алгоритмы внутренней сортировки: Шелла, быстрая, пирамидальная, слиянием и лексикографическая. Внешняя сортировка: основные понятия, построение цепочек, многофазное слияние.

Основные понятия. Поиск в последовательно организованных структурах. Поиск в древовидных структурах. Хеширование и разрешение коллизий.

#### **Алгоритмы на графах**

Основные понятия. Способы задания графов. Алгоритмы обхода графов в ширину и глубину. Основные деревья минимальной стоимости и алгоритмы их построения. Алгоритмы нахождения кратчайших расстояний между вершинами.

#### **Тестирование программных продуктов**

Виды контроля качества разрабатываемого ПО. Ручной контроль. Структурное тестирование. Функциональное тестирование. Тестирование модулей и комплексное тестирование. Оценочное тестирование.

#### **Отладка программного обеспечения**

Качество ПО. Методы контроля качества ПО. Ошибки в программах. Методы отладки ПО. Документирование ПО. Сопровождение ПО.

### 4.3. Семинарские, практические, лабораторные занятия, их содержание

Таблица 4.3 1. Перечень лабораторных занятий

№ раздела	Наименование лабораторных работ	Формируемые компетенции
1.1	Создание одностраничных документов в msword	ОПК-4 ОПК-5
1.2	Форматирование документов в msword	ОПК-4 ОПК-5
1.3	Организация вычислений в электронной таблице MSEXEL	ОПК-4 ОПК-5
1.4	Решение квадратных уравнений систем линейных алгебраических уравнений(слау) средствами ms excel	ОПК-4 ОПК-5
2.1	Освоение работы в интегрированной среде программирования	ОПК-4 ОПК-5
2.2	Базовые операции языка Си	ОПК-4 ОПК-5
2.3	Организация вычислительных процессов	ОПК-4 ОПК-5
2.4	Условные операторы и операторы выбора	ОПК-4 ОПК-5
2.5	Обработка одномерных массивов	ОПК-4 ОПК-5
2.6	Обработка двумерных массивов	ОПК-4 ОПК-5
2.7	Функции пользователя и динамическое распределение памяти	ОПК-4 ОПК-5
2.8	Организация работы с файлами	ОПК-4 ОПК-5
2.9	Обработка списков	ОПК-4 ОПК-5
2.10	Вывод графиков функций	ОПК-4 ОПК-5
2.11	Освоение работы в среде C++	ОПК-4 ОПК-5
3.1	Обработка списков	ОПК-4 ОПК-5

3.2	Вывод графиков функций	ОПК-4 ОПК-5
3.3	Освоение работы в среде C++	ОПК-4 ОПК-5

**5. Учебно-методическое обеспечение самостоятельной работы студентов и оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины**

**5.1. Текущий контроль**

**Примерные темы курсовых работ:**

Номер варианта	Задание
Вариант 1 Разработать модульный алгоритм и программу, выполняющую действия с массивом:	Задан текст в виде прямоугольного рваного массива. Написать программу, выполняющую следующие действия в виде функций: Определить 1) Количество строк, содержащих одинаковое количество символов. 2) Найти строку, начинающуюся с заданного символа. 3) Найти строку, содержащую хотя бы один пробел
Вариант 2 Разработать модульный алгоритм и программу, выполняющую действия с массивом:	Задана целочисленная прямоугольная матрица. Написать программу, выполняющую следующие действия в виде функций: 1) Осуществить динамический ввод матрицы. 2) Определить количество столбцов, не содержащих ни одного нулевого элемента. 3) Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов.
Вариант 3 Разработать модульный алгоритм и программу, выполняющую действия с массивом:	Дана целочисленная прямоугольная матрица. Написать программу, выполняющую следующие действия в виде функций: 1) Осуществить динамический ввод матрицы. 2) Определить количество столбцов, содержащих хотя бы один нулевой элемент; 3) Определить номер строки, в которой находится самая длинная серия одинаковых элементов.

**Порядок оформления курсовой работы.**

Объем курсовой работы от 30 до 35 страниц машинописного текста через 1-1,5 интервала. При наличии приложений не более 40 страниц. **Примерная структура курсовой работы:**

- титульный лист (1стр.) – наименование темы,

- содержание (1стр.)
- введение (1-2 стр.);
- изложение основной части, состоящей из 2 глав и 2-3 вопросов (20-25 стр.);
- заключение, в котором должны быть сформулированы теоретические выводы, а также рекомендации и предложения (2-3 стр.);
- список использованной литературы (1-2 стр.);
- приложения (не более 5 стр.).

**Во введении** студент обязан обосновать актуальность выбранной темы, кратко осветить существующий уровень её разработки, сформулировать цель и задачи КР, раскрыть предмет и объект исследования.

Указываются конкретные задачи КР (3-5 задач), которые предстоит решать в соответствии с целью. Это обычно делается в форме перечисления (*изучить, описать, установить, выявить, вывести формулу, разработать методiku и т.п.*). Формулировки этих задач необходимо делать как можно более тщательно, поскольку описание их решения должно составить содержание разделов курсовой работы. Это важно также и потому, что заголовки разделов рождаются именно из формулировок задач предпринимаемого исследования.

Обязательным элементом введения является формулировка объекта и предмета исследования. **Объект** - это процесс или явления, порождающие проблемную ситуацию и избранные для изучения. **Предмет** - это то, что находится в границах объекта. Объект и предмет исследования как категории научного процесса соотносятся между собой как общее и частное. В объекте выделяется та его часть, которая служит предметом исследования. Именно на него и направлено основное внимание исследователя. Именно предмет работы определяет тему работы, которая обозначается на титульном листе как заглавие.

**В основной части** подробно раскрывается содержание разделов и вопросов темы. Их рассмотрение должно отвечать требованиям научности, логической последовательности, конкретности и доказательности. В работах, посвященных современным проблемам важно показать тесную связь с жизнью.

