

Федеральное государственное бюджетное образовательное учреждение высшего образования «Тамбовский государственный университет имени Г.Р. Державина»
Инженерно-технический институт
Кафедра функционального анализа

УТВЕРЖДАЮ:
И.о. директора института



Т. В. Пасько
«17» июня 2026 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине Б1.В.10 Программирование на языках высокого уровня

Направление подготовки/специальность: 01.03.01 - Математика

Профиль/направленность/специализация: Искусственный интеллект и моделирование

Уровень высшего образования: бакалавриат

Квалификация: Бакалавр

год набора: 2026

Авторы программы:

Рыбаков Михаил Анатольевич

Кандидат физико-математических наук, Переславцева Оксана Николаевна

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 01.03.01 - Математика (уровень бакалавриата) (приказ Министерства образования и науки РФ от «10» января 2018 г. № 8).

Рабочая программа принята на заседании Кафедры функционального анализа «16» июня 2026 г. Протокол № 11

Рассмотрена и одобрена на заседании Ученого совета Инженерно-технического института, Протокол от «17» июня 2026 г. № 10.

СОДЕРЖАНИЕ

1. Цели и задачи дисциплины.....	4
2. Место дисциплины в структуре ОП бакалавриата.....	4
3. Объем и содержание дисциплины.....	4
4. Контроль знаний обучающихся и типовые оценочные средства.....	13
5. Методические указания для обучающихся по освоению дисциплины (модуля).....	22
6. Учебно-методическое и информационное обеспечение дисциплины.....	24
7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы.....	25

1. Цели и задачи дисциплины

1.1 Цель дисциплины – формирование компетенций:

1.2 Типы задач профессиональной деятельности, к которым готовятся обучающиеся в рамках освоения дисциплины:

- научно-исследовательский
- организационно-управленческий
- проектно-технологический

1.3 Дисциплина ориентирована на подготовку обучающихся к профессиональной деятельности в сфере: 40 Сквозные виды профессиональной деятельности в промышленности (в сфере научно-исследовательских и опытноконструкторских разработок)

1.4 В результате освоения дисциплины у обучающихся должны быть сформированы:

Обобщенные трудовые функции / трудовые функции / трудовые или профессиональные действия (при наличии профстандарта)	Код и наименование компетенции ФГОС ВО, необходимой для формирования трудового или профессионального действия	Индикаторы достижения компетенций
---	---	-----------------------------------

1.5 Согласование междисциплинарных связей дисциплин, обеспечивающих освоение компетенций:

2. Место дисциплины в структуре ОП бакалавриата:

Дисциплина «Программирование на языках высокого уровня» относится к части, формируемой участниками образовательных отношений, учебного плана ОП по направлению подготовки 01.03.01 - Математика.

Дисциплина «Программирование на языках высокого уровня» изучается в 2 семестре.

3. Объем и содержание дисциплины

3.1. Объем дисциплины: 4 з.е.

Очная: 4 з.е.

Вид учебной работы	Очная (всего часов)
Общая трудоёмкость дисциплины	144
Контактная работа	48
Лекции (Лекции)	16
Лабораторные (Лаб. раб.)	32
Самостоятельная работа (СР)	60
Экзамен	36

3.2. Содержание курса:

№ темы	Название раздела/темы	Вид учебной работы, час.			Формы текущего контроля
		Лек ции	Лаб · раб.	СР	
		О	О	О	
2 семестр					

1	Обзор языка java.	1	1	3	Лабораторная работа
2	Типы данных, переменные, массивы, операции и управляющие операторы языка java	1	2	3	Лабораторная работа
3	ООП в java.	1	2	3	Лабораторная работа
4	Обработка исключений в java	1	1	3	Лабораторная работа
5	Многопоточное программирование в java	1	1	2	Лабораторная работа
6	Обработка символьных строк в java.	1	1	2	Лабораторная работа
7	Пакет java. util, Collections Framework и служебные классы	1	1	2	Лабораторная работа
8	Пакет java. io для ввода-вывода и система ввода-вывода NIO.	1	1	2	Лабораторная работа
9	Работа в сети java.	1	1	2	Лабораторная работа
10	Утилиты параллелизма java	1	1	2	Лабораторная работа
11	Потоковый API java.	-	2	2	Лабораторная работа
12	Библиотека AWT java	-	2	2	Лабораторная работа; Тестирование
13	: Программирование ГПИ средствами Swing java.	-	2	2	Лабораторная работа
14	Программирование ГПИ средствами JavaFX java.	-	2	4	Лабораторная работа
15	Сервлеты java	1	2	4	Лабораторная работа
16	Введение в программирование на языке Python, синтаксис.	1	2	4	Лабораторная работа
17	Разные стили написания программ на Python	1	2	4	Лабораторная работа

18	Технологии программирования Python	1	2	4	Лабораторная работа
19	Применение языка Python в разных областях	1	2	4	Лабораторная работа
20	Python для анализа данных.	1	2	6	Лабораторная работа

Тема 1. Обзор языка java.

Лекция.

Отличительные особенности Java. ООП Java. Ввод кода программы. Компиляция программы

Лабораторные работы.

Лабораторное занятие.

- 1 Ввод кода программы.
- 2 Компиляция программы

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

- 1 Происхождение Java.
- 2 Отличительные особенности Java.
- 3 Эволюция языка Java.
- 4 ООП Java.
- 5 Ввод кода программы.
- 6 Компиляция программы.
- 7 Использование блоков кода.
- 8 Вопросы лексики.

Тема 2. Типы данных, переменные, массивы, операции и управляющие операторы языка java

Лекция.

Примитивные типы. Целые числа. Числа с плавающей точкой. Символы. Логические значения. Переменные. Область и срок действия переменных. Преобразование и приведение типов. Одномерные массивы. Многомерные массивы. Арифметические операции. Поразрядные операции. Операции отношения. Логические операции. Операция присваивания. Операторы выбора. Операторы цикла. Операторы перехода.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java для работы с рекурсией.
2. Разработка приложения на Java для поддержки ООП.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для работы с многомерными массивами.

Тема 3. ООП в java.

Лекция.

