

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Таныгин Максим Олегович

Должность: и.о. декана факультета фундаментальной и прикладной информатики

Дата подписания: 06.09.2024 14:55:26

Уникальный программный ключ:

65ab2aa0d384efe8480e6a4c688edd0bc475e411a

## Аннотация к рабочей программе

### дисциплины « Объектно-ориентированное программирование»

#### 1. Цель преподавания дисциплины

воспитание общей логико-алгоритмической культуры и формирование определенного стиля мышления, необходимых для эффективного использования современных средств вычислительной техники;

- обучение студентов методикам постановки, подготовки и решению инженерно-технических задач на современных ЭВМ с использованием объектно-ориентированного подхода.

#### 2. Задачи изучения дисциплины

- ознакомить студентов с основными понятиями алгоритмизации и программирования задач на ЭВМ с использованием объектно-ориентированного подхода;

- привить интерес к программированию, как к одному из важнейших направлений развития современной вычислительной техники;

- научить студентов разработке алгоритмов и структур данных с их последующей записью на языке программирования высокого уровня с использованием объектно-ориентированного подхода;

- дать студентам удобный, надежный и современный инструментарий для решения алгоритмических задач на компьютере (современные среды разработки приложений, текстовые и графические редакторы. Интернет – технологии);

- развить логическое мышление и сформировать практические навыки разработки эффективных алгоритмов с использованием объектно-ориентированного подхода;

- выработать практические навыки работы с современными средами быстрой разработки приложений;

- обучить студентов методике отладки и тестирования программ в современных интегрированных средах и их документации с использованием текстовых и графических редакторов в рамках объектно-ориентированного подхода.

#### 3. Индикаторы компетенций, формируемые в результате освоения дисциплины:

ПК – 1.1 - Тестирует программные продукты

ПК – 1.2 - Обрабатывает данные тестирования программных продуктов

ПК – 1.3 - Оценивает достоверность и надежность результатов тестирования программных продуктов

ПК – 5.1 – Анализирует требования к программным системам

ПК – 5.2 – Аргументирует оценку и обоснование рекомендуемой архитектуры программного обеспечения

ПК – 5.3 – Разрабатывает программное средство и/или базу данных

ПК – 5.4 – Проектирует пользовательский интерфейс

ПК – 5.5 – Разрабатывает техническую документацию информационных систем

#### **4. Разделы дисциплины**

1. Введение
2. Жизненный цикл программы
3. Определение требований к программной системе
4. Проектирование программной системы
5. Логическая организация данных
6. Определение требований к компонентам программной системы
7. Проектирование компонент программной системы
8. Программирование (реализация) компонентов программной системы
9. Отладка компонент программной системы
10. Тестирование и компоновка программной системы
11. Качество и надежность программных средств
12. Управление разработкой программной системы
13. Объектно-ориентированный метод
14. Реализация средств ООП в языке программирования C++
15. Автоматизация разработки программных средств

МИНОБРНАУКИ РОССИИ

Юго-Западный государственный университет

УТВЕРЖДАЮ:

Декан факультета

фундаментальной и прикладной  
информатики

*(наименование ф-та полностью)*



М.О. Таныгин

*(подпись, инициалы, фамилия)*

«31» 08 2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование

*(наименование дисциплины)*

ОПОП ВО 09.03.01 Информатика и вычислительная техника

*шифр и наименование направления подготовки (специальности)*

направленность (профиль, специализация) «Интеллектуальные системы в цифровой экономике»

*наименование направленности (профиля, специализации)*

форма обучения очная

*(очная, очно-заочная, заочная)*

Курск – 2021

Рабочая программа дисциплины составлена в соответствии с ФГОС ВО – бакалавриат по направлению подготовки (специальности) 09.03.01 Информатика и вычислительная техника на основании учебного плана ОПОП ВО 09.03.01 Информатика и вычислительная техника, направленность (профиль, специализация) «Интеллектуальные системы в цифровой экономике», одобренного Ученым советом университета (протокол № 6 от «26» февраля 2021 г.).

Рабочая программа дисциплины обсуждена и рекомендована к реализации в образовательном процессе для обучения студентов по ОПОП ВО 09.03.01 Информатика и вычислительная техника, направленность (профиль, специализация) «Интеллектуальные системы в цифровой экономике» на заседании кафедры вычислительной техники «30» июня 2021 г. протокол № 12.

(наименование кафедры, дата, номер протокола)

Зав. кафедрой  д.т.н., профессор Титов В.С.

Разработчик программы

к.т.н.  Ватутин Э.И.

(ученая степень и ученое звание, Ф.И.О.)

Согласовано:

Директор научной библиотеки  Макаровская В.Г.

Рабочая программа дисциплины пересмотрена, обсуждена и рекомендована к реализации в образовательном процессе на основании учебного плана ОПОП ВО 09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Интеллектуальные системы в цифровой экономике», одобренного Ученым советом университета, протокол № 9 «25» 06 2021 г., на заседании кафедры вычислительной техники протокол № 15 «30» 06 2022 г.

Зав. кафедрой  И.Е. Чернуха

Рабочая программа дисциплины пересмотрена, обсуждена и рекомендована к реализации в образовательном процессе на основании учебного плана ОПОП ВО 09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Интеллектуальные системы в цифровой экономике», одобренного Ученым советом университета, протокол № 7 «28» 02 2022 г., на заседании кафедры вычислительной техники протокол № 1 «31» 08 2023 г.

Зав. кафедрой  И.Е. Чернуха

Рабочая программа дисциплины пересмотрена, обсуждена и рекомендована к реализации в образовательном процессе на основании учебного плана ОПОП ВО 09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Интеллектуальные системы в цифровой экономике», одобренного Ученым советом университета, протокол № 9 «27» 02 2023 г., на заседании кафедры вычислительной техники протокол № 1 «30» 08 2024 г.

Зав. кафедрой  И.Е. Чернуха

# **1 Цель и задачи дисциплины. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы**

## **1.1 Цель дисциплины**

Целями преподавания дисциплины являются:

- воспитание общей логико-алгоритмической культуры и формирование определенного стиля мышления, необходимых для эффективного использования современных средств вычислительной техники;
- обучение студентов методикам постановки, подготовки и решению инженерно-технических задач на современных ЭВМ с использованием объектно-ориентированного подхода.

## **1.2 Задачи дисциплины**

Основные задачи дисциплины следующие:

- ознакомить студентов с основными понятиями алгоритмизации и программирования задач на ЭВМ с использованием объектно-ориентированного подхода;
- привить интерес к программированию, как к одному из важнейших направлений развития современной вычислительной техники;
- научить студентов разработке алгоритмов и структур данных с их последующей записью на языке программирования высокого уровня с использованием объектно-ориентированного подхода;
- дать студентам удобный, надежный и современный инструментарий для решения алгоритмических задач на компьютере (современные среды разработки приложений, текстовые и графические редакторы, Интернет-технологии);
- развить логическое мышление и сформировать практические навыки разработки эффективных алгоритмов с использованием объектно-ориентированного подхода;
- выработать практические навыки работы с современными средами быстрой разработки приложений;
- обучить студентов методике отладки и тестирования программ в современных интегрированных средах и их документирования с использованием текстовых и графических редакторов в рамках объектно-ориентированного подхода.