### **Глава 1. Теоретическая часть**

Раскрываются основы темы, ее сущность и содержание, содержание основных понятий и терминов. Обоснование метода решения темы и ее формализация. Данная часть должна содержать алгоритм решения задачи, обоснование выбора языка и пакета прикладного программирования для решения данной задачи. Обосновывается выбор переменных и их типов. Определяется необходимый объем оперативной памяти для хранения данных и результатов. Составляется трассировочная таблица пошагового выполнения этапов решения задачи с оценочными данными.

### **Глава 2. Практическая часть**

Составляется программа реализации алгоритма задачи. Для удобства программирования и отладки программы производится декомпозиция алгоритма на отдельные модули (функции) Программа должна содержать необходимые заголовочные файлы, макросы и шаблоны функций пользователя.

Программируются функции пользователя. Составляется главная функция, как набор отдельных функций (модулей). Приводятся примеры результатов тестирования и отладки программы для всех условий, сформулированных в задании на КР. Даются к ним комментарии и обоснования.

**Заключение** представляет собой краткое обобщение сказанного в основной части работы, выводы, разработку рекомендаций и предложений, а также может включать краткую характеристику перспективы внедрения разработанной программы .

**Список использованных источников.** Включает только те источники, которые

использовались при написании курсовой работы, в том числе и электронные. Приводятся только те, на которые есть ссылки или сноски на источники из списка литературы.

В Приложении приводится инструкция и руководство пользователя.

## **5.2. Методические указания по организации самостоятельной работы**

Во время самостоятельной работы студенты знакомятся с существующими пакетами программирования графики. читают методические указания по выполнению лабораторных работ, читают дополнительный материал в виде лекционных занятий. В перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине «Языки программирования» входит:

1. Методические указания по выполнению лабораторных работ.
2. Дополнительные литературные источники

**Контроль исполнения** самостоятельных работ осуществляется преподавателем с участием студентов в форме обсуждения выполненных заданий и работ.

Источники для самостоятельной подготовки:

1. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования Си. - М.: Финансы и статистика, 2010.-352 с.
2. Шишкин А.Д., Чернецова Е.А. Практикум по учебной дисциплине “Языки программирования .” Программирование на языке С и С++.- СПб.: изд. РГГМУ, 2012.- 62 с.
3. Шишкин А.Д. Программирование на языке СИ. Учебное пособие. СПб.: Изд. РГГМУ, 2003.-104 с

## **5.3. Промежуточный контроль: зачет, экзамен**

### **Перечень вопросов для промежуточной аттестации Информатика**

1. Место и роль информатики в современном обществе.
2. Основные понятия об информации (свойства, формы, классификация информации).
3. Измерение информации.
4. Количество информации
5. Кодирование информации
6. Основные понятия систем счисления. Классы систем счисления
7. Преобразование чисел из одной системы счисления в другую
8. Основные этапы развития ЭВМ.
9. Основные характеристики ЭВМ и вычислительных систем.
10. Классификация средств вычислительной техники.
11. Структурная схема и основные компоненты ПК.
12. Общие принципы построения современных ЭВМ.
13. Основные узлы и блоки IBM-совместимого компьютера.
14. Основные архитектуры ЭВМ
15. Описание архитектуры фон-неймановских машин
16. Классификация программного обеспечения
17. Системное программное обеспечение
18. Назначение и функции операционных систем.
19. Интерфейсы операционных систем.
20. Инструментальное программное обеспечение
21. Классификация языков программирования
22. Этапы решения задач с помощью ЭВМ
23. Понятие об алгоритме (формы, свойства, виды алгоритмов).
24. Формы представления алгоритмов

25. Правила изображения графических алгоритмов
26. Основные структуры алгоритмов
27. Вычислительные комплексы и сети
28. Базовая модель OSI
29. Услуги сети Интернет
30. Поисковые системы счисления

**Перечень вопросов для промежуточной аттестации Языки программирования(зачёта):**

1. Какие стандартные функции содержатся в файлах `stdio.h` и `conio.h` ?
2. Что означает «инкремент»?
3. Что означает «декремент»?
4. Что означает операция присваивания?
5. Какой спецификатор применяется для ввода и вывода строки символов?
6. Назовите основные операторы циклических процессов.
7. Назовите основные параметры цикла.
8. Как образуется бесконечный цикл и как выйти из него?
9. Какие управляющие средства выполнения программ существуют в языке Си?

Объясните их синтаксис.

10. Объясните назначение всех функций, использованных в программах.
11. Какие операции используются в управляющих средствах выполнения программ.
12. Объясните различия между циклами `while` и `do ... while`.
13. Для каких целей используется оператор `switch` ?
14. С какой целью используется индексация массивов?
15. С какой целью используется тип данных?
16. Опишите примененный Вами алгоритм обработки массива.
17. Как определить адрес  $(i, j)$  – того элемента в памяти машины?
18. Как распределяется двумерный массив в памяти машины?
19. Что означает транспонирование матрицы?
20. как поменять местами строку и столбец?
21. Какие циклы называются вложенными?
22. Какой переход называется условным / безусловным?
23. Для чего используется оператор `#define`?
24. Назовите операторы условной компиляции.
25. В чем состоит отличие объявления функции от ее определения?
26. Как объявить функцию с переменным числом параметров?
27. Какие типы объектов могут быть использованы в качестве формальных

параметров?

28. Что такое локальные объекты? Какова их область видимости и «время жизни»?
29. В чем состоит отличие автоматических переменных от статических?
30. Объясните механизм передачи параметров по значению, по указателю, по ссылке.
31. Какой массив называется динамическим?
32. Что такое поток данных?
33. В чем состоит различие ввода данных в стандартном потоке и с внешнего

устройства?