Основы классов. Объявление объектов. Конструкторы. Ключевое слово `this`. Сборка "мусора". Метод `finalize()`. Класс `Stack`. Перегрузка методов. Перегрузка конструкторов. Применение объектов в качестве параметров. Возврат объектов. Рекурсия. Введение в управление доступом. Ключевое слово `static`. Ключевое слово `final`. Вложенные и внутренние классы. Применение аргументов командной строки. Аргументы переменной длины. Основы наследования. Ключевое слово `super`. Переопределение методов. Динамическая диспетчеризация методов. Применение абстрактных классов. Класс `Object`. Пакеты. Защита доступа. Интерфейсы. Методы по умолчанию. Применение статических методов в интерфейсе.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java для работы с рекурсией.
2. Разработка приложения на Java для поддержки ООП.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java. Создание иерархии классов.
2. Разработка приложения на Java. Создание интерфейсов.

Тема 4. Обработка исключений в java

Лекция.

Основы обработки исключений. Типы исключений. Использование блоков операторов `try` и `catch`. Применение нескольких операторов `catch`. Вложенные операторы `try`. Оператор `throw`. Оператор `throws`. Оператор `finally`. Встроенные в Java исключения. Создание собственных подклассов исключений.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java для обработки исключений.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Использование блоков операторов `try` и `catch`.
2. Создание собственных подклассов исключений

Тема 5. Многопоточное программирование в java

Лекция.

Модель потоков исполнения в java. Главный поток исполнения. Создание потока исполнения. Создание многих потоков исполнения. Применение методов `isAlive()` и `join()`. Приоритеты потоков исполнения. Синхронизация. Взаимодействие потоков исполнения.

Лабораторные работы.

Лабораторное занятие.

Разработать программы, реализующие:

1. Синхронное и асинхронное взаимодействие основного потока исполнения с дочерними.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Приоритеты потоков исполнения.
2. Семафоры, мьютексы, критические секции.

Тема 6. Обработка символьных строк в java.

Лекция.

Конструкторы символьных строк. Длина символьной строки. Специальные строковые операции. Извлечение символов. Сравнение символьных строк. Поиск в символьных строках. Видоизменение символьных строк. Преобразование данных методом `valueOf()`. Изменение регистра символов в строке. Соединение символьных строк. Дополнительные методы из класса `String`. Класс `StringBuffer`. Класс `StringBuilder`.

Лабораторные работы.**Лабораторное занятие.**

1. Разработка приложения на Java для работы с классом `StringBuffer`.
2. Разработка приложения на Java для работы с классом `StringBuilder`.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для синтаксического анализа заданного текста.

Тема 7. Пакет `java.util`, `Collections Framework` и служебные классы**Лекция.**

Краткий обзор коллекций. Интерфейсы коллекций. Классы коллекций. Доступ к коллекциям через итератор. Итераторы-разделители. Интерфейс `RandomAccess`. Обращение с отображениями. Компараторы. Алгоритмы коллекций. Массивы. Унаследованные классы и интерфейсы. Класс `StringTokenizer`. Класс `BitSet`. Класс `Date`. Класс `Calendar`. Класс `GregorianCalendar`. Класс `TimeZone`. Класс `SimpleTimeZone`. Класс `Locale`. Класс `Random`. Класс `Observable`. Класс `Timer` и `TimerTask`. Класс `Currency`. Класс `Formatter`. Класс `Scanner`. Прочие служебные классы и интерфейсы.

Лабораторные работы.**Лабораторное занятие.**

1. Разработка приложения на Java для работы с коллекциями.
2. Разработка приложения на Java для работы с алгоритмами коллекций.
3. Разработка приложения на Java для работы с временем и датой.
4. Разработка приложения на Java для работы с классом `Formatter`.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для работы с классами `Timer` и `TimerTask`.
2. Разработка приложения на Java для работы с классом `GregorianCalendar`.
3. Разработка приложения на Java для работы с классом `RandomAccess`.

Тема 8. Пакет `java.io` для ввода-вывода и система ввода-вывода NIO.**Лекция.**

Классы и интерфейсы ввода-вывода. Класс `File`. Исключения ввода-вывода. Классы потоков ввода-вывода. Потоки ввода-вывода байтов. Потоки ввода-вывода символов. Класс `Console`. Сериализация. Классы системы ввода-вывода NIO. Основные положения о системе ввода-вывода NIO. Применение системы ввода-вывода NIO.

Лабораторные работы.**Лабораторное занятие.**

1. Разработка приложения на Java для работы с потоками ввода-вывода байтов.
2. Разработка приложения на Java для работы с потоками ввода-вывода символов.
3. Разработка приложения на Java для работы с системы ввода-вывода NIO.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для работы с классом File.
2. Разработка приложения на Java для работы с классом Console.
3. Разработка приложения на Java для работы с сериализацией.

Тема 9. Работа в сети java.

Лекция.

Основы работы в сети. Сетевые классы и интерфейсы. Класс InetAddress. Классы Inet4Address и Inet6Address. Клиентские сокеты по протоколу TCP /IP. Класс URL. Класс URLConnection. Класс HttpURLConnection. Класс URI. Cookie-файлы. Серверные сокеты по протоколу TCP /IP. Дейтаграммы.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java для работы с классом InetAddress.
2. Разработка приложения на Java для работы с сокетами (клиент/сервер).

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для работы с классами Inet4Address и Inet6Address.
2. Разработка приложения на Java для работы с дейтаграммами.

Тема 10. Утилиты параллелизма java

Лекция.

Пакеты параллельного API. Применение объектов синхронизации. Класс Phaser. Применение исполнителя. Перечисление TimeUnit. Параллельные коллекции. Блокировки. Атомарные операции. Параллельное программирование средствами Fork/Join Framework.

Лабораторные работы.

1. Разработка приложения на Java для работы с объектами синхронизации.
2. Параллельное программирование средствами Fork/Join Framework.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для работы с классом Phaser.
2. Разработка приложения на Java для работы с перечислениями TimeUnit.

Тема 11. Поточковый API java.

Лекция.

Основные положения о потоках данных. Операции сведения. Параллельные потоки данных. Отображение. Накопление. Итераторы и потоки данных.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java для работы с параллельными потоками данных.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для работы с отображениями.

Тема 12. Библиотека AWT java

Лекция.