## **1.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы**

<i>Планируемые результаты освоения основной профессиональной образовательной программы (компетенции, закрепленные за дисциплиной)</i>		<i>Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной</i>	<i>Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения компетенций</i>
<i>код компетенции</i>	<i>наименование компетенции</i>		

<i>Планируемые результаты освоения основной профессиональной образовательной программы (компетенции, закрепленные за дисциплиной)</i>		<i>Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной</i>	<i>Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения компетенций</i>
<i>код компетенции</i>	<i>наименование компетенции</i>		
ПК-1	Способен проводить юзабилити- исследование программных продуктов	ПК-1.1 Тестирует программные продукты	<b>Знать:</b> методы тестирования программных продуктов <b>Уметь:</b> решать задачи в области тестирования программных продуктов <b>Иметь опыт деятельности</b> по тестированию программных продуктов
		ПК-1.2 Обрабатывает данные тестирования программных продуктов	<b>Знать:</b> методы обработки данных при тестировании программных продуктов <b>Уметь:</b> решать задачи в области обработки данных при тестировании программных продуктов <b>Иметь опыт деятельности</b> по обработке данных при тестированию программных продуктов
		ПК-1.3 Оценивает достоверность и надежность результатов тестирования программных продуктов	<b>Знать:</b> методы оценки достоверности и надежности результатов тестирования программных продуктов <b>Уметь:</b> решать задачи в области оценки достоверности и надежности тестирования программных продуктов <b>Иметь опыт деятельности</b> по оценке достоверности и надежности при тестировании программных продуктов
ПК-5	Способен разрабатывать требования и проектировать программное обеспечение	ПК-5.1 Анализирует требования к программным системам	<b>Знать:</b> принципы анализа требований к программным продуктам <b>Уметь:</b> проводить анализ требований к программным продуктам <b>Иметь опыт деятельности</b> по анализу требований к программным продуктам
		ПК-5.2 Аргументирует оценку и обоснование рекомендуемой архитектуры программного обеспечения	<b>Знать:</b> современные методы аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения <b>Уметь:</b> производить аргументацию, оценку и обоснование архитектуры программного обеспечения <b>Владеть:</b> навыками аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения
		ПК-5.3 Разрабатывает программное средство и/или базу данных	<b>Знать:</b> современные методы разработки программных средств <b>Уметь:</b> разрабатывать программные средства <b>Владеть:</b> навыками в области разработки программных средств

Планируемые результаты освоения основной профессиональной образовательной программы (компетенции, закрепленные за дисциплиной)		Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной	Планируемые результаты обучения по дисциплине, соотнесенные с индикаторами достижения компетенций
код компетенции	наименование компетенции		
		ПК-5.4 Проектирует пользовательский интерфейс	<b>Знать:</b> методы проектирования пользовательского интерфейса <b>Уметь:</b> выполнять проектирование пользовательского интерфейса <b>Иметь опыт деятельности:</b> в области проектирования пользовательского интерфейса
		ПК-5.5 Разрабатывает техническую документацию информационных систем	<b>Знать:</b> базовые методы и приемы разработки документации информационных систем <b>Уметь:</b> разрабатывать техническую документацию информационных систем <b>Владеть</b> навыками разработки технической документации информационных систем

## 2 Указание места дисциплины в структуре основной профессиональной образовательной программы

Дисциплина «Объектно-ориентированное программирование» входит в часть, формируемую участниками образовательных отношений, блока 1 «Дисциплины (модули)» основной профессиональной образовательной программы – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль, специализация) «Интеллектуальные системы в цифровой экономике». Дисциплина изучается на 2 курсе в 3 семестре.

## 3 Объем дисциплины в зачетных единицах с указанием количества академических или астрономических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины составляет 5 зачетных единиц (з.е.), академических 180 часов.

Таблица 3 – Объем дисциплины

Виды учебной работы	Всего, часов
Общая трудоемкость дисциплины	180
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	55,15
в том числе:	
лекции	18
лабораторные занятия	18

практические занятия	18
Самостоятельная работа обучающихся (всего)	88,85
Контроль (подготовка к экзамену)	36
Контактная работа по промежуточной аттестации (всего АтКР)	1,15
в том числе:	
зачет	не предусмотрен
зачет с оценкой	не предусмотрен
курсовая работа (проект)	не предусмотрена
экзамен (включая консультацию перед экзаменом)	1,15

#### 4 Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1 Содержание дисциплины

№ п/п	Раздел (тема) дисциплины	Содержание
1.	Введение	Цель и задачи курса. Понятие «технологии программирования». Необходимость технологии программирования. Классификация программных средств. Понятие программного средства и программного продукта. Сложность программных средств. Характеристики качества программных средств.
2.	Жизненный цикл программы	Понятие жизненного цикла программы. Фазы жизненного цикла. Модели жизненного цикла программных средств: лавинообразная, спиральная, эволюционная, кубическая. Этапы разработки программных средств. Определение требований к программной системе, проектирование; определение требований к компонентам; проектирование компонент; реализация и программирование компонент; компоновка и тестирование; сопровождение. Модели распределения людских ресурсов при разработке программных средств.
3.	Определение требований к программной системе	Характеристики этапа определения требований и его особенности. Задача автоматизации разработки и верификации требований. Методы формализованного формирования требований к программной системе. Методы анализа контекста. Стратегия «почему-что-как». Цикл «читатель-автор». Формализация представления и документация требований. Язык описания требований. Requirement Statement Language (RST) – язык формализованного представления требований. Документирование требований к программной системе. Техническое задание на разработку системы. Состав и содержание пунктов технического задания.
4.	Проектирование программной системы	Содержание и особенности этапа. Методы проектирования программных систем. Их классификация и особенности. Метод функциональных схем. Условные графические обозначения элементов схем программ и систем. Принцип функциональной (алгоритмической) декомпозиции. Метод нисходящего проектирования. Понятие декомпозиции по данным. Метод проектирования «исток-преобразование-сток». Метод анализа потоков данных (метод Джексона).
5.	Логическая организация данных	Модели и структуры данных. Методы поиска информации по структурам данных. Поиск в списках, линейный поиск, прямой поиск, бинарный поиск, хэш-



		поиск. Поиск в графах: поиск в глубину, поиск в ширину, поиск с возвратом. Методы сортировки информации. Сортировка слиянием. Сортировка вычерпыванием.
6.	Определение требований к компонентам программной системы	Общая характеристика и особенности этапа. Внешние спецификации на компоненты системы. Метод НПО-диаграмм. НПО-технология определения требований к компонентам.
7.	Проектирование компонент программной системы	Способы описания алгоритмов: граф-схемы, псевдокоды, структурограммы, языки проектирования, R-схемы. Метод структурного программирования. Алгоритмические конструкции структурного программирования. Метод пошаговой детализации. Модульное программирование. Понятие модуля. Характеристики модулей: прочность и сцепление.
8.	Программирование (реализация) компонентов программной системы	Содержание этапа. Требования к компонентам программных систем. Стиль программирования на языках высокого уровня. Принципы стилистического оформления программ. Венгерская нотация идентификаторов. Ошибки в программах. Категории программных ошибок. Понятие о системных, технологических и алгоритмических ошибках. Способы защиты программ от ошибок.
9.	Отладка компонент программной системы	Методы отладки. Простые методы: «прокрутка за столом», анализ промежуточных результатов, обратное отслеживание, цикл «читатель-автор». Сложные методы: индукция, дедукция. Отладка программ без их исполнения: индивидуальный просмотр, коллективный сквозной контроль, инспекция. Использование программ по отладочному заданию. Регистрация результатов отладки, метод вставок, метод моделирования. Принципы и технологии отладки.
10.	Тестирование и компоновка программной системы	Тестирование как основной способ отладки. Общие принципы тестирования. Методы разработки тестов: методы «черного ящика», методы «белого ящика». Общая стратегия тестирования. Категории тестов. Характеристика объектов тестирования на различных этапах разработки программных систем. Компоновка программной системы. Методы компоновки: восходящий, нисходящий, метод ядра, метод «сэндвича». Организация приемо-сдаточных испытаний. Сертификация и аттестация. Документа стадии ввода в эксплуатацию.
11.	Качество и надежность программных средств	Критерии и характеристики качества программных средств. Стандарты качества. Надежность как основная характеристика качества программ. Характеристики и показатели надежности. Факторы, определяющие надежность программных средств. Методы обеспечения надежности программных средств. Понятие об информационной, временной и программной избыточности.
12.	Управление разработкой программных средств	Основные функции управления разработкой программных средств. Трудоемкость разработки программ. Методы оценки трудоемкости программных средств: нормативный метод, метод аналогии, метод экспертных оценок, оценка по количеству строк, проект бесконечной длины. Организация коллективной разработки программных средств. Бригадная форма организации разработки. Субъекты коллективной и бригадной разработки. Методы организации бригад.
13.	Объектно-ориентированный метод	Методология объектно-ориентированного программирования (ООП). Понятие объекта. Принципы ООП: абстракция данных, инкапсуляция, наследование, полиморфизм. Каноническая форма сложной системы. Понятие иерархии классов