34. Напишите команды открытия файла для чтения и записи, закрытия файла.
35. В чем состоит различие команд `fputs()` и `fgets()`?
36. Какой список называется линейным?
37. Какой список называется кольцевым?
38. Как проверить: входит ли заданный элемент в список?
39. Как найти список с заданным значением информационного поля?

40. Дайте определение класса и объекта.
41. Дайте определение инкапсуляции и наследования.
42. Какие функции называются перегружаемые и почему?
43. Что общего между классом и структурой в языке C++?

**Перечень вопросов для промежуточной аттестации Языка программирования (экзамена):**

1. Основные этапы и лексемы создания программы.
2. Назначение компилятора и компоновщика.
3. Классификация языков программирования.
4. Структурное программирование
5. Переменные, константы, строки.
6. Имена и типы, область видимости.
7. Стандартные математические функции.
8. Стандартные библиотечные функции.
9. Типы данных. Простые типы.
10. Операции над данными, приведение типов.
11. Инкрементные операции языка Си.
12. Арифметические операции языка. Приоритет операций.
13. Бинарные операции.
14. Вывод результатов. Функции вывода.
15. Функции форматированного ввода данных
16. Спецификации полей данных.
17. Функции ввода строк и символов
18. Адресация данных. Операции над адресами.
19. Функции ввода данных.
20. Поразрядные (битовые) операции.
21. Назначение и использование оператора `if`.
22. Вложенные конструкции оператора `if`
23. Оператор для организации циклов `for`.
24. Операторы для организации циклов `while`, `do... while`
25. Операторы передачи управления.
26. Синтаксис использования операторов `break` и `continue`.
27. Директивы условной компиляции.
28. Препроцессоры языка Си.
29. Макросы: назначение и использование
30. Объявления и инициализация массивов.
31. Символьные массивы. Объявления и инициализация.
32. Указатели: назначение, объявления и инициализация. Операции над указателями.
33. Связь массива и указателя. Различие операций `&` и `*`.
34. Типовая структура программы на языке Си с функциями.
35. Функции пользователя: основные определения.
36. Рекурсивные функции.
37. Назначение и использование оператора **`return`**.
38. Передача аргументов в функцию `main()`.
39. Динамическое распределение памяти. Функции для выделения и освобождения памяти.
40. Назначение и использование двойных указателей. Динамические массивы.
41. Структуры языка Си. Инициализация структуры.
42. Массивы структур языка Си.
43. Глобальные и локальные переменные. Правила инициализации.

44. Функции обработки строк strcmp(), strncmp().
45. Функции обработки строк strcpy(), strncpy().
46. Функции обработки строк strcat(), strncat().
47. Назначение и содержание заголовочного файла ctype.h
48. Организация работы с файлами: Открытие и закрытие.
49. Организация работы с файлами: режимы работы с файлами.
50. Файловые функции обработки символов и строк.
51. Форматированный ввод/вывод в файл/из файла.
52. Управление экраном, курсором и цветом.
53. Организация видеопамяти в текстовом режиме

**Перечень вопросов для промежуточной аттестации Методы программирования (экзамена):**

1. Что понимают под термином «технология программирования»?
2. Что называют подходом и чем подход отличается от метода?
3. Назовите основные периоды истории развития технологии программирования. Чем характеризуются эти периоды? Как изменялись основные подходы и используемые средства?
4. Дайте определение понятию «сложная иерархическая система». Какой подход используют при разработке таких систем? На каких характеристиках этих систем он основан? В чем особенность данного подхода при разработке программного обеспечения?
5. Что понимают под термином «жизненный цикл программного обеспечения»? Какие основные процессы включают в это понятие?
6. Назовите основные этапы разработки программного обеспечения. Какие основные задачи решаются на этих этапах?
7. Назовите основные модели жизненного цикла программного обеспечения. С чем связано появление новых моделей?
8. Какие технологии называют CASE-технологиями? Почему?
9. Назовите основные составляющие любой CASE-технологии.
10. Перечислите основные положения технологии RAD? Какие программные системы нельзя разрабатывать с использованием этой технологии?
11. Что понимают под моделями качества процессов разработки программного обеспечения? Для чего они разработаны? Что гарантирует сертификация качества процессов? Почему?
12. Почему мы говорим, что современный этап развития технологии программирования характеризуется переходом от ремесленного к промышленному производству программного обеспечения.
13. Как описывают структуру программного обеспечения при объектном подходе? Что такое «пакет»? Для чего используют диаграммы пакетов?
14. Какие стереотипы классов введены и почему?
15. Разработайте диаграмму пакетов графического редактора, описанного вами при выполнении задания. Какие пакеты включены в эту диаграмму и почему? Какие пакеты будут связаны между собой?
16. Постройте диаграмму последовательности действий для объектов любых предложенных вами пакетов. Какими сообщениями обмениваются объекты? Какую информацию программист получит; анализируя эту диаграмму?
17. Какую диаграмму используют при уточнении взаимодействия объектов? Постройте эту диаграмму для объектов предыдущего задания.
18. Перечислите основные компоненты классов. Как описывают эти компоненты?
19. В каких случаях используют диаграммы состояний объекта? Постройте диаграмму состояний для любого управляющего объекта.
20. Постройте уточненную диаграмму классов по результатам исследования