Обработка событий. Два механизма обработки событий. Модель делегирования событий. События. Источники событий. Классы событий. Источники событий. Интерфейсы приемников событий. Применение модели делегирования событий. Обработка событий от мыши. Обработка событий от клавиатуры. Классы адаптеров. Классы библиотеки AWT. Основные положения об окнах. Работа с обрамляющими окнами. Обработка событий в обрамляющем окне. Создание оконной прикладной программы. Отображение информации в окне. Работа с цветом. Работа со шрифтами. Управление форматированием выводимого текста. Основные положения об элементах управления. Ввод и удаление элементов управления. Метки. Экранные кнопки. Флажки. Кнопки-переключатели. Элементы управления выбором. Использование списков. Управление полосами прокрутки. Текстовые поля. Текстовые области. Диспетчеры компоновки. Меню и строки меню. Диалоговые окна.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java. Обработка событий от мыши.
2. Разработка приложения на Java. Обработка событий от клавиатуры.
3. Разработка приложения на Java. Работа с цветом и шрифтом.
4. Разработка приложения на Java. Элементы управления.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java. Диспетчеры компоновки.
2. Разработка приложения на Java. Меню и строки меню.
3. Разработка приложения на Java. Диалоговые окна выбора файлов.
4. Разработка приложения на Java. Основы работы с изображениями: создание, загрузка и отображение.

Тема 13. : Программирование ГПИ средствами Swing java.

Лекция.

Не предусмотрена

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java для работы с классом JList.
2. Разработка приложения на Java для работы с классом JComboBox.
3. Разработка приложения на Java для работы с классом JTable.
4. Разработка приложения на Java для работы с классом JTree.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java для работы с классом JMenuBar.
2. Разработка приложения на Java для работы с классом JMenu.
3. Разработка приложения на Java для работы с классом JMenuItem.
4. Разработка приложения на Java. Создание всплывающего меню.
5. Разработка приложения на Java. Создание панели инструментов

Тема 14. Программирование ГПИ средствами JavaFX java.

Лекция.

Лекция. Не предусмотрена.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java для работы с классом RadioButton JavaFX.

2. Разработка приложения на Java для работы с классом CheckBox JavaFX.
3. Разработка приложения на Java для работы с классом ListView JavaFX.
4. Разработка приложения на Java для работы с классом ComboBox JavaFX.
5. Разработка приложения на Java для работы с классом TextField JavaFX.
6. Разработка приложения на Java для работы с классом TreeView JavaFX.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java. Эффекты и преобразования.
2. Разработка приложения на Java для работы с классами RadioMenuItem и CheckMenuItem JavaFX.
3. Разработка приложения на Java. Создание контекстного меню JavaFX.
4. Разработка приложения на Java. Создание панели инструментов JavaFX.

Тема 15. Сервлеты java

Лекция.

Не предусмотрена

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Java. Запуск веб-браузера и запрос сервлета.
2. Разработка приложения на Java. Прикладной программный интерфейс Servlet API.
3. Разработка приложения на Java. Ввод параметров сервлета.
4. Разработка приложения на Java. Пакет javax.servlet.http.
5. Разработка приложения на Java. Обработка HTTP -запросов и ответов HTTP.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Разработка приложения на Java. Применение cookie-файлов.
2. Разработка приложения на Java. Отслеживание сеансов связи.

Тема 16. Введение в программирование на языке Python, синтаксис.

Лекция.

Лекция. Не предусмотрена.

Лабораторные работы.

Лабораторное занятие.

1. Разработка приложения на Python. Синтаксис языка, управляющие структуры.
2. Разработка приложения на Python. Типы данных.
3. Разработка приложения на Python. Списки, кортежи, словари, юникод, даты

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Установки на различные платформы.
2. Регулярные выражения.
3. Генераторы, итераторы, декораторы методов и классов.

Тема 17. Разные стили написания программ на Python

Лекция.

Лекция. Не предусмотрена.

Лабораторные работы.

Лабораторное занятие.

1. Процедурное программирование на Python.
2. ООП на Python.
3. Функциональное программирование на Python.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Программирование в ограничениях, декларативное программирование на Python.

Тема 18. Технологии программирования Python**Лекция.**

Лекция. Не предусмотрена.

Лабораторные работы.**Лабораторное занятие.**

1. Цикл разработки.
2. Профилирование, отладка, тестирование.
3. Модули.

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Интеграция с другими ЯП.
2. Создание своих модулей и пакетов.
3. Параллельное программирование.

Тема 19. Применение языка Python в разных областях**Лекция.**

Лекция. Не предусмотрена

Лабораторные работы.**Лабораторное занятие.**

1. Python и математика.
2. Python и программы с GUI.
3. Python, web и xml.
4. Web-программирование на Python

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Python и серверный скриптинг (автоматизации администрирования серверов).
2. Python под разные платформы.

Тема 20. Python для анализа данных.**Лекция.**

Лекция. Не предусмотрена

Лабораторные работы.**Лабораторное занятие.**

1. Эффективные библиотеки работы с данными на Python: numpy, pandas.
2. Машинное обучение на Python: sk-learn.
3. Визуализация данных на Python

Задания для самостоятельной работы.

Подготовка к опросу.

Задачи:

1. Эффективные библиотеки работы с данными на Python: numpy, pandas.
2. Машинное обучение на Python: sk-learn.
3. Визуализация данных на Python.

4. Контроль знаний обучающихся и типовые оценочные средства

4.1. Распределение баллов:

2 семестр

- посещаемость – 5 баллов
- текущий контроль – 75 баллов
- контрольные срезы – 2 среза по 10 баллов каждый
- премиальные баллы – 10 баллов

Распределение баллов по заданиям:

№ те мы	Название темы / вид учебной работы	Формы текущего контроля / срезы	Мах. кол-во баллов	Методика проведения занятия и оценки
1.	Обзор языка java.	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов.
2.	Типы данных, переменные, массивы, операции и управляющие операторы языка java	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов.
3.	ООП в java.	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов.
4.	Обработка исключений в java	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов.
5.	Многопоточное программирование в java	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов.
6.	Обработка символьных строк в java.	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов
7.	Пакет java. util, Collections Framework и служебные классы	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 10 баллов
8.	Пакет java. io для ввода-вывода и система ввода-вывода NIO.	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов

9.	Работа в сети java.	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 8 баллов
10.	Утилиты параллелизма java	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 6 баллов
11.	Потоковый API java.	Лабораторная работа	4	лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 5 баллов
12.	Библиотека AWT java	Лабораторная работа	4	лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студент получает 5 баллов
		Тестирование(контрольный срез)	10	В случае правильных ответов на 51% заданий тестирования студент получает 10 баллов.
13.	Программирование ГПИ средствами Swing java.	Лабораторная работа(контрольный срез)	10	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 10 баллов
14.	Программирование ГПИ средствами JavaFX java.	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 10 баллов
15.	Сервлеты java	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 10 баллов.
16.	Введение в программирование на языке Python, синтаксис.	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 6 баллов.
17.	Разные стили написания программ на Python	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 6 баллов
18.	Технологии программирования Python	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 6 баллов
19.	Применение языка Python в разных областях	Лабораторная работа	4	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 6 баллов
20.	Python для анализа данных.	Лабораторная работа	3	Выполнение и защита лабораторной работы. В случае успешного выполнения всех заданий лабораторной работы студенту выставляются 6 баллов
21.	Посещаемость		5	5 баллов за отсутствие пропусков занятий без уважительных причин и выполнение в срок всех заданий
22.	Премиальные баллы		10	10 баллов за участие в студенческих научных конференциях и олимпиадах
23.	Индивидуальные задания, с помощью которых можно набрать дополнительные баллы		20	20 баллов за выполнение индивидуальных заданий
24.	Итого за семестр		100	

Итоговая оценка по экзамену выставляется в 100-балльной шкале и в традиционной четырехбалльной шкале. Перевод 100-балльной рейтинговой оценки по дисциплине в традиционную четырехбалльную осуществляется следующим образом:

100-балльная система	Традиционная система
85 - 100 баллов	Отлично
70 - 84 баллов	Хорошо
50 - 69 баллов	Удовлетворительно
Менее 50	Неудовлетворительно

4.2 Типовые оценочные средства текущего контроля

Лабораторная работа

Тема 1. Обзор языка java.

Задания к лабораторной работе по теме №1 " Обзор языка java ".

- 1 Ввод кода программы.
- 2 Компиляция программы.

Тема 2. Типы данных, переменные, массивы, операции и управляющие операторы языка java

Задания к лабораторной работе по теме №2 " Типы данных, переменные, массивы, операции и управляющие операторы языка java ".

1. Разработка приложения на Java для работы с рекурсией.
2. Разработка приложения на Java для поддержки ООП.

Тема 3. ООП в java.

Задания к лабораторной работе по теме №3 "ООП в java".

1. Разработка приложения на Java для работы с рекурсией.
2. Разработка приложения на Java для поддержки ООП.

Тема 4. Обработка исключений в java

Задания к лабораторной работе по теме №4 " Обработка исключений в java ".

1. Разработка приложения на Java для обработки исключений.

Тема 5. Многопоточное программирование в java

Задания к лабораторной работе по теме №5 "Многопоточное программирование в java".

Разработать программы, реализующие:

1. Синхронное и асинхронное взаимодействие основного потока исполнения с дочерними.

Тема 6. Обработка символьных строк в java.

Задания к лабораторной работе по теме №6 " Обработка символьных строк в java ".

1. Разработка приложения на Java для работы с классом StringBuffer.
2. Разработка приложения на Java для работы с классом StringBuilder.

Тема 7. Пакет java. util, Collections Framework и служебные классы

Задания к лабораторной работе по теме №7 "Пакет java. util, Collections Framework и служебные классы".

1. Разработка приложения на Java для работы с коллекциями.
2. Разработка приложения на Java для работы с алгоритмами коллекций.

3. Разработка приложения на Java для работы с временем и датой.
4. Разработка приложения на Java для работы с классом Formatter.

Тема 8. Пакет java. io для ввода-вывода и система ввода-вывода NIO.

Задания к лабораторной работе по теме №8 "Пакет java. io для ввода-вывода и система ввода-вывода NIO".

1. Разработка приложения на Java для работы с потоками ввода-вывода байтов.
2. Разработка приложения на Java для работы с потоками ввода-вывода символов.
3. Разработка приложения на Java для работы с системы ввода-вывода NIO.

Тема 9. Работа в сети java.

Задания к лабораторной работе по теме №9 "Работа в сети java".

1. Разработка приложения на Java для работы с классом InetAddress.
2. Разработка приложения на Java для работы с сокетами (клиент/сервер).

Тема 10. Утилиты параллелизма java

Задания к лабораторной работе по теме №10 "Утилиты параллелизма java".

1. Разработка приложения на Java для работы с объектами синхронизации.
2. Параллельное программирование средствами Fork/Join Framework.

Тема 11. Поточковый API java.

Задания к лабораторной работе по теме №11 "Поточковый API java".

1. Разработка приложения на Java для работы с параллельными потоками данных.

Тема 12. Библиотека AWT java

Задания к лабораторной работе по теме №12 "Библиотека AWT java".

1. Разработка приложения на Java. Обработка событий от мыши.
2. Разработка приложения на Java. Обработка событий от клавиатуры.
3. Разработка приложения на Java. Работа с цветом и шрифтом.
4. Разработка приложения на Java. Элементы управления

Тема 13. : Программирование ГПИ средствами Swing java.

Задания к лабораторной работе по теме №13 "Программирование ГПИ средствами Swing java".

1. Разработка приложения на Java для работы с классом JList.
2. Разработка приложения на Java для работы с классом JComboBox.
3. Разработка приложения на Java для работы с классом JTable.
4. Разработка приложения на Java для работы с классом JTree.

Тема 14. Программирование ГПИ средствами JavaFX java.

Задания к лабораторной работе по теме №14 "Программирование ГПИ средствами JavaFX java".

1. Разработка приложения на Java для работы с классом RadioButton JavaFX.
2. Разработка приложения на Java для работы с классом CheckBox JavaFX.
3. Разработка приложения на Java для работы с классом ListView JavaFX.
4. Разработка приложения на Java для работы с классом ComboBox JavaFX.
5. Разработка приложения на Java для работы с классом TextField JavaFX.
6. Разработка приложения на Java для работы с классом TreeView JavaFX.

Тема 15. Сервлеты java

Задания к лабораторной работе по теме №15 "Сервлеты java".

1. Разработка приложения на Java. Запуск веб-браузера и запрос сервлета.
2. Разработка приложения на Java. Прикладной программный интерфейс Servlet API.
3. Разработка приложения на Java. Ввод параметров сервлета.
4. Разработка приложения на Java. Пакет javax.servlet.http.
5. Разработка приложения на Java. Обработка HTTP -запросов и ответов HTTP.

Тема 16. Введение в программирование на языке Python, синтаксис.

Задания к лабораторной работе по теме №16 "Введение в программирование на языке Python, синтаксис".

1. Разработка приложения на Python. Синтаксис языка, управляющие структуры.
2. Разработка приложения на Python. Типы данных.
3. Разработка приложения на Python. Списки, кортежи, словари, юникод, даты.