		и объектной структуры. Объектно-ориентированная технология разработки программ. Этапы объектно-ориентированной технологии. Управление разработкой. Организация коллективной разработки. Субъекты коллективной разработки.
14.	Реализация средств ООП в языке программирования C++	Понятие класса. Класс как объектный тип. Определение класса. Структура класса. Компонентные данные и компонентные функции. Описание объектов класса. Организация доступа к компонентам класса. Статические компоненты. Понятие указателя на компонент класса. Указатели на компонентные данные и указатели на компонентные функции. Создание объектов. Работа с динамическими объектами. Конструкторы и деструкторы. Видимость компонент класса. Схемы доступа к компонентам. Способы модификации схем доступа. Ключи категории класса class, struct, union. Спецификаторы доступа public, private, protected. Дружественные компонентные функции. Дружественные классы. Спецификатор friend. Локальные классы. Механизм наследования. Определение производных классов. Понятие множественности наследования. Виртуальные классы. Полиморфизм и его реализация в C++. Статический и виртуальный полиморфизм. Переопределение компонентных функций. Виртуальные компонентные функции. Абстрактные классы и «чистые» виртуальные функции. Моделирование абстрактных понятий. Ограничения при использовании абстрактных классов. Параметризация классов. Шаблоны классов. Механизм классов и перегрузка (расширение действия) стандартных операций. Понятие операторной функции. Способы определения операторных функций. Перегрузка унарных и бинарных операций.
15.	Автоматизация разработки программных средств	Средства автоматизации проектирования и программирования. CASE-средства. Принципы построения CASE-систем. Концепция BASE-систем.

Таблица 4.1.2 – Содержание дисциплины и ее методическое обеспечение

№ п/п	Раздел (темы) дисциплины	Виды деятельности			Учебно-методические материалы	Форма текущего контроля успеваемости (по неделям семестра)	Компетенции
		лек., час.	№, лаб.	№, пр.,			
1	2	3	4	5	6	7	8
1	Введение	1		1	У-1, У-2, У-3	С	ПК-1
2	Жизненный цикл программы	1		1	У-1, У-2, У-3	С	ПК-1
3	Определение требований к программной системе	1		2	У-1, У-2, У-3	С	ПК-1
4	Проектирование программной системы	1		2	У-1, У-2, У-3	С	ПК-1
5	Логическая организация данных	1		2	У-1, У-2, У-3	С	ПК-1
6	Определение требований к компонентам программной системы	1		1	У-1, У-2, У-3	С	ПК-1
7	Проектирование компонент программной системы	1		1	У-1, У-2, У-3	С	ПК-1
8	Программирование (реализация) компонентов программной системы	1		1	У-1, У-2, У-3	С	ПК-1
9	Отладка компонент	1		1	У-1, У-2, У-3	С	ПК-5

	программной системы						
10	Тестирование и компоновка программной системы	1		1	У-1, У-2, У-3	С	ПК-5
11	Качество и надежность программных средств	1		1	У-1, У-2, У-3	С	ПК-5
12	Управление разработкой программных средств	1		1	У-1, У-2, У-3	С	ПК-5
13	Объектно-ориентированный метод	2		1	У-1, У-2, У-3	С	ПК-5
14	Реализация средств ООП в языке программирования С++	3	18	1	У-1, У-2, У-3, МУ-1 – МУ-6	С, ЗО	ПК-5
15	Автоматизация разработки программных средств	1		1	У-1, У-2, У-3	С	ПК-5
<b>Итого</b>		18	18	18			

У-і – учебная литература; МУ-і – методические указания; С – собеседование; ЗО – защита лабораторной работы в виде собеседования

## 4.2 Лабораторные занятия и (или) практические занятия

### 4.2.1 Лабораторные занятия

#### 4.2.1 Лабораторные занятия

Таблица 4.2.1 – Лабораторные работы

№	Наименование лабораторной работы	Объем, час.
1	Программирование линейных алгоритмов	4
2	Программирование разветвленных алгоритмов	4
3	Работа с массивами	4
4	Записи и файлы	2
5	Программирование циклических алгоритмов	2
6	Работа с массивами	2
Итого:		18

#### 4.2.2 Практические занятия

Таблица 4.2.2 – Практические занятия

№	Наименование практического занятия	Объем, час.
1	Программирование объектно-ориентированной иерархии классов	18
Итого:		18

## 4.3 Самостоятельная работа студентов (СРС)

Таблица 4.3 – Самостоятельная работа студентов

№	Наименование раздела дисциплины	Срок выполнения	Время, затрачиваемое на выполнение СРС, час.
1	Работа с графикой с использованием объектно-ориентированного подхода	2-18 недель	88,85
<b>Итого:</b>			88,85

### **5 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине**

Студенты могут при самостоятельном изучении отдельных тем и вопросов дисциплин пользоваться учебно-наглядными пособиями, учебным оборудованием и методическими разработками кафедры в рабочее время, установленное Правилами внутреннего распорядка работников.

Учебно-методическое обеспечение для самостоятельной работы обучающихся по данной дисциплине организуется:

*библиотекой университета:*

- библиотечный фонд укомплектован учебной, методической, научной, периодической, справочной и художественной литературой в соответствии с УП и данной РПД;
- имеется доступ к основным информационным образовательным ресурсам, информационной базе данных, в том числе библиографической, возможность выхода в Интернет.

*кафедрой:*

- путем обеспечения доступности всего необходимого учебно-методического и справочного материала;
- путем предоставления сведений о наличии учебно-методической литературы, современных программных средств, методических рекомендаций, пособий по организации самостоятельной работы студентов, вопросов к зачету, методических указаний к выполнению лабораторных и практических работ и т.д.

*типографией университета:*

- помощь авторам в подготовке и издании научной, учебной и методической литературы;
- удовлетворение потребности в тиражировании научной, учебной и методической литературы.

### **6 Образовательные технологии. Технологии использования воспитательного потенциала дисциплины**

Реализация компетентного подхода предусматривает широкое использование в образовательном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования универсальных, общепрофессиональных и профессиональных компетенций обучающихся.

Таблица 6.1. Интерактивные образовательные технологии, используемые при проведении аудиторных занятий

№	Наименование раздела (лекции и лабораторного занятия)	Используемые интерактивные образовательные технологии	Объем в часах
1	Работа с графикой	Разбор конкретных ситуаций	40
Итого			40

Содержание дисциплины обладает значительным воспитательным потенциалом, поскольку в нем аккумулирован исторический и современный социокультурный и (или) научный опыт человечества. Реализация воспитательного потенциала дисциплины осуществляется в рамках единого образовательного и воспитательного процесса и способствует непрерывному развитию личности каждого обучающегося. Дисциплина вносит значимый вклад в формирование общей и (или) профессиональной культуры обучающихся. Содержание дисциплины способствует духовно-нравственному, гражданскому, патриотическому, правовому, экономическому, профессионально-трудовому, культурно-творческому, физическому, экологическому воспитанию обучающихся.