- взаимодействия объектов. Какая еще информация необходима для реализации этих классов?
21. Что понимают под диаграммой компонентов? Какую информацию она содержит? В каких случаях целесообразно строить диаграммы компонентов?
  22. Какую информацию содержит диаграмма размещения?
  23. В каких случаях целесообразно использовать эти диаграммы?
  24. Какой список называется линейным?
  25. Какой список называется кольцевым?
  26. Как проверить: входит ли заданный элемент в список?
  27. Как найти список с заданным значением информационного поля?
  28. Дайте определение класса и объекта.
  29. Дайте определение инкапсуляции и наследования.
  30. Какие функции называются перегружаемые и почему?
  31. Что общего между классом и структурой в языке C?
  32. Что является целью тестирования программ? Почему?
  33. Перечислите известные вам виды контроля качества программного обеспечения. На каких этапах применяют каждый из них?
  34. Какие подходы к тестированию вы знаете? В чем они заключаются?
  35. Почему функциональное тестирование называют «тестированием по методу черного ящика»? Перечислите методы функционального тестирования и определите, в каких случаях следует использовать каждый из них.
  36. Почему структурное тестирование называют «тестированием по методу белого или прозрачного ящика»? Перечислите методы структурного тестирования и определите возможности каждого из них. Какой метод структурного тестирования обеспечивает наибольшую вероятность обнаружения ошибок?
  37. Используя методы обоих подходов, сформируйте пакет тестов для тестирования программы, вычисляющей действительные корни квадратного уравнения. Какие методы вы использовали и почему?
  38. Чем нисходящее тестирование отличается от восходящего? Что понимают под комплексным тестированием и чем оно отличается от тестирования компонент? Когда можно прекращать тестирование компонентов?
  39. Перечислите виды тестирования системы в целом. В каких случаях применяют каждый из них?

**Образец билета:**

Российский государственный гидрометеорологический университет  
Кафедра информационных технологий и систем безопасности

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1**

Дисциплина «Языки программирования»

1. Организация работы с файлами: режимы работы с файлами.
2. Связь массива и указателя. Различие операций & и \*.

Одобрено на заседании кафедры 15 декабря 20\_\_ г.

«Утверждаю»  
Зав. кафедрой

проф. В.Г.Бурлов.

## 6. Учебно-методическое и информационное обеспечение дисциплины

### а) основная литература:

1. Черпаков, И. В. Теоретические основы информатики : учебник и практикум для академического бакалавриата / И. В. Черпаков. — М. : Издательство Юрайт, 2018. — 353 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-9916-8562-7. — Режим доступа : [www.biblio-online.ru/book/78AD1E84-B91E-4ABA-9F16-5C4786292A2E](http://www.biblio-online.ru/book/78AD1E84-B91E-4ABA-9F16-5C4786292A2E)

2. Мойзес, О. Е. Информатика. Углубленный курс : учебное пособие для прикладного бакалавриата / О. Е. Мойзес, Е. А. Кузьменко. — М. : Издательство Юрайт, 2018. — 157 с. — (Серия : Университеты России). — ISBN 978-5-9916-7051-7. — Режим доступа : [www.biblio-online.ru/book/9AB4BED0-28D5-4A02-BC68-3ABC7EB50E0D](http://www.biblio-online.ru/book/9AB4BED0-28D5-4A02-BC68-3ABC7EB50E0D)

3. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — М. : Издательство Юрайт, 2018. — 104 с. — (Серия : Университеты России). — ISBN 978-5-534-07559-5. — Режим доступа : [www.biblio-online.ru/book/3D35AAB8-863B-47A8-BA39-ABF5D579204D](http://www.biblio-online.ru/book/3D35AAB8-863B-47A8-BA39-ABF5D579204D)

4. Огнева, М. В. Программирование на языке с++: практический курс : учебное пособие для бакалавриата и специалитета / М. В. Огнева, Е. В. Кудрина. — М. : Издательство Юрайт, 2018. — 335 с. — (Серия : Бакалавр и специалист). — ISBN 978-5-534-05123-0. — Режим доступа : [www.biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4](http://www.biblio-online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4)

5. Малявко, А. А. Формальные языки и компиляторы : учебное пособие для вузов / А. А. Малявко. — М. : Издательство Юрайт, 2018. — 429 с. — (Серия : Университеты России). — ISBN 978-5-534-04288-7. — Режим доступа : [www.biblio-online.ru/book/B4D96654-71D5-4748-986D-66E8309C25E3](http://www.biblio-online.ru/book/B4D96654-71D5-4748-986D-66E8309C25E3)

6. Зыков, С. В. Программирование : учебник и практикум для академического бакалавриата / С. В. Зыков. — М. : Издательство Юрайт, 2018. — 320 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-534-02444-9. — Режим доступа : [www.biblio-online.ru/book/122D27F3-13E4-4095-8946-C619F0FCC5C3](http://www.biblio-online.ru/book/122D27F3-13E4-4095-8946-C619F0FCC5C3)

### б) дополнительная литература

1. Шишкин А.Д., Чернецова Е.А. Практикум "Программирование на языке Си" [Текст] : учеб. пособие / РГГМУ. - Санкт-Петербург : РГГМУ, 2003. - 51 с. - 25.00 р.

2. Шишкин А.Д., Чернецова Е.А. Практикум по дисциплине "Языки программирования". Раздел: Программирование С и С++ [Текст] : практикум / РГГМУ. - 2-е изд., испр. и доп. - Санкт-Петербург : РГГМУ, 2012. - 68 с. - 29.43 р.

3. Шишкин А.Д. Программирование на языке Си [Текст] : учебное пособие / РГГМУ. - Санкт-Петербург : РГГМУ, 2003. - 103 с. - 70.00 р.

4. Назаров, С. В. Программирование в пакетах MS Office [Электронный ресурс] : учеб. пособие / С. В. Назаров, П. П. Мельников, Л. П. Смольников и др.; под ред. С. В. Назарова. - М.: Финансы и статистика, 2007. - 656 с.: ил. - ISBN 978-5-279-02926-6 - Режим доступа: <http://znanium.com/bookread2.php?book=369386>

## 7. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид учебных занятий	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и

	попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на лабораторном занятии.
Лабораторные	На лабораторных занятиях выполняются работы по программированию физических и математических задач, изученные во время лекций. Как правило, на каждом занятии студент должен показать результаты выполнения лабораторной преподавателю.
Внеаудиторная работа	представляет собой вид занятий, которые каждый студент организует и планирует самостоятельно. Самостоятельная работа студентов включает: <ul style="list-style-type: none"> <li>– самостоятельное изучение разделов дисциплины;</li> <li>– выполнение дополнительных индивидуальных творческих заданий.</li> </ul>
Подготовка к зачёту/экзамену	При подготовке к зачёту/экзамену необходимо ориентироваться на конспекты лекций, рекомендуемую литературу и др.