Тема 17. Разные стили написания программ на Python

Задания к лабораторной работе по теме №17 "Разные стили написания программ на Python".

1. Процедурное программирование на Python.
2. ООП на Python.
3. Функциональное программирование на Python.

Тема 18. Технологии программирования Python

Задания к лабораторной работе по теме №18 "Технологии программирования Python".

1. Цикл разработки.
2. Профилирование, отладка, тестирование.
3. Модули.

Тема 19. Применение языка Python в разных областях

Задания к лабораторной работе по теме №19 "Применение языка Python в разных областях".

1. Python и математика.
2. Python и программы с GUI.
3. Python, web и xml.
4. Web-программирование на Python.

Тема 20. Python для анализа данных.

Задания к лабораторной работе по теме №20 "Python для анализа данных".

1. Эффективные библиотеки работы с данными на Python: numpy, pandas.
2. Машинное обучение на Python: sk-learn.
3. Визуализация данных на Python.

Тестирование

Тема 12. Библиотека AWT java

Тестовые задания

1. Дан код: `public class Quest1 {public static void main(String[] args) {String str = new String("java"); int i=1; char j=3; System.out.println(str.substring(i,j)); }}`

В результате при компиляции и запуске будет выведено:

1) ja; 2) av; 3) ava; 4) jav; 5) ошибка компиляции: заданы некорректные параметры для метода substring().

2. Какую инструкцию следует использовать, чтобы обнаружить позицию буквы v в строке str = "Java"?

1) charAt(2,str); 2) str.charAt(2); 3) str.indexOf('v'); 4) indexOf(str,'v');

3. Какие из следующих операций корректны при объявлении?

String s = new String("Java");

String t = new String();

String r = null;

1) r = s + t + r; 2) r = s + t + 'r'; 3) r = s & t & r; 4) r = s && t && r;

4. Дан код: public class Quest4 { public static void main(String[] args) { String str="ava"; char ch=0x74; // 74 - это код символа 'J' str=ch+str; System.out.print(str); }}

В результате при компиляции и запуске будет выведено:

1) 74ava; 2) Java; 3) 0H74ava; 4) ошибка компиляции: недопустимая операция ch+str; 5) ошибка компиляции: недопустимое объявление char ch=0x74; 6) нет правильного ответа.

5. Что будет результатом компиляции и выполнения следующего кода?

public class Quest5 { public static void main(String[] args) { StringBuffer s = new StringBuffer("you java"); s.insert(2, "like "); System.out.print(s); }}

1) yolike u java; 2) you like java; 3) ylike ou java; 4) you java like; 5) ошибка компиляции; 6) нет правильного ответа.

6. Дан код: class Quest1 {int counter; java.io.OutputStream out; Quest1() { /* инициализация out */ float inc() { try { counter++; out.write(counter); } //комментарий }}

Какой код достаточно добавить в метод inc() вместо комментария, чтобы компиляция прошла без ошибок? (выберите 2).

1) catch (java.io.OutputStreamException e){}; 2) catch (java.io.IOException e){}; 3) catch (java.io.OutputException e){}; 4) finally{}; 5) return counter; 6) return;.

7. Какое значение будет возвращено при вызове meth(5)?

public int meth(int x) { int y = 010; //1 try { y += x; //2 if(x<=5) throw new Exception(); //3 y++; } //4 catch(Exception e) { y--; } //5 return y; } //6

1) 12; 2) 13; 3) 14; 4) 15; 5) ошибка компиляции: невыполнимый код в строке 4.

8. Какое значение будет возвращено при вызове meth(12), если при вызове mexcept(int x) возникает исключительная ситуация ArithmeticException?

int meth(int x) { int count=0; try { count += x; count += mexcept(count); count++; } catch(Exception e) { --count; return count; } finally { count += 3; return count; } }

1) 11; 2) 12; 3) 13; 4) 14; 5) ошибка компиляции из-за отсутствия return после блока finally.

9. Какое из следующих определений метода show() может законно использоваться вместо комментария //КОД в классе Quest4?

class Base {public void show(int i) { /*реализация*/ } } public class Quest4 extends Base { //КОД }

1) void show (int i) throws Exception { /*реализация*/ } 2) void show (long i) throws IOException { /*реализация*/ } 3) void show (short i) { /*реализация*/ } 4) public void show (int i) throws IOException { /*реализация*/ }

10. Дан код: import java.io.*;

public class Quest5 { //ОБЪЯВЛЕНИЕ ioRead() public static void main(String[] args) { try { ioRead(); } catch(IOException e) { } } }

Какое объявление метода ioRead() должно быть использовано вместо комментария, чтобы компиляция и выполнение кода прошли успешно?

1) private static void ioRead() throws IOException{}; 2) public static void ioRead() throw IOException{}; 3) public static void ioRead(){}; 4) public static void ioRead() throws Exception{}.

11. Дан код: `class Q implements Runnable{int i = 0; public int run(){ System.out.println("i = "+ ++i); return i; }} public class Quest1 { public static void main(String[] args) { Q ob = new Q(); ob.run(); }}`

При попытке компиляции и запуска будет выведено:

- 1) i = 0; 2) i = 1; 3) ошибка компиляции: создать объект потока можно только с помощью инициализации объекта класса Thread или его наследников;
- 4) ошибка компиляции: неправильно реализован метод run();
- 5) ошибка времени выполнения: поток должен запускаться методом start().

12. Дан код: `Thread t1=new Thread(); t1.setPriority(7); ThreadGroup tg=new ThreadGroup("TG"); tg.setMaxPriority(8); Thread t2=new Thread(tg,"t2"); System.out.print("приоритет t1=" + t1.getPriority()); System.out.print(", приоритет t2=" + t2.getPriority());`

В результате компиляции и выполнения будет выведено:

- 1) приоритет t1 = 7, приоритет t2 = 5; 2) приоритет t1 = 7, приоритет t2 = 8; 3) приоритет t1 = 10, приоритет t2 = 8; 4) нет правильного.

13. Дан код: `class T1 implements Runnable{ public void run(){ System.out.print("t1 "); } }class T2 extends Thread{ public void run(){ System.out.print("t2 "); } } public class Quest3 { public static void main(String[] args) { T1 t1 = new T1(); T2 t2 = new T2(t1); t1.start(); t2.start(); } }`

В результате компиляции и запуска будет выведено:

- 1) t1 t2; 2) t2 t1; 3) ошибка компиляции: метод start() не определен в классе T1;
- 4) ошибка компиляции: в классе T2 не определен конструктор, принимающий в качестве параметра объект Thread; 5) ничего из перечисленного.