Реализация воспитательного потенциала дисциплины подразумевает:

- целенаправленный отбор преподавателем и включение в лекционный материал, материал для практических и (или) лабораторных занятий содержания, демонстрирующего обучающимся образцы настоящего научного подвижничества создателей и представителей данной отрасли науки (производства, экономики, культуры), высокого профессионализма ученых (представителей производства, деятелей культуры), их ответственности за результаты и последствия деятельности для природы, человека и общества; примеры подлинной нравственности людей, причастных к развитию науки, культуры, экономики и производства, а также примеры высокой духовной культуры, патриотизма, гражданственности, гуманизма, творческого мышления;

- применение технологий, форм и методов преподавания дисциплины, имеющих высокий воспитательный эффект за счет создания условий для взаимодействия обучающихся с преподавателем, другими обучающимися, представителями работодателей (командная работа, проектное обучение, деловые игры, разбор конкретных ситуаций, решение кейсов, мастер-классы, круглые столы, диспуты и др.);

- личный пример преподавателя, демонстрацию им в образовательной деятельности и общении с обучающимися за рамками образовательного процесса высокой общей и профессиональной культуры.

Реализация воспитательного потенциала дисциплины на учебных занятиях направлена на поддержание в университете единой развивающей образовательной и воспитательной среды. Реализация воспитательного потенциала дисциплины в ходе самостоятельной работы обучающихся способствует развитию в них целеустремленности, инициативности, креативности, ответственности за результаты своей ра-

боты – качеств, необходимых для успешной социализации и профессионального становления.

## 7 Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

### 7.1 Перечень компетенций с указанием этапов их формирования в процессе освоения основной профессиональной образовательной программы

Таблица 7.1 – Этапы формирования компетенций

Код и наименование компетенции	Этапы формирования компетенций и дисциплины (модули) и практики, при изучении/ прохождении которых формируется данная компетенция		
	начальный	основной	завершающий
1	2	3	4
ПК-1 Способен проводить юзабилити- исследование программных продуктов	Объектно-ориентированное программирование, Учебная ознакомительная практика	Разработка мобильных приложений	Web-программирование
ПК-5 Способен разрабатывать требования и проектировать программное обеспечение	Объектно-ориентированное программирование	Операционные системы, Разработка мобильных приложений, Базы данных	Производственная преддипломная практика, Цифровые платформы, Информационные системы маркетинга и менеджмента в цифровой экономике, Web-программирование

### 7.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Таблица 7.2 – Показатели и критерии оценивания компетенций, шкала оценивания

Код компетенции / этап (указывается название этапа из п.7.1)	Показатели оценивания компетенций (индикаторы достижения компетенций, закрепленные за дисциплиной)	Уровни сформированности компетенции		
		Пороговый (удовлетворительный)	Продвинутый (хорошо)	Высокий (отлично)
ПК-1	ПК-1.1 Тестирует программные продукты	<b>Знать:</b> методы тестирования программных продуктов на пороговом уровне <b>Уметь:</b> решать задачи в области тестирования программных продуктов на пороговом уровне <b>Иметь опыт деятельности</b> по тестированию программных продуктов на пороговом уровне	<b>Знать:</b> методы тестирования программных продуктов на продвинутом уровне <b>Уметь:</b> решать задачи в области тестирования программных продуктов на продвинутом уровне <b>Иметь опыт деятельности</b> по тестированию программных продуктов на продвинутом уровне	<b>Знать:</b> методы тестирования программных продуктов на высоком уровне <b>Уметь:</b> решать задачи в области тестирования программных продуктов на высоком уровне <b>Иметь опыт деятельности</b> по тестированию программных продуктов на высоком уровне

Код компетенции / этап (указывается название этапа из п.7.1)	Показатели оценивания компетенций (индикаторы достижения компетенций, закрепленные за дисциплиной)	Уровни сформированности компетенции		
		Пороговый (удовлетворительный)	Продвинутый (хорошо)	Высокий (отлично)
	ПК-1.2 Обрабатывает данные тестирования программных продуктов	<p><b>Знать:</b> методы обработки данных при тестирования программных продуктов на пороговом уровне</p> <p><b>Уметь:</b> решать задачи в области обработки данных при тестирования программных продуктов на пороговом уровне</p> <p><b>Иметь опыт деятельности</b> по обработке данных при тестированию программных продуктов на пороговом уровне</p>	<p><b>Знать:</b> методы обработки данных при тестирования программных продуктов на продвинутом уровне</p> <p><b>Уметь:</b> решать задачи в области обработки данных при тестирования программных продуктов на продвинутом уровне</p> <p><b>Иметь опыт деятельности</b> по обработке данных при тестированию программных продуктов на продвинутом уровне</p>	<p><b>Знать:</b> методы обработки данных при тестирования программных продуктов на высоком уровне</p> <p><b>Уметь:</b> решать задачи в области обработки данных при тестирования программных продуктов на высоком уровне</p> <p><b>Иметь опыт деятельности</b> по обработке данных при тестированию программных продуктов на высоком уровне</p>
	ПК-1.3 Оценивает достоверность и надежность результатов тестирования программных продуктов	<p><b>Знать:</b> методы оценки достоверности и надежности результатов тестирования программных продуктов на пороговом уровне</p> <p><b>Уметь:</b> решать задачи в области оценки достоверности и надежности тестирования программных продуктов на пороговом уровне</p> <p><b>Иметь опыт деятельности</b> по оценке достоверности и надежности при тестировании программных продуктов на пороговом уровне</p>	<p><b>Знать:</b> методы оценки достоверности и надежности результатов тестирования программных продуктов на продвинутом уровне</p> <p><b>Уметь:</b> решать задачи в области оценки достоверности и надежности тестирования программных продуктов на продвинутом уровне</p> <p><b>Иметь опыт деятельности</b> по оценке достоверности и надежности при тестировании программных продуктов на продвинутом уровне</p>	<p><b>Знать:</b> методы оценки достоверности и надежности результатов тестирования программных продуктов на высоком уровне</p> <p><b>Уметь:</b> решать задачи в области оценки достоверности и надежности тестирования программных продуктов на высоком уровне</p> <p><b>Иметь опыт деятельности</b> по оценке достоверности и надежности при тестировании программных продуктов на высоком уровне</p>
ПК-5	ПК-5.1 Анализирует требования к программным системам	<p><b>Знать:</b> принципы анализа требований к программным продуктам на пороговом уровне</p> <p><b>Уметь:</b> проводить анализ требований к программным продуктам на пороговом уровне</p> <p><b>Иметь опыт деятельности</b> по анализу требований к программным продуктам на пороговом уровне</p>	<p><b>Знать:</b> принципы анализа требований к программным продуктам на продвинутом уровне</p> <p><b>Уметь:</b> проводить анализ требований к программным продуктам на продвинутом уровне</p> <p><b>Иметь опыт деятельности</b> по анализу требований к программным продуктам на продвинутом уровне</p>	<p><b>Знать:</b> принципы анализа требований к программным продуктам на высоком уровне</p> <p><b>Уметь:</b> проводить анализ требований к программным продуктам на высоком уровне</p> <p><b>Иметь опыт деятельности</b> по анализу требований к программным продуктам на высоком уровне</p>