**8. Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

Тема (раздел) дисциплины	Образовательные и информационные технологии	Перечень программного обеспечения и информационных справочных систем
Введение	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++
Особенности языка Си	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Базовые понятия языка: Основные операции и конструкции	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Указатели, массивы, ссылки, функции пользователя	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Типы данных, определяемые пользователем	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Препроцессорные средства. Обработка файлов	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++

		C++Builder 6.0
Препроцессорные средства. Обработка файлов	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Динамическое распределение памяти	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Расширенное представление данных	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Основные принципы объектно-ориентированного программирования	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Технологии ООП	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Наследование	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Полиморфизм	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Объекты и классы	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Потоки данных	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0

## **9. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

Обучение обучающихся с ограниченными возможностями здоровья при необходимости осуществляется на основе адаптированной рабочей программы с использованием специальных методов обучения и дидактических материалов, составленных с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся

(обучающегося).

При определении формы проведения занятий с обучающимся-инвалидом учитываются рекомендации, содержащиеся в индивидуальной программе реабилитации инвалида, относительно рекомендованных условий и видов труда.

При необходимости для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья создаются специальные рабочие места с учетом нарушенных функций и ограничений жизнедеятельности.

#### **10. Материально-техническое обеспечение дисциплины**

Для проведения лекционных занятий используются обычные, и в некоторых случаях, мультимедийные аудитории. Лабораторные занятия проводятся в компьютерном классе кафедры с ЛВС, связанной Интернетом.

## Раздел III

### Методы программирования

#### 1. Цели освоения дисциплины

**Цель дисциплины:** формирование теоретических знаний, освоение методов и средств программирования практических инженерных задач на современных пакетах объектно-ориентированного языка, подготовка специалистов к деятельности, связанной с разработкой и эффективной реализацией программных средств для решения профессиональных задач.

**Задачи** изучения дисциплины:

- изучение основных подходов к организации процесса разработки программного обеспечения;
- изучение базовых структур данных;
- изучение основных алгоритмов сортировки и поиска;
- изучение алгоритмов генерации перестановок, подмножеств;
- изучение основных алгоритмов на графах и паттернов программирования;
- освоение основных методов оценки вычислительной сложности алгоритмов.
- обучить технологии составления проектно-технической документации на разрабатываемый программный продукт

#### 2. Место дисциплины в структуре ОП

Дисциплина читается студентам 2-го курса в рамках дисциплины «Организационное и правовое обеспечение информационной безопасности».

Дисциплина «Методы программирования» относится к числу дисциплин базовой части профессионального цикла. Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в процессе изучения дисциплин:

- «Алгебра и геометрия» – владеть методами линейной алгебры;
- «Информатика» – уметь пользоваться сетевыми средствами для обмена данными, в том числе с использованием глобальной информационной сети Интернет, владеть навыками работы с офисными приложениями;
- «Методы программирования» - знать язык программирования высокого уровня (объектно-ориентированное программирование); уметь работать с интегрированной средой разработки программного обеспечения; владеть навыками разработки, документирования, тестирования и отладки программ;
- «Английский язык» – владеть в объеме, необходимом для работы с языками программирования.

Дисциплина «Методы программирования» является предшествующей для изучения следующих базовых дисциплин: «Информационные технологии», «Моделирование систем и сетей телекоммуникаций». Знания и практические навыки, полученные в результате освоения дисциплины «Методы программирования», используются студентами при разработке курсовых и дипломных работ, в научно-исследовательской работе.

## Компетенции обучающегося, формируемые в результате освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Код компетенции	Компетенция
ОПК-4	способностью понимать значение информации в развитии современного общества, применять достижения информационных технологий для поиска и обработки информации
ОПК-5	способностью применять программные средства системного и прикладного назначения, языки, методы и инструментальные средства программирования для решения профессиональных задач

### 3. Требования к результатам освоения дисциплины

В результате изучения дисциплины студент должен

Код компетенции	Результаты обучения
ОПК-4	<p>Знать:</p> <ul style="list-style-type: none"><li>– основы и стандарты разработки алгоритмов вычислительных процессов;</li><li>– методы составления алгоритмов;</li></ul> <p>Уметь:</p> <ul style="list-style-type: none"><li>– работать с интегрированной средой разработки программного обеспечения;</li></ul> <p>Владеть:</p> <ul style="list-style-type: none"><li>– навыками разработки и документирования программ.</li></ul>
ОПК-5	<p>Знать:</p> <ul style="list-style-type: none"><li>– методы формализации алгоритмов;</li><li>– основные приемы реализации алгоритмов на языках структурного программирования;</li><li>– основные приемы программирования, области и особенности применения языков объектно-ориентированного программирования (ООП).</li></ul> <p>Уметь:</p> <ul style="list-style-type: none"><li>– реализовывать на языке высокого уровня алгоритмы решения профессиональных задач, в том числе задач обработки потоков;</li></ul> <p>Владеть:</p> <ul style="list-style-type: none"><li>– навыками тестирования и отладки программ.</li></ul>

Основные признаки проявленности формируемых компетенций в результате освоения дисциплины «Методы программирования» сведены в таблице.

**Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания**

Основные признаки проявленности компетенции (дескрипторное описание уровня)				
6.	7.	8.	9.	
не владеет	слабо ориентируется в терминологии и содержании	Способен выделить основные идеи текста, работает с критической литературой	Владеет основными навыками работы с источниками и критической литературой	Способен критически анализировать материалы
не умеет	не выделяет основные идеи	Способен показать основную идею в развитии	Способен представить ключевую проблему в ее связи с другими процессами	Может анализировать идеи проблематики
не знает	допускает грубые ошибки	Знает основные рабочие категории, однако не ориентируется в их специфике	Понимает специфику основных рабочих категорий	Способен анализировать характерные черты
не владеет	плохо ориентируется в терминологии и содержании	Владеет приемами поиска и систематизации, но не способен свободно изложить материал	Свободно излагает материал, однако не демонстрирует навыков сравнения основных идей и концепций	Способен анализировать концепции и излагать материал
не умеет	выделяет основные идеи, но не видит проблем	Выделяет конкретную проблему, однако излишне упрощает ее	Способен выделить и сравнить концепции, но испытывает сложности с их практической привязкой	Аргументированно проводит анализ по заданным параметрам
не знает	допускает много ошибок	Может изложить основные рабочие категории	Знает основные отличия концепций в заданной проблемной области	Способен анализировать специфику заданной области
не владеет	ориентируется в терминологии и содержании	В общих чертах понимает основную идею, однако плохо связывает ее с существующей проблематикой	Видит источники современных проблем в заданной области анализа, владеет подходами к их решению	Способен анализировать собственную и современную проблематику заданной области
не умеет	выделяет основные идеи, но не видит их в развитии	Может понять практическое назначение основной идеи, но затрудняется выявить ее основания	Выявляет основания заданной области анализа, понимает ее практическую ценность, однако испытывает затруднения в описании сложных объектов анализа	Свободно анализирует заданную область. Понимает специфику и выделяет основные черты заданной области
не знает	допускает ошибки при выделении рабочей области анализа	Способен изложить основное содержание современных научных идей в рабочей области анализа	Знает основное содержание современных научных идей в рабочей области анализа, способен их сопоставить	Может анализировать проблематику рабочей области анализа

**Таблица 4.3.2 Перечень практических занятий**

**5. Учебно-методическое обеспечение самостоятельной работы студентов и оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины**

**5.1. Текущий контроль**

**Примерные темы курсовых работ:**

Номер варианта

Задание

- Вариант 1.  
Разработать модульный алгоритм программы, выполняющую следующие действия структурой:
- и
- и
- со
- Описать структуру с именем STUDENT, содержащую поля:
- фамилия и инициалы;
  - номер группы;
  - успеваемость (массив из пяти элементов);
- Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию номера группы;
  - вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0;
  - если таких студентов нет, вывести соответствующее сообщение.
- Вариант 2.  
Разработать модульный алгоритм программы, выполняющую следующие действия структурой:
- и
- и
- со
- Описать структуру с именем STUDENT, содержащую поля:
- фамилия и инициалы;
  - номер группы;
  - успеваемость (массив из пяти элементов);
- Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
  - вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
  - если таких студентов нет, вывести соответствующее сообщение.
- Вариант 3.  
Разработать модульный алгоритм программы, выполняющую следующие действия структурой:
- и
- и
- со
- Описать структуру с именем STUDENT, содержащую поля:
- фамилия и инициалы;
  - номер группы;
  - успеваемость (массив из пяти элементов);
- Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по алфавиту;
  - вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;
  - если таких студентов нет, вывести соответствующее сообщение.
- Вариант 4.  
Разработать модульный алгоритм программы, выполняющую следующие действия структурой:
- и
- и
- со
- Описать структуру с именем AEROFLOT, содержащую следующие поля:
- название пункта назначения рейса;
  - номер рейса;
  - тип самолета;
- Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT; записи должны быть



модульный алгоритм программу, выполняющую следующие действия структурой:

- название пункта назначения ;
- и - номер поезда;
- время отправления;
- Написать программу, выполняющую следующие действия:
- со - ввод с клавиатуры данных в массив, состоящий из шести элементов типа TRAIN; записи должны быть упорядочены по времени отправления поезда;
- вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры;
- если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 9.  
Разработать модульный алгоритм программу, выполняющую следующие действия структурой:

- Описать структуру с именем TRAIN, содержащую следующие поля:
- название пункта назначения ;
- и - номер поезда;
- время отправления;
- Написать программу, выполняющую следующие действия:
- со - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN; записи должны быть упорядочены по номерам поездов;
- вывод на экран информации о поезде, номер которого введен с клавиатуры;
- если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 10.  
Разработать модульный алгоритм программу, выполняющую следующие действия структурой:

- Описать структуру с именем MARSH, содержащую следующие поля:
- название начального пункта маршрута;
- и - название конечного пункта маршрута;
- номер маршрута;
- Написать программу, выполняющую следующие действия:
- со - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH; записи должны быть упорядочены по номерам маршрутов;
- вывод на экран информации о маршруте, номер которого введен с клавиатуры;
- если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

Вариант 11.  
Разработать модульный алгоритм программу, выполняющую следующие действия структурой:

- Описать структуру с именем MARSH, содержащую следующие поля:
- название начального пункта маршрута;
- и - название конечного пункта маршрута;
- номер маршрута;
- Написать программу, выполняющую следующие действия:
- со - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH; записи должны быть упорядочены по номерам маршрутов;
- вывод на экран информации о маршрутах, которые начинаются или оканчиваются в пункте, название которого