14. Какое из указанных действий приведет к тому, что поток переходит в состояние “пассивный”? (выберите 2)

- 1) вызов метода sleep() без параметра; 2) вызов метода stop(); 3) окончание выполнения метода run(); 4) вызов метода notifyAll(); 5) вызов метода wait() с параметром null.

15. Дан код: `class Quest5 extends Thread {Quest5 () { } Quest5 (Runnable r) { super(r); } public void run() { System.out.print("thread "); } public static void main(String[] args) { Runnable r = new Quest5(); //1 Quest5 t = new Quest5(r); //2 t.start(); } }`

При попытке компиляции и запуска будет выведено:

- 1) ошибка компиляции в строке //1; 2) ошибка компиляции в строке //2; 3) thread;
- 4) thread thread; 5) код будет откомпилирован, но ничего выведено не будет.

16. Какой интерфейс наиболее пригоден для создания класса, содержащего несортированные уникальные объекты? 1) Set; 2) List; 3) Map; 4) Vector; 5) нет правильного ответа.

17. Какие из фрагментов кода создадут объект класса ArrayList и добавят элемент?

- 1) `ArrayList a = new ArrayList(); a.add("0");`
- 2) `ArrayList a = new ArrayList(); a[0]="0";`
- 3) `List a = new List(); a.add("0");`
- 4) `List a = new ArrayList(10); a.add("0");`

18. Какой интерфейс реализует класс Hashtable?

- 1) Set; 2) Vector; 3) AbstractMap; 4) List; 5) Map.

19. Дан код:

```
import java.util.*;
class Quest4 {
public static void main (String args[]) {
Object ob = new HashSet();
System.out.print((ob instanceof Set) + ", ");
System.out.print(ob instanceof SortedSet);}}
```

Что будет выведено при попытке компиляции и запуска программы?

- 1) true, false; 2) true, true; 3) false, true; 4) false, false; 5) ничего из перечисленного.

20. Какие из приведенных ниже названий являются именами интерфейсов пакета java.util?

1) SortedMap; 2) HashMap; 3) HashSet; 4) SortedSet; 5) Stack; 6) AbstractMap.

21. Выбрать необходимое условие принадлежности класса к апплетам?

- 1) класс – наследник класса Applet при отсутствии метода main();
- 2) класс – наследник класса Applet или его подкласса;
- 3) класс – наследник класса Applet с переопределенным методом paint();
- 4) класс – наследник класса Applet с переопределенным методом init();
- 5) класс – наследник класса Applet, и все его методы объявлены со спецификатором доступа public.

22. Дан код:

```
import java.awt.*;
public class Quest2 extends Frame {
    public static void main(String[] args) {
        Quest2 fr = new Quest2();
        fr.setSize(222, 222);
        fr.setVisible(true); } }
```

Как сделать поверхность фрейма белой?

- 1) fr.setBackground(Color.white); 2) fr.setColor(Color.white); 3) fr.Background(Color.white); 4) fr.color=Color.White; 5) fr.setColor(0,0,0).

23. Что произойдет при попытке компиляции и запуска следующего кода?

```
import java.awt.*;
import java.awt.event.*;
public class Quest3 extends Frame
    implements WindowListener {
    public Quest3() {
        setSize(300,300);
        setVisible(true); }
    public void windowClosing(WindowEvent e) {
        System.exit(0); }
    public static void main(String args[]) {
        Quest3 q = new Quest3(); } }
```

- 1) ошибка компиляции;
- 2) компиляция и запуск с выводом пустого фрейма;
- 3) компиляция без запуска;
- 4) ошибка времени выполнения.

24. Какие из приведенных классов являются классами-адаптерами?

- 1) WindowAdapter; 2) WindowsAdapter; 3) AdjustmentAdapter; 4) ItemAdapter; 5) FocusAdapter.

25. Выберите из предложенных названий интерфейсы EventListener.

- 1) MouseMotionListener; 2) WindowListener; 3) KeyTypedListener; 4) ItemsListener.

26. Каким способом будет подключен объект socket, если он объявлен следующим образом

```
Socket socket = new Socket("host", 23);
```

- 1) POP3; 2) FTP; 3) TCP/IP; 4) IPX; 5) UDP.

27. Как получить содержимое страницы, используя его URL при следующем объявлении?

```
String url = new String("http://bsu.iba.by");
```

- 1) Socket content = new Socket(new URL(url)).connect();
- 2) Object content = new URL(url).getContent();
- 3) String content = new URLHttp(url).getString();
- 4) Object content = new URLConnection(url).getContent();
- 5) String content = new URLConnection(url).connect().

28. С помощью какого метода можно получить содержимое страницы по определенному адресу в сети Интернет?

1) getDocumentBase(); 2) getCodeBase(); 3) getURLAddress(); 4) getCodeAddress(); 5) getURLBase().

29. Какие исключительные ситуации возможны при открытии сокетного соединения вида:

Socket s = new Socket("bsu.iba.by", 8080);

1) IOException; 2) MalformedURLException; 3) UnknownHostException; 4) UnknownURLErrorException;
5) UnknownPortException.

30. Дан код:

Socket s = null;

ServerSocket server = new ServerSocket(8080);

s = server.accept();

PrintStream p = new PrintStream(s.getOutputStream());

p.print("привет!");

Как поместить сообщение "привет!" в сокет и дать указание закрыть сокетное соединение после передачи информации?

1) p.flush(); 2) p.close(); 3) s.flush(); 4) s.close(); 5) нет правильного.