Код компетенции / этап (указывается название этапа из п.7.1)	Показатели оценивания компетенций (индикаторы достижения компетенций, закрепленные за дисциплиной)	Уровни сформированности компетенции		
		Пороговый (удовлетворительный)	Продвинутый (хорошо)	Высокий (отлично)
ПК-5.2	Аргументирует оценку и обоснование рекомендуемой архитектуры программного обеспечения	<p><b>Знать:</b> современные методы аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения на пороговом уровне</p> <p><b>Уметь:</b> производить аргументацию, оценку и обоснование архитектуры программного обеспечения на пороговом уровне</p> <p><b>Владеть:</b> навыками аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения на пороговом уровне</p>	<p><b>Знать:</b> современные методы аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения на продвинутом уровне</p> <p><b>Уметь:</b> производить аргументацию, оценку и обоснование архитектуры программного обеспечения на продвинутом уровне</p> <p><b>Владеть:</b> навыками аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения на продвинутом уровне</p>	<p><b>Знать:</b> современные методы аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения на высоком уровне</p> <p><b>Уметь:</b> производить аргументацию, оценку и обоснование архитектуры программного обеспечения на высоком уровне</p> <p><b>Владеть:</b> навыками аргументации оценки и обоснования рекомендуемой архитектуры программного обеспечения на высоком уровне</p>
ПК-5.3	Разрабатывает программное средство и/или базу данных	<p><b>Знать:</b> современные методы разработки программных средств на пороговом уровне</p> <p><b>Уметь:</b> разрабатывать программные средства на пороговом уровне</p> <p><b>Владеть:</b> навыками в области разработки программных средств на пороговом уровне</p>	<p><b>Знать:</b> современные методы разработки программных средств на продвинутом уровне</p> <p><b>Уметь:</b> разрабатывать программные средства на продвинутом уровне</p> <p><b>Владеть:</b> навыками в области разработки программных средств на продвинутом уровне</p>	<p><b>Знать:</b> современные методы разработки программных средств на высоком уровне</p> <p><b>Уметь:</b> разрабатывать программные средства на высоком уровне</p> <p><b>Владеть:</b> навыками в области разработки программных средств на высоком уровне</p>
ПК-5.4	Проектирует пользовательский интерфейс	<p><b>Знать:</b> методы проектирования пользовательского интерфейса на пороговом уровне</p> <p><b>Уметь:</b> выполнять проектирование пользовательского интерфейса на пороговом уровне</p> <p><b>Иметь опыт деятельности:</b> в области проектирования пользовательского интерфейса на пороговом уровне</p>	<p><b>Знать:</b> методы проектирования пользовательского интерфейса на продвинутом уровне</p> <p><b>Уметь:</b> выполнять проектирование пользовательского интерфейса на продвинутом уровне</p> <p><b>Иметь опыт деятельности:</b> в области проектирования пользовательского интерфейса на продвинутом уровне</p>	<p><b>Знать:</b> методы проектирования пользовательского интерфейса на высоком уровне</p> <p><b>Уметь:</b> выполнять проектирование пользовательского интерфейса на высоком уровне</p> <p><b>Иметь опыт деятельности:</b> в области проектирования пользовательского интерфейса на высоком уровне</p>



Код компетенции / этап (указывается название этапа из п.7.1)	Показатели оценивания компетенций (индикаторы достижения компетенций, закрепленные за дисциплиной)	Уровни сформированности компетенции		
		Пороговый (удовлетворительный)	Продвинутый (хорошо)	Высокий (отлично)
	ПК-5.5 Разрабатывает техническую документацию информационных систем	<p><b>Знать:</b> базовые методы и приемы разработки документации информационных систем на пороговом уровне</p> <p><b>Уметь:</b> разрабатывать техническую документацию информационных систем на пороговом уровне</p> <p><b>Владеть</b> навыками разработки технической документации информационных систем на пороговом уровне</p>	<p><b>Знать:</b> базовые методы и приемы разработки документации информационных систем на продвинутом уровне</p> <p><b>Уметь:</b> разрабатывать техническую документацию информационных систем на продвинутом уровне</p> <p><b>Владеть</b> навыками разработки технической документации информационных систем на продвинутом уровне</p>	<p><b>Знать:</b> базовые методы и приемы разработки документации информационных систем на высоком уровне</p> <p><b>Уметь:</b> разрабатывать техническую документацию информационных систем на высоком уровне</p> <p><b>Владеть</b> навыками разработки технической документации информационных систем на высоком уровне</p>

### 7.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе основной профессиональной образовательной программы

Таблица 7.3 - Паспорт комплекта оценочных средств для текущего контроля

№ п/п	Раздел (тема) дисциплины	Код контролируемой компетенции (или ее части)	Технология формирования	Оценочные средства		Описание шкал оценивания
				наименование	№№ заданий	
1	Введение	ПК-1	Лекция, СРС, практическая работа	ФОС для собеседования	По заданной теме	Согласно табл. 7.2
2	Жизненный цикл программы Определение требований к программной системе Проектирование программной системы	ПК-1	Лекция, СРС, практическая работа	ФОС для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№1	
				Контрольные вопросы к ЛР	№1	
3	Логическая организация данных Определение требований к компонентам программной системы Проектирование компонент программной системы	ПК-1	Лекция, СРС, практическая работа	ФОС для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№1	
				Контрольные вопросы к ЛР	№1	
4	Программирование (реализация)	ПК-1	Лекция, СРС, практическая	ФОС для собеседования	По заданной	Согласно табл. 7.2

	компонентов программной системы Отладка компонент программной системы Тестирование и компоновка программной системы		работа	я	теме	
				Практическая работа	№1	
				Контрольные вопросы к ЛР	№2	
5	Качество и надежность программных средств Управление разработкой программных средств Объектно-ориентированный метод	ПК-1	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№1	
				Контрольные вопросы к ЛР	№2	
6	Реализация средств ООП в языке программирования C++	ПК-1	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№1	
7	Введение Жизненный цикл программы	ПК-1	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Контрольные вопросы к ЛР	№3	
8	Определение требований к программной системе Проектирование программной системы	ПК-1	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№1	
9	Логическая организация данных Определение требований к компонентам программной системы	ПК-5	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Контрольные вопросы к ЛР	№4	
10	Проектирование компонент программной системы Программирование (реализация) компонентов программной системы	ПК-5	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Контрольные вопросы к ЛР	№5	
11	Отладка компонент программной системы	ПК-5	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
12	Тестирование и компоновка программной системы Качество и надежность программных средств	ПК-5	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№1	
13	Управление разработкой программных средств Объектно-ориентированный метод	ПК-5	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№1	
14	Реализация средств ООП в языке программирования C++ Введение	ПК-5	Лекция, СРС, лабораторная работа, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№2	

				Контрольные вопросы к ЛР	№6	
15	Жизненный цикл программы	ПК-5	Лекция, СРС, практическая работа	вопросы для собеседования	По заданной теме	Согласно табл. 7.2
				Практическая работа	№2	
				Контрольные вопросы к ЛР	№6	
				Контрольные вопросы к ЛР	№11	

### **Примеры типовых контрольных заданий для собеседования (текущего контроля)**

1. Что будет выведено на экран в результате выполнения программы:

```
var i,j: integer; x,y: real;
begin i:=5; j:=2; x:=3.1; y:=x*j; i:=i mod j; x:=i / j; j:=j + i div j; writeln(i,j,x,y) end.
```

Варианты ответа:

	i	j	x	y
a	5	2	3.1	6.2
b	1	2	0	6.2
c	1	2	0.5	6.2
d	2	2	0.5	6.2

2. Что будет выведено на экран в результате выполнения программы:

```
var v,w,p: byte;
begin v:=3; w:=2; p:=21; v:=p shl w; p:=not p shl w; w:=v and p; writeln(v,w,p) end.
```

Варианты ответа:

	v	w	p
a	42	4	86
b	84	4	172
c	84	8	86
d	84	8	172

3. Что будет выведено на экран в результате выполнения программы:

```
var x,y,z,w: boolean;
begin x:=true; y:=true; z:=false; z:=not x and y or z; w:=y or z xor y; y:=x and not y and z;
x:=y xor z; writeln(x,y,z,w) end.
```

Варианты ответа:

	x	y	z	w
a	false	false	false	false
b	false	true	false	false
c	true	true	false	false
d	true	false	true	false