- введено с клавиатуры;  
 - если таких маршрутов нет, выдать на дисплей соответствующее сообщение.
- Вариант 12.            Описать структуру с именем NOTE, содержащую следующие поля:  
 Разработать           и  
 модульный            - фамилия, имя;  
 алгоритм              - номер телефона;  
 программу,            - дата рождения (массив из трех чисел);  
 выполняющую         Написать программу, выполняющую следующие  
 следующие             действия:  
 действия              со   - ввод с клавиатуры данных в массив, состоящий из  
 структурой:            восьми элементов типа NOTE; записи должны быть  
                               упорядочены по датам рождения;  
                               - вывод на экран информации о человеке, номер телефона  
                               которого введен с клавиатуры;  
                               - если такого нет, выдать на дисплей соответствующее  
                               сообщение.
- Вариант 13.            Описать структуру с именем NOTE, содержащую следующие поля:  
 Разработать           и  
 модульный            - фамилия, имя;  
 алгоритм              - номер телефона;  
 программу,            - дата рождения (массив из трех чисел);  
 выполняющую         Написать программу, выполняющую следующие  
 следующие             действия:  
 действия              со   - ввод с клавиатуры данных в массив, состоящий из  
 структурой:            восьми элементов типа NOTE; записи должны быть  
                               размещены по алфавиту;  
                               - вывод на экран информации о людях, чьи дни рождения  
                               приходятся на месяц, значение которого введен с клавиатуры;  
                               - если таких нет, выдать на дисплей соответствующее  
                               сообщение.
- Вариант 14.            Описать структуру с именем NOTE, содержащую следующие поля:  
 Разработать           и  
 модульный            - фамилия, имя;  
 алгоритм              - номер телефона;  
 программу,            - дата рождения (массив из трех чисел);  
 выполняющую         Написать программу, выполняющую следующие  
 следующие             действия:  
 действия              со   - ввод с клавиатуры данных в массив, состоящий из  
 структурой:            восьми элементов типа NOTE; записи должны быть  
                               упорядочены по трем первым цифрам номера телефона;  
                               - вывод на экран информации о человеке, чья фамилия  
                               введена с клавиатуры;  
                               - если такого нет, выдать на дисплей соответствующее  
                               сообщение.
- Вариант 15.            Описать структуру с именем ZNAK, содержащую следующие поля:  
 Разработать           и  
 модульный            - фамилия, имя;  
 алгоритм              - знак Зодиака;  
 программу,            - дата рождения (массив из трех чисел);  
 выполняющую         Написать программу, выполняющую следующие  
 следующие             действия:  
 действия

действия со - ввод с клавиатуры данных в массив, состоящий из  
структурой: со - восьми элементов типа ZNAK; записи должны быть  
упорядочены по датам рождения;  
- вывод на экран информации о человеке, чья фамилия  
введена с клавиатуры;  
- если такого нет, выдать на дисплей соответствующее  
сообщение.

Вариант 16. Описать структуру с именем ZNAK, содержащую  
Разработать следующие поля:  
модульный - фамилия, имя;  
алгоритм и - знак Зодиака;  
программу, - дата рождения (массив из трех чисел);  
выполняющую Написать программу, выполняющую следующие  
следующие действия:  
действия со - ввод с клавиатуры данных в массив, состоящий из  
структурой: со - восьми элементов типа ZNAK; записи должны быть  
упорядочены по датам рождения;  
- вывод на экран информации о людях, родившихся под  
знаком, название которого введено с клавиатуры;  
- если таких нет, выдать на дисплей соответствующее  
сообщение.

Вариант 17. Описать структуру с именем ZNAK, содержащую  
Разработать следующие поля:  
модульный - фамилия, имя;  
алгоритм и - знак Зодиака;  
программу, - дата рождения (массив из трех чисел);  
выполняющую Написать программу, выполняющую следующие  
следующие действия:  
действия со - ввод с клавиатуры данных в массив, состоящий из  
структурой: со - восьми элементов типа ZNAK; записи должны быть  
упорядочены по знакам Зодиака;  
- вывод на экран информации о людях, родившихся в  
месяц, значение которого введено с клавиатуры;  
- если таких нет, выдать на дисплей соответствующее  
сообщение.

Вариант 18. Описать структуру с именем PRICE, содержащую  
Разработать следующие поля:  
модульный - название товара;  
алгоритм и - название магазина, в котором продается товар;  
программу, - стоимость товара в руб.;;  
выполняющую Написать программу, выполняющую следующие  
следующие действия:  
действия со - ввод с клавиатуры данных в массив, состоящий из  
структурой: со - восьми элементов типа PRICE; записи должны быть  
размещены в алфавитном порядке по названиям товаров;  
- вывод на экран информации о товаре, название которого  
введено с клавиатуры;  
- если таких товаров нет, выдать на дисплей  
соответствующее сообщение.

Вариант 19. Описать структуру с именем PRICE, содержащую  
Разработать следующие поля:  
модульный - название товара;

<p>алгоритм программу, выполняющую следующие действия структурой:</p>	<p>и</p> <p>со</p>	<p>- название магазина, в котором продается товар; - стоимость товара в руб. Написать программу, выполняющую следующие действия: - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа PRICE; записи должны быть размещены в алфавитном порядке по названиям магазинов; - вывод на экран информации о товарах, продающихся в магазине, название которого введено с клавиатуры; - если такого магазина нет, выдать на дисплей соответствующее сообщение.</p>
<p>Вариант 20. Разработать модульный алгоритм программу, выполняющую следующие действия структурой:</p>	<p>и</p> <p>со</p>	<p>Описать структуру с именем ORDER, содержащую следующие поля: - расчетный счет плательщика; - расчетный счет получателя; - перечисляемая сумма в руб. Написать программу, выполняющую следующие действия: - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ORDER; записи должны быть размещены в алфавитном порядке по расчетным счетам плательщиков; - вывод на экран информации о сумме, снятой с расчетного счета плательщика, введенного с клавиатуры; - если такого расчетного счета нет, выдать на дисплей соответствующее сообщение.</p>

### **Порядок оформления курсовой работы.**

Объём курсовой работы от 30 до 35 страниц машинописного текста через 1-1,5 интервала. При наличии приложений не более 40 страниц.