4.3 Промежуточная аттестация по дисциплине проводится в форме экзамена

Типовые вопросы экзамена

- 1 Происхождение библиотеки Swing. Главные особенности библиотеки
- 2 Компоненты и контейнеры. Пакеты библиотеки Swing.
- 3 Рисование средствами Swing.
- 4 Классы JLabel и ImageIcon. Класс JTextField.
- 5 Кнопки из библиотеки Swing. Флажки. Кнопки-переключатели.
- 6 Класс JTabbedPane. Класс JScrollPane.
- 7 Класс JList. Класс JComboBox.
- 8 Деревья. Класс JTable.
- 9 Класс JMenuBar. Класс JMenu. Класс JMenuItem.
- 10 Ввод мнемоники и оперативных клавиш в меню.
- 11 Создание всплывающего меню.
- 12 Создание панели инструментов.
- 13 Основные понятия JavaFX. Скелет JavaFX-приложения.
- 14 Компиляция и выполнение JavaFX-приложения. Основы обработки событий в JavaFX.
- 15 Рисование непосредственно на холсте.
- 16 Классы Image и ImageView JavaFX. Класс ToggleButton JavaFX.
- 17 Класс RadioButton JavaFX. Класс CheckBox JavaFX.
- 18 Класс ListView JavaFX. Класс ComboBox JavaFX.
- 19 Класс TextField JavaFX. Класс ScrollPane JavaFX. Класс TreeView JavaFX.
- 20 Эффекты и преобразования. Класс MenuBar JavaFX.
- 21 Класс Menu JavaFX. Класс JMenuItem JavaFX.
- 22 Создание главного меню JavaFX.
- 23 Ввод мнемоники и оперативных клавиш в меню JavaFX.
- 24 Ввод изображений в пункты меню JavaFX. Классы RadioMenuItem и CheckMenuItem JavaFX.
- 25 Создание контекстного меню JavaFX.
- 26 Создание панели инструментов JavaFX.
- 27 Предпосылки для разработки сервлетов. Жизненный цикл сервлета.
- 28 Варианты разработки сервлетов. Применение контейнера сервлетов Tomcat. Создание и компиляция исходного кода сервлета.
- 29 Запуск контейнера сервлетов Tomcat на выполнение. Запуск веб-браузера и запрос сервлета.

- 30 Прикладной про граммный интерфейс Servlet API.
- 31 Ввод параметров сервлета.
- 32 Пакет javax.servlet.http.
- 33 Обработка HTTP -запросов и ответов HTTP. Применение cookie-файлов. Отслеживание сеансов связи.
- 34 Области применения Python. Основы языка Python. Ввод-вывод информации. Типы данных в Python.
- 35 Операции ветвления в Python. Операции выбора в Python. Циклы с предусловием и постусловием в Python. Цикл for в Python.
- 36 Массивы в Python. Алгоритмы поиска и сортировки в Python. Строки в Python. Функции пользователя в Python.
- 37 Перегрузка функций в Python.Исключения в Python.Основные алгоритмические конструкции языка Python.
- 38 Основные встроенные структуры данных в языке Python. Получение автономного исполняемого файла из Python-программы
- 39 Лямбда-функция.
- 40 ООП в Python. Модули стандартной библиотеки языка Python. Регулярные выражения в Python.

Типовые задания для экзамена

- 1 Создать приложение GUI SWING JAVA в котором пользователь указывает путь до существующего каталога файловой системы, и по нажатию кнопки построить/отобразить, данный каталог трансформируется в компонент JTree. С возможностью просмотра свойств каждого элемента файловой системы по событию выбора элемента.
- 2 Создать приложение GUI SWING JAVA. Изобразить в окне приложения гармонические колебания точки вдоль некоторого горизонтального отрезка. Если длина отрезка равна q , то расстояние от точки до левого конца в момент времени t можно считать равным $q(1 + \cos(wt))/2$, где w – некоторая константа. Предусмотреть поля для ввода указанных величин и кнопку для остановки и пуска процесса.
- 3 Описать графом городскую маршрутную сеть и составить правила поиска оптимального выбора маршрутов общественного транспорта для поездки между двумя заданными пунктами на Python.

4.4. Шкала оценивания промежуточной аттестации

Оценка	Компетенции	Дескрипторы (уровни) – основные признаки освоения (показатели достижения результата)
«отлично» (85 - 100 баллов)		
«хорошо» (70 - 84 баллов)		
«удовлетворительно» (50 - 69 баллов)		
«неудовлетворительно» (менее 50 баллов)		

5. Методические указания для обучающихся по освоению дисциплины (модуля)

5.1 Методические указания по организации самостоятельной работы обучающихся:

Приступая к изучению дисциплины, в первую очередь обучающимся необходимо ознакомиться содержанием рабочей программы дисциплины (РПД), которая определяет содержание, объем, а также порядок изучения и преподавания учебной дисциплины, ее раздела, части.

Для самостоятельной работы важное значение имеют разделы «Объем и содержание дисциплины», «Учебно-методическое и информационное обеспечение дисциплины» и «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы».

В разделе «Объем и содержание дисциплины» указываются все разделы и темы изучаемой дисциплины, а также виды занятий и планируемый объем в академических часах.

В разделе «Учебно-методическое и информационное обеспечение дисциплины» указана рекомендуемая основная и дополнительная литература.

В разделе «Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы» содержится перечень профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины.

5.2 Рекомендации обучающимся по работе с теоретическими материалами по дисциплине

При изучении и проработке теоретического материала необходимо:

- просмотреть еще раз презентацию лекции в системе MOODLe, повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной дополнительной литературы;
- при самостоятельном изучении теоретической темы сделать конспект, используя рекомендованные в РПД источники, профессиональные базы данных и информационные справочные системы;
- ответить на вопросы для самостоятельной работы, по теме представленные в пункте 3.2 РПД.
- при подготовке к текущему контролю использовать материалы фонда оценочных средств (ФОС).

5.3 Рекомендации по работе с научной и учебной литературой

Работа с основной и дополнительной литературой является главной формой самостоятельной работы и необходима при подготовке к устному опросу на семинарских занятиях, к дебатам, тестированию, экзамену. Она включает проработку лекционного материала и рекомендованных источников и литературы по тематике лекций.

Конспект лекции должен содержать реферативную запись основных вопросов лекции, в том числе с опорой на размещенные в системе MOODLe презентации, основных источников и литературы по темам, выводы по каждому вопросу. Конспект может быть выполнен в рамках распечатки выдачи презентаций лекций или в отдельной тетради по предмету. Он должен быть аккуратным, хорошо читаемым, не содержать не относящуюся к теме информацию или рисунки.

Конспекты научной литературы при самостоятельной подготовке к занятиям должны содержать ответы на каждый поставленный в теме вопрос, иметь ссылку на источник информации с обязательным указанием автора, названия и года издания используемой научной литературы. Конспект может быть опорным (содержать лишь основные ключевые позиции), но при этом позволяющим дать полный ответ по вопросу, может быть подробным. Объем конспекта определяется самим студентом.

В процессе работы с основной и дополнительной литературой студент может:

- делать записи по ходу чтения в виде простого или развернутого плана (создавать перечень основных вопросов, рассмотренных в источнике);
- составлять тезисы (цитирование наиболее важных мест статьи или монографии, короткое изложение основных мыслей автора);
- готовить аннотации (краткое обобщение основных вопросов работы);
- создавать конспекты (развернутые тезисы).