4. Каким будет результат записи выражения  $y = \frac{1 + 8 \cos^2 x \cdot \log_2 |\operatorname{tg} x|}{2 \arcsin x}$  по правилам Паскаля:

```
a: y:=(1+8*sqr(cos(x))*ln(tan(x))/2*arcsin(x);
b: y:=(1+8*sqr(cos(x))*log2(sin(x)/cos(x))/(2*arctan(x/sqrt(1-sqr(x))));
c: y:=(1+8*sqr(cos(x))*ln(abs(sin(x)/cos(x)))/ln(2))/2/arctan(x/sqrt(1-sqr(x)));
d: y:=(1+8*sqr(cos(x))*ln(abs(sin(x)/cos(x)))/ln(2))/(2*arctan(sqrt(1-sqr(x))/x));
```

5. Что будет выведено на экран в результате выполнения программы:

```
var x,y: real; i,j: integer;
begin
  x:=sqrt(abs(-4))+0.1;
  i:=succ(round(x));
  y:=int(x)+frac(x);
  j:=ord(add(i))+i;
  writeln(x,y,i,j)
end.
```

Варианты ответа:

	x	y	i	j
a	2.1	2.1	3	4
b	16.1	16.1	17	18

c 2.1 2.1 2 3  
 d 2.1 2.1 3 3

6. Что будет выведено на экран в результате выполнения программы:

```
var x,y,z: integer;
begin
  x:=1; y:=2; z:=3;
  if x<y then x:=y else y:=x; if x=y then x:=z else y:=z;
  writeln(x,y,z)
end.
```

Варианты ответа:

	x	y	z
a	2	2	3
b	3	2	3
c	3	3	3
d	2	3	3

7. Что будет выведено на экран в результате выполнения программы:

```
var v,w: integer; x,y,z: real;
begin v:=12; w:=4; x:=0.5; y:=2.5e-1; v:=w div v; w:=w mod 3; z:=x/y; x:=x*z; y:=x+z-y;
writeln(v,w,x,y,z) end.
```

Варианты ответа:

	v	w	x	y	z
a	0	0	1	2.75	2
b	3	0	1	2.75	2
c	3	1	0.5	2.75	2
d	0	1	1	2.75	2

8. Что будет выведено на экран в результате выполнения программы:

```
var i,j,k: byte;
begin i:=2; j:=4; k:=j shl i; i:=j and k and i; j:=not i xor j; writeln(i,j,k) end.
```

Варианты ответа:

	i	j	k
a	255	251	32
b	255	251	16
c	0	251	16
d	0	4	16

9. Что будет выведено на экран в результате выполнения программы:

```
var i,j,k: boolean;
begin i:=false; j:=true; k:=not i xor j; i:=i or j and k; j:=j xor not i and k; writeln(i,j,k) end.
```

Варианты ответа:

	i	j	k
a	false	true	true
b	true	true	false
c	false	false	true
d	false	true	false

10. Что будет выведено на экран в результате выполнения программы:

```
var m,n,k: integer;
begin m:=0; n:=2; k:=4; if m>n then if n<k then n:=k else k:=m else m:=k; writeln(m,n,k) end.
```

Варианты ответа:

	m	n	k
a	4	2	4
b	0	2	0
c	0	4	4
d	0	2	4

11. Что будет выведено на экран в результате выполнения программы:

```
var i,n,k: integer;
begin
  k:=1; n:=1;
  for i=1 to 5 do begin
    if odd(k) then n:=n+1 else n:=n+2;
    k:=k+i
  end;
  writeln(i,n,k)
end.
```

Варианты ответа:

	i	n	k
a	6	8	16
b	6	7	16
c	6	8	15
d	6	8	17

12. Что будет выведено на экран в результате выполнения программы:

```
var i,n,k: integer;
begin
n:=10; i:=0; k:=1;
repeat
i:=i+k; k:=k+1;
if not odd(k) then n:=n+1;
until i>n;
writeln(i,n,k)
end.
```

Варианты ответа:

	i	n	k
a	14	13	5
b	14	13	6
c	15	13	6
d	15	12	6

13. Дан текст программы на Паскале:

```
var i,n,k: integer;
begin
n:=1; i:=5; k:=10;
while true do begin
n:=n*i; k:=k-1;
if n>i+k then break;
i:=i+1;
end;
writeln(i,n,k);
end.
```

Какое из перечисленных утверждений будет верно после ее выполнения:

- a:  $n < i$  и  $n < k$
- b:  $i < n$  и  $i < k$
- c:  $i < n$  и  $n < k$
- d:  $n < i$  и  $i < k$

14. Что будет выведено на экран в результате выполнения программы:

```
var a: array[1..10] of byte = (1,3,5,7,9,11,13,15,17,19);
var i,j,k: byte;
begin
i:=low(a)+2;
while i<=high(a) do begin
j:=i-1; k:=j-1;
a[i]:=a[j]+a[k];
i:=i+1;
end;
writeln(a[high(a)]);
end.
```

Варианты ответа:

- a: 19
- b: 100
- c: 122
- d: 123

15. Что будет выведено на экран в результате выполнения программы:

```
const Digits: array[0..10] of Char = '0123456789#0';
var i: byte = 5; j: byte = 3;
begin
writeln( Digits + i - j );
end.
```

Варианты ответа:

- a: 23456789
- b: '23456789'

c: 3456789  
d: 123456789

16. Что будет выведено на экран в результате выполнения программы:

```
var Digits: array[0..10] of Char = '0123456789'#0;
var i,j: byte;
begin
i:=StrLen(Digits)-1;
repeat dec(i); Digits[i-1]:=Digits[i] until i=1;
writeln( Digits );
end.
```

Варианты ответа:

a: 9999999999  
b: 9999999998  
c: 8888888889  
d: 8888888888

17. Что будет выведено на экран в результате выполнения программы:

```
type
TMonth = (Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec);
TMonths = set of TMonth;
const summer: TMonths = [Jun..Aug];
var ms: TMonths = []; i: TMonth;
begin
Include(ms,Jul); Include(ms,Sep);
if ms <= summer then i:=pred(Jun) else
if summer <= ms then i:=succ(Mar) else i:=pred(Dec);
writeln(ord(i))
end.
```

Варианты ответа:

a: 11  
b: 10  
c: 9  
d: Nov

18. Что будет выведено на экран в результате выполнения программы:

```
procedure P(const a: real; var b: real; out c: real);
begin b:=2*a*b; c:=0.5*a*b end;
var x,y,z: real;
begin
x:=3.5; y:=-1.5; z:=1.33;
P(x,y,z);
writeln(x,y,z)
end.
```

Варианты ответа:

	x	y	z
a	3.5	-10.5	1.33
b	3.5	-10.5	0.0
c	3.5	-10.5	-18.4
d	-3.5	-10.5	-18.4

19. Какая пара значений будет выведена на экран в результате выполнения программы:

```
function F(var n: integer): integer;
begin
if n<=0 then begin n:=1; result:=0; exit end;
if odd(n) then begin inc(n); result:=n*n; exit end;
result:=2*n
end;
var n: integer = 5;
begin
writeln(F(n)+F(n),n)
end.
```

Варианты ответа:

a: 46 6  
b: 48 5  
c: 46 5  
d: 48 6

20. Что будет выведено на экран в результате выполнения программы:

```
function Q(const x: integer): real;
begin
    if x <= 0 then begin result:=1; exit end;
    if x in [1,2] then begin result:=Q(x-1)*0.5; exit end;
    result:=2.0*Q(x-1);
end;
begin writeln(Q(5)) end.
```

Варианты ответа:

a: 0.25

b: 0.5

c: 1.0

d: 2.0

Экзамен проводится в виде бланкового и/или компьютерного тестирования.

Для тестирования используются контрольно-измерительные материалы (КИМ) – вопросы и задания в тестовой форме, составляющие банк тестовых заданий (БТЗ) по дисциплине, утвержденный в установленном в университете порядке.