#### ***Примерная структура курсовой работы:***

- титульный лист (1стр.) – наименование темы,
- содержание (1стр.)
- введение (1-2 стр.);
- изложение основной части, состоящей из 2 глав и 2-3 вопросов (20-25 стр.);
- заключение, в котором должны быть сформулированы теоретические выводы, а также рекомендации и предложения (2-3 стр.);
- список использованной литературы (1-2 стр.);
- приложения (не более 5 стр.).

**Во введении** студент обязан обосновать актуальность выбранной темы, кратко осветить существующий уровень её разработки, сформулировать цель и задачи КР, раскрыть предмет и объект исследования.

Указываются конкретные задачи КР (*3-5 задач*), которые предстоит решать в соответствии с целью. Это обычно делается в форме перечисления (*изучить, описать, установить, выявить, вывести формулу, разработать методику и т.п.*). Формулировки этих задач необходимо делать как можно более тщательно, поскольку описание их решения должно составить содержание разделов курсовой работы. Это важно также и потому, что заголовки разделов рождаются именно из формулировок задач предпринимаемого исследования.

Обязательным элементом введения является формулировка объекта и предмета исследования. **Объект** - это процесс или явления, порождающие проблемную ситуацию и избранные для изучения. **Предмет** - это то, что находится в границах объекта. Объект

и предмет исследования как категории научного процесса соотносятся между собой как общее и частное. В объекте выделяется та его часть, которая служит предметом исследования. Именно на него и направлено основное внимание исследователя. Именно предмет работы определяет тему работы, которая обозначается на титульном листе как заглавие.

**В основной части** подробно раскрывается содержание разделов и вопросов темы. Их рассмотрение должно отвечать требованиям научности, логической последовательности, конкретности и доказательности. В работах, посвященных современным проблемам важно показать тесную связь с жизнью.

### **Глава 1. Теоретическая часть**

Раскрываются основы темы, ее сущность и содержание, содержание основных понятий и терминов. Обоснование метода решения темы и ее формализация. Данная часть должна содержать алгоритм решения задачи, обоснование выбора языка и пакета прикладного программирования для решения данной задачи. Обосновывается выбор переменных и их типов. Определяется необходимый объем оперативной памяти для хранения данных и результатов. Составляется трассировочная таблица пошагового выполнения этапов решения задачи с оценочными данными.

### **Глава 2. Практическая часть**

Составляется программа реализации алгоритма задачи. Для удобства программирования и отладки программы производится декомпозиция алгоритма на отдельные модули (функции) Программа должна содержать необходимые заголовочные файлы, макросы и шаблоны функций пользователя.

Программируются функции пользователя. Составляется главная функция, как набор отдельных функций (модулей). Приводятся примеры результатов тестирования и отладки программы для всех условий, сформулированных в задании на КР. Даются к ним комментарии и обоснования.

**Заключение** представляет собой краткое обобщение сказанного в основной части работы, выводы, разработку рекомендаций и предложений, а также может включать краткую характеристику перспективы внедрения разработанной программы.

**Список использованных источников.** Включает только те источники, которые использовались при написании курсовой работы, в том числе и электронные. Приводятся только те, на которые есть ссылки или сноски на источники из списка литературы.

В Приложении приводится инструкция и руководство пользователя.

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Вид учебных занятий	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на лабораторном занятии.

Лабораторные	На лабораторных занятиях выполняются работы по программированию физических и математических задач, изученные во время лекций. Как правило, на каждом занятии студент должен показать результаты выполнения лабораторной преподавателю.
Внеаудиторная работа	представляет собой вид занятий, которые каждый студент организует и планирует самостоятельно. Самостоятельная работа студентов включает: <ul style="list-style-type: none"> <li>– самостоятельное изучение разделов дисциплины;</li> <li>– выполнение дополнительных индивидуальных творческих заданий.</li> </ul>
Подготовка к зачёту/экзамену	При подготовке к зачёту/экзамену необходимо ориентироваться на конспекты лекций, рекомендуемую литературу и др.

**8. Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

Тема (раздел) дисциплины	Образовательные и информационные технологии	Перечень программного обеспечения и информационных справочных систем
Введение		MS Office 2007/2010, Turbo C++ Dev-C++
Проектирование программного обеспечения при структурном подходе	Лабораторные работы	MS Office 2007/2010,
Проектирование программного обеспечения при объектно-ориентированном подходе	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Жизненный цикл и этапы разработки программного обеспечения	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Паттерны проектирования		MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Базовые структуры данных. Алгоритмы	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Алгоритмы сортировки и поиска	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Алгоритмы генерации перестановок, подмножеств,	Лабораторные работы	MS Office 2007/2010, Turbo C++

сочетаний		Dev-C++ C++Builder 6.0
Алгоритмы на графах	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Тестирование программных продуктов	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Отладка программного обеспечения	Лабораторные работы	MS Office 2007/2010, Turbo C++ Dev-C++ C++Builder 6.0
Составление программной документации	Лабораторные работы	MS Office 2007/2010,

## **9. ОСОБЕННОСТИ ОСВОЕНИЯ ДИСЦИПЛИНЫ ДЛЯ ИНВАЛИДОВ И ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ**

Обучение обучающихся с ограниченными возможностями здоровья при необходимости осуществляется на основе адаптированной рабочей программы с использованием специальных методов обучения и дидактических материалов, составленных с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся (обучающегося).

При определении формы проведения занятий с обучающимся-инвалидом учитываются рекомендации, содержащиеся в индивидуальной программе реабилитации инвалида, относительно рекомендованных условий и видов труда.

При необходимости для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья создаются специальные рабочие места с учетом нарушенных функций и ограничений жизнедеятельности.

### **10. Материально-техническое обеспечение дисциплины**

Для проведения лекционных занятий используются обычные, и в некоторых случаях, мультимедийные аудитории. Лабораторные занятия проводятся в компьютерном классе с ЛВС, связанной Интернетом.