УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УНИВЕРСИТЕТ УПРАВЛЕНИЯ «ТИСБИ»

Кафедра информационных технологий

Утверждаю
Зав. кафедрой
О.В.Федорова

Протокол заседания
кафедры № 10
от 06.04.2026

Рабочая программа дисциплины

Наименование дисциплины	Методы программирования
По направлению подготовки	09.03.04 «Программная инженерия»
Профиль подготовки	Информационные технологии
Год набора	2026

Составитель:
канд.пед.наук, доц. Таренко Л.Б.

Казань

Содержание

1. Цели и задачи учебной дисциплины 3
2. Место дисциплины в структуре ОПОП 4
3. Требования к результатам освоения дисциплины 4
4. Структура и содержание дисциплины 5
4.1 Модульно-тематический план и пояснительная записка с указанием этапов формирования компетенций 5
4.2 Содержание дисциплины по темам (разделам) 10
4.3 Планы практических и семинарских занятий 14
4.4 Планы практической подготовки/лабораторных занятий 15
5. Учебно-методическое обеспечение самостоятельной работы студентов 17
6. Учебно-методическое и информационное обеспечение дисциплины 19
7. Материально-техническое обеспечение дисциплины 20
8. Оценка компетенций по изучаемой дисциплине 20
Приложение 1. Методические указания для обучающихся по освоению дисциплины
Приложение 2. Фонд оценочных средств для проведения текущей и промежуточной аттестации по дисциплине

1. Цели и задачи учебной дисциплины

Дисциплина является одной из основных по направлению подготовки **09.03.04 — Программная инженерия**.

Целью изучения дисциплины является освоение студентами основных вопросов классического структурного программирования, таких как типизация данных и основные стандартные типы данных, общая структура программ, основные инструкции обработки данных, организация циклических вычислений, обработка массивов и записей, использование файлов и подпрограмм, модульная структура программ, рекурсивные подпрограммы.

Задачи:

После изучения курса студент должен иметь представление:

- о современных алгоритмических языках, их области применения и особенностях.
- об основных методах и инструментах разработки программного обеспечения;

Знать: технологию разработки алгоритмов и программ; приемы структурного программирования; способы записи алгоритма на базовом языке высокого уровня; методы отладки и решения задач на ЭВМ в различных режимах.

Уметь: применять основные методы и инструменты для разработки алгоритмов и программ средней сложности и документировать их