Проверяемыми на промежуточной аттестации элементами содержания являются темы дисциплины, указанные в разделе 4 настоящей программы. Все темы дисциплины отражены в КИМ в равных долях (%). БТЗ включает в себя не менее 100 заданий и постоянно пополняется. БТЗ хранится на бумажном носителе в составе УММ и электронном виде в ЭИОС университета.

Для проверки *знаний* используются вопросы и задания в различных формах:

- закрытой (с выбором одного или нескольких правильных ответов),
- открытой (необходимо вписать правильный ответ),
- на установление правильной последовательности,
- на установление соответствия.

*Умения, навыки(или опыт деятельности) и компетенции* проверяются с помощью компетентностно-ориентированных задач (ситуационных, производственных или кейсового характера) и различного вида конструкторов. Все задачи являются многоходовыми. Некоторые задачи, проверяющие уровень сформированности компетенций, являются многовариантными. Часть умений, навыков и компетенций прямо не отражена в формулировках задач, но они могут быть проявлены обучающимися при их решении.

В каждый вариант КИМ включаются задания по каждому проверяемому элементу содержания во всех перечисленных выше формах и разного уровня сложности. Такой формат КИМ позволяет объективно определить качество освоения обучающимися основных элементов содержания дисциплины и уровень сформированности компетенций.

#### **7.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, регулируются следующими нормативными актами университета:

– положение П 02.016–2018 «О балльно-рейтинговой системе оценивания результатов обучения по дисциплинам (модулям) и практикам при освоении обучающимися образовательных программ»;

– методические указания, используемые в образовательном процессе, указанные в списке литературы.

Для *текущего контроля успеваемости* по дисциплине в рамках действующей в университете балльно-рейтинговой системы применяется следующий порядок начисления баллов:

Таблица 7.4 – Порядок начисления баллов в рамках БРС

Форма контроля	Минимальный балл		Максимальный балл	
	балл	примечание	балл	примечание
1	2	3	4	5
Лабораторная работа № 1 (Программирование линейных алгоритмов)	2	Выполнил, но не защитил	4	Выполнил и защитил
Лабораторная работа № 2 (Программирование разветвленных алгоритмов)	2	Выполнил, но не защитил	4	Выполнил и защитил
Лабораторная работа № 3 (Работа с массивами)	2	Выполнил, но не защитил	4	Выполнил и защитил
Лабораторная работа № 4 (Записи и файлы)	2	Выполнил, но не защитил	4	Выполнил и защитил
Лабораторная работа № 5 (Программирование циклических алгоритмов)	2	Выполнил, но не защитил	4	Выполнил и защитил
Лабораторная работа № 6 (Программирование циклических алгоритмов)	2		4	
Практическая работа №1 (Работа с массивами)	2	Выполнил, но не защитил	4	Выполнил и защитил
<i>СРС</i>	<i>10</i>		<i>20</i>	
<i>Итого за успеваемость</i>	<i>24</i>		<i>48</i>	
Посещаемость	0		16	
Зачет	0		36	
<i>Итого</i>	<i>24</i>		<i>100</i>	

Для *промежуточной аттестации обучающихся*, проводимой в виде тестирования, используется следующая методика оценивания знаний, умений, навыков и (или) опыта деятельности. В каждом варианте КИМ –16 заданий (15 вопросов и одна задача).

Каждый верный ответ оценивается следующим образом:

- задание в закрытой форме – 2 балла,
- задание в открытой форме – 2 балла,
- задание на установление правильной последовательности – 2 балла,
- задание на установление соответствия – 2 балла,
- решение компетентностно-ориентированной задачи – 6 баллов.

Максимальное количество баллов за тестирование –36 баллов.

## **8 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**



### 8.1. Основная учебная литература

1. Процедурно-модульное программирование на Delphi : учебное пособие / С. Г. Емельянов [и др.]. - Москва : Аргмак-Медиа, 2014. - 352 с. - Текст : непосредственный.
2. Фаронов, Валерий Васильевич. Delphi. Программирование на языке высокого уровня : учебник / В. В. Фаронов. - СПб. : Питер, 2012. - 640 с. - Текст : непосредственный.
3. Зотов, Игорь Валерьевич. Процедурно-ориентированное программирование на C++ : учебное пособие / И. В. Зотов, Э. И. Ватутин, Д. Б. Борзов ; Курский государственный технический университет. - Курск : КурскГТУ, 2008. - 211 с. – Текст : непосредственный.
4. Зотов, Игорь Валерьевич. Процедурно-ориентированное программирование на C++ : учебное пособие / И. В. Зотов, Э. И. Ватутин, Д. Б. Борзов ; Курский государственный технический университет. - Курск : КурскГТУ, 2008. - 211 с. – Текст : электронный.

### 8.2. Дополнительная учебная литература

5. Павловская, Т. А. С/С ++. Структурное программирование : практикум / Т. А. Павловская, Ю. А. Щюпак. - СПб. : Питер, 2007. - 239 с. - Текст : непосредственный.
6. Гудрич, М. Т. Структуры данных и алгоритмы в Java / пер. с англ. А. М. Чернухо. - Мн. : Новое знание, 2003. - 671 с. - Текст : непосредственный.
7. Белова, Т. М. Delphi: основы программирования : учебное пособие / Т. М. Белова ; Курск. гос. техн. ун-т. - Курск : КурскГТУ, 2009. - 204, [2] с. - Текст : непосредственный.
8. Белова, Т. М. Delphi: основы программирования : учебное пособие / Т. М. Белова ; Курский государственный технический университет. - Курск : КурскГТУ, 2009. - 208 с. – Текст : электронный.
9. Давыдов, В. Г. Программирование и основы алгоритмизации : учебное пособие / В. Г. Давыдов. - 2-е изд., стер. - М. : Высшая школа, 2005. - 447 с. - Текст : непосредственный.

### 8.3. Перечень методических указаний

1. Программирование линейных алгоритмов : методические указания по выполнению лабораторной работы для студентов направления подготовки 09.03.01 / Юго-Зап. гос. ун-т ; сост. Э. И. Ватутин. - Курск : ЮЗГУ, 2016. - 10 с. – Текст : электронный.
2. Программирование разветвленных алгоритмов : методические указания по выполнению лабораторной работы для студентов направления подготовки 09.03.01 / Юго-Зап. гос. ун-т ; сост. Э. И. Ватутин. - Курск : ЮЗГУ, 2016. - 10 с. - Текст : электронный.
3. Работа с массивами : методические указания по выполнению лабораторной работы для студентов направления подготовки 09.03.01 / Юго-Зап. гос. ун-т ; сост. Э. И. Ватутин. - Курск : ЮЗГУ, 2016. - 13 с. - Текст : электронный.

4. Записи и файлы : методические указания по выполнению лабораторной работы для студентов направления подготовки 09.03.01 / Юго-Зап. гос. ун-т ; сост. Э. И. Ватутин. - Курск : ЮЗГУ, 2016. - 23 с. - Текст : электронный.

5. Программирование циклических алгоритмов : методические указания по выполнению лабораторной работы для студентов направления подготовки 09.03.01 / Юго-Зап. гос. ун-т ; сост. Э. И. Ватутин. - Курск : ЮЗГУ, 2016. - 14 с. - Текст : электронный.

6. Программирование подпрограмм. Рекурсия : методические указания по выполнению лабораторной работы для студентов направления подготовки 09.03.01 / Юго-Зап. гос. ун-т ; сост. Э. И. Ватутин. - Курск : ЮЗГУ, 2016. - 26 с. - Текст : электронный.

7. Программирование на языках высокого уровня : методические указания по выполнению практических работ для студентов направления подготовки 09.03.01 / Юго-Зап. гос. ун-т ; сост. Э. И. Ватутин. - Курск : ЮЗГУ, 2016. - 22 с. - Текст : электронный.