5.4. Рекомендации по подготовке к отдельным заданиям текущего контроля

Собеседование предполагает организацию беседы преподавателя со студентами по вопросам практического занятия с целью более обстоятельного выявления их знаний по определенному разделу, теме, проблеме и т.п. Все члены группы могут участвовать в обсуждении, добавлять информацию, дискутировать, задавать вопросы и т.д.

Устный опрос может применяться в различных формах: фронтальный, индивидуальный, комбинированный. Основные качества устного ответа подлежащего оценке:

- правильность ответа по содержанию;

- полнота и глубина ответа;
- сознательность ответа;
- логика изложения материала;
- рациональность использованных приемов и способов решения поставленной учебной задачи;
- своевременность и эффективность использования наглядных пособий и технических средств при ответе;
- использование дополнительного материала;
- рациональность использования времени, отведенного на задание.

Устный опрос может сопровождаться презентацией, которая подготавливается по одному из вопросов практического занятия. При выступлении с презентацией необходимо обращать внимание на такие моменты как:

- содержание презентации: актуальность темы, полнота ее раскрытия, смысловое содержание, соответствие заявленной темы содержанию, соответствие методическим требованиям (цели, ссылки на ресурсы, соответствие содержания и литературы), практическая направленность, соответствие содержания заявленной форме, адекватность использования технических средств учебным задачам, последовательность и логичность презентуемого материала;
- оформление презентации: объем (оптимальное количество), дизайн (читаемость, наличие и соответствие графики и анимации, звуковое оформление, структурирование информации, соответствие заявленным требованиям), оригинальность оформления, эстетика, использование возможности программной среды, соответствие стандартам оформления;
- личностные качества: ораторские способности, соблюдение регламента, эмоциональность, умение ответить на вопросы, систематизированные, глубокие и полные знания по всем разделам программы;
- содержание выступления: логичность изложения материала, раскрытие темы, доступность изложения, эффективность применения средств ИКТ, способы и условия достижения результативности и эффективности для выполнения задач своей профессиональной или учебной деятельности, доказательность принимаемых решений, умение аргументировать свои заключения, выводы.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Основная литература:

1. Федоров Д. Ю. Программирование на языке высокого уровня Python : Учебное пособие для вузов. - пер. и доп; 2-е изд.. - Москва: Юрайт, 2020. - 161 с. - Текст : электронный // ЭБС «ЮРАЙТ» [сайт]. - URL: <https://urait.ru/bcode/454100>
2. Гниденко И. Г., Павлов Ф. Ф., Федоров Д. Ю. Технологии и методы программирования : Учебное пособие для вузов. - Москва: Юрайт, 2020. - 235 с. - Текст : электронный // ЭБС «ЮРАЙТ» [сайт]. - URL: <https://urait.ru/bcode/450999>
3. Искусство программирования : [Пер. с англ.], Т.1: Основные алгоритмы. - [3-е изд.]. - М. и др.: Вильямс, 2005. - 712 с.
4. Искусство программирования : [Пер. с англ.], Т.2: Получисленные алгоритмы. - [3-е изд.]. - М. и др.: Вильямс, 2004. - 828 с.
5. Искусство программирования : [Пер. с англ.], Т.3: Сортировка и поиск. - 2-е изд.. - М. и др.: Вильямс, 2005. - 822 с.

6.2 Дополнительная литература:

1. Сузи, Р. А. Язык программирования Python : учебное пособие. - 2022-07-28; Язык программирования Python. - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. - 350 с. - Текст : электронный // IPR BOOKS [сайт]. - URL: <http://www.iprbookshop.ru/97589.html>

6.3 Иные источники:

1. Математическое программирование - <http://znanium.com/catalog.php?bookinfo=415097>

2. Национальный Открытый Университет «ИНТУИТ» - <http://www.intuit.ru/>
3. Программирование - это просто - <http://easyprog.ru>
4. СКА MahtPartner - <http://mathpar.cloud.unihub.ru/>
5. <http://docs.python.org/> - <http://docs.python.org/>

7. Материально-техническое обеспечение дисциплины, программное обеспечение, профессиональные базы данных и информационные справочные системы

Для проведения занятий по дисциплине необходимо следующее материально-техническое обеспечение: учебные аудитории для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы.

Учебные аудитории и помещения для самостоятельной работы укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы укомплектованы компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду Университета.

Для проведения занятий лекционного типа используются наборы демонстрационного оборудования, обеспечивающие тематические иллюстрации (проектор, ноутбук, экран/ интерактивная доска).

Лицензионное и свободно распространяемое программное обеспечение:

LibreOffice

Microsoft Office Профессиональный плюс 2007

Microsoft Windows 10

Операционная система "Альт Образование"

Профессиональные базы данных и информационные справочные системы:

1. Научная электронная библиотека eLIBRARY.ru. – URL: <https://elibrary.ru>
2. Springer Journal – база данных журналов коллекции Springer Journal изд-ва Springer Nature (1997-2015 гг.). – URL: <https://link.springer.com>
3. Президентская библиотека имени Б.Н. Ельцина. – URL: <https://www.prilib.ru>
4. Российская государственная библиотека: официальный сайт. – URL: <https://www.rsl.ru>
5. Российская национальная библиотека: официальный сайт. – URL: <http://nlr.ru>
6. Тамбовская областная универсальная научная библиотека им. А.С. Пушкина: официальный сайт. – URL: <http://www.tambovlib.ru>
7. Университетская библиотека онлайн: электронно-библиотечная система. – URL: <https://biblioclub.ru>
8. Федеральное хранилище «Единая коллекция цифровых образовательных ресурсов». – URL: <http://school-collection.edu.ru>
9. Федеральный портал «Российское образование». – URL: <https://www.edu.ru>
10. ЭБС «Университетская библиотека онлайн» . – URL: <http://www.biblioclub.ru>
11. Юрайт: образовательная платформа, электронно-библиотечная система. – URL: <https://urait.ru>

Электронная информационно-образовательная среда

https://auth.tsutmb.ru/authorize?response_type=code&client_id=moodle&state=xyz

Взаимодействие преподавателя и студента в процессе обучения осуществляется посредством мультимедийных, гипертекстовых, сетевых, телекоммуникационных технологий, используемых в электронной информационно-образовательной среде университета.