8. Организация самостоятельной работы студентов : методические указания для студентов направлений подготовки 09.03.01 и 09.04.01 «Информатика и вычислительная техника» / Юго-Зап. гос. ун-т ; сост.: В. С. Титов, И. Е. Чернецкая, Т. А. Ширабакина. - Курск : ЮЗГУ, 2017. - 39 с. - Текст : электронный.

## **9. Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины**

1. <http://www.edu.ru/> Федеральный портал Российское образование.
2. Электронно-библиотечная система «Лань» - <http://e.lanbook.com/>
3. <http://window.edu.ru/> Электронная библиотека «Единое окно доступа к образовательным ресурсам».
4. <http://www.iqlib.ru> Электронно-библиотечная система IQLib
5. <http://www.intuit.ru/> Национальный открытый университет дистанционного образования
6. <https://ru.wikipedia.org> Википедия.

## **10 Методические указания для обучающихся по освоению дисциплины**

Основными видами аудиторной работы студента при изучении дисциплины являются лекции и лабораторные занятия. Студент не имеет права пропускать занятия без уважительных причин.

На лекциях излагаются и разъясняются основные понятия темы, связанные с ней теоретические и практические проблемы, даются рекомендации для самостоятельной работы. В ходе лекции студент должен внимательно слушать и конспектировать материал.

Изучение наиболее важных тем или разделов дисциплины завершают лабораторные занятия, которые обеспечивают: контроль подготовленности студента; закрепление учебного материала; практические занятия способствуют приобретению опыта устных публичных выступлений, ведения дискуссии, в том числе аргументации и защиты выдвигаемых положений и тезисов.

Лабораторному занятию предшествует самостоятельная работа студента, связанная с освоением материала, полученного на лекциях, и материалов, изложенных в учебниках и учебных пособиях, а также литературе, рекомендованной преподавателем.

По согласованию с преподавателем или по его заданию студенты готовят рефераты по отдельным темам дисциплины, выступают на занятиях с докладами. Основу докладов составляет, как правило, содержание подготовленных студентами рефератов.

Качество учебной работы студентов преподаватель оценивает по результатам тестирования, собеседования, защиты отчетов по лабораторным работам.

Преподаватель уже на первых занятиях объясняет студентам, какие формы обучения следует использовать при самостоятельном изучении дисциплины: конспектирование учебной литературы и лекции, составление словарей понятий и терминов и т. п.

В процессе обучения преподаватели используют активные формы работы со студентами: чтение лекций, привлечение студентов к творческому процессу на лекциях, промежуточный контроль путем отработки студентами пропущенных лекций, участие в групповых и индивидуальных консультациях (собеседовании). Эти формы способствуют выработке у студентов умения работать с учебником и литературой.

Изучение литературы составляет значительную часть самостоятельной работы студента. Это большой труд, требующий усилий и желания студента. В самом начале работы над книгой важно определить цель и направление этой работы. Прочитанное следует закрепить в памяти. Одним из приемов закрепления освоенного материала является конспектирование, без которого немислима серьезная работа над литературой. Систематическое конспектирование помогает научиться правильно, кратко и четко излагать своими словами прочитанный материал.

Самостоятельную работу следует начинать с первых занятий. От занятия к занятию нужно регулярно прочитывать конспект лекций, знакомиться с соответствующими разделами учебника, читать и конспектировать литературу по каждой теме дисциплины. Самостоятельная работа дает студентам возможность равномерно распределить нагрузку, способствует более глубокому и качественному усвоению учебного материала. В случае необходимости студенты обращаются за консультацией к преподавателю с целью усвоения и закрепления компетенций.

Основная цель самостоятельной работы студента при изучении дисциплины - закрепить теоретические знания, полученные в процессе лекционных занятий, а также сформировать практические навыки самостоятельного анализа особенностей дисциплины.

## **11 Перечень информационных технологий**

1. ОС Windows 7 (<https://www.microsoft.com>, договор ИТ 000012385).
2. LibreOffice, [ru.libreoffice.org/download/](http://ru.libreoffice.org/download/) (Бесплатная, GNU General Public License).
3. Visual Studio Community? <https://www.visualstudio.com/ru/vs/community> (Бесплатная, лицензионное соглашение).
4. NASM, <http://www.nasm.us/> (Бесплатная, FreeBSD License)

5. Lazarus, <http://www.lazarus.freepascal.org/> (Бесплатная, Freeware)

## **12 Материально-техническое обеспечение учебной дисциплины**

Учебные аудитории для проведения занятий лекционного типа и лаборатории кафедры вычислительной техники оснащены учебной мебелью: столы, стулья для обучающихся; стол, стул для преподавателя; доска; ПЭВМ INTEL Core i3-7100/H110M-R C/SI White Box LGA1151.mATX/8Gb/1TB/DVDRW/LCD 21.5"/k+m/; Многопроцессорный вычислительный комплекс; Core 2 Duo 1863/2\*DDR2 1024 Mb/2\*HDD 200G/SVGA/DVD-RW/20'LCD\*2/Secret Net; Ноутбук ASUS X50VL PMD – T2330/14"/1024 Mb/160 Gb/ сумка; Проектор in Focus IN24+, экран настенный, видеопроектор.

Стандартно оборудованные лекционные аудитории и аудитории для проведения занятий семинарского типа. Компьютерный класс оснащенный ПК ВаРИАНт PD2160/I C33/2\*512 Mb/HDD 160Gb/DVD-ROM/FDD/ATX 350W/Км/WXP/DFP/17'TFTE 700 или интерактивная панель Интерактивная панель JeminiCo. JQ75MW с ОПС модулем и мобильной стойкой; компьютер в сборе (ТИП-2) или рабочая станция Core 2 Duo 1863/2\*DDR2 1024 Mb/2\*HDD 200G/SVGA/DVD-RW/20'LCD\*2/Secret Net; ПЭВМ INTEL Gore i3-7100/H110M-R C/SI White Box LGA1151.mATX/8GB/1TB/DVDRW/LCD 21.5"/k+m/ в зависимости от предоставленной аудитории.

### **13 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

При обучении лиц с ограниченными возможностями здоровья учитываются их индивидуальные психофизические особенности. Обучение инвалидов осуществляется также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).

*Для лиц с нарушением слуха* возможно предоставление учебной информации в визуальной форме (краткий конспект лекций; тексты заданий, напечатанные увеличенным шрифтом), на аудиторных занятиях допускается присутствие ассистента, а также сурдопереводчиков и тифлосурдопереводчиков. Текущий контроль успеваемости осуществляется в письменной форме: обучающийся письменно отвечает на вопросы, письменно выполняет практические задания. Доклад (реферат) также может быть представлен в письменной форме, при этом требования к содержанию остаются теми же, а требования к качеству изложения материала (понятность, качество речи, взаимодействие с аудиторией и т. д.) заменяются на соответствующие требования, предъявляемые к письменным работам (качество оформления текста и списка литературы, грамотность, наличие иллюстрационных материалов и т.д.). Промежуточная аттестация для лиц с нарушениями слуха проводится в письменной форме, при этом используются общие критерии оценивания. При необходимости время подготовки к ответу может быть увеличено.

*Для лиц с нарушением зрения* допускается аудиальное предоставление информации, а также использование на аудиторных занятиях звукозаписывающих устройств (диктофонов и т.д.). Допускается присутствие на занятиях ассистента (помощника), оказывающего обучающимся необходимую техническую помощь. Текущий контроль успеваемости осуществляется в устной форме. При проведении промежуточной аттестации для лиц с нарушением зрения тестирование может быть заменено на устное собеседование по вопросам.

*Для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата*, на аудиторных занятиях, а также при проведении процедур текущего контроля успеваемости и промежуточной аттестации могут быть предоставлены необходимые технические средства (персональный компьютер, ноутбук или другой гаджет); допускается присутствие ассистента (ассистентов), оказывающего обучающимся необходимую техническую помощь (занять рабочее место, передвигаться по аудитории, прочитать задание, оформить ответ, общаться с преподавателем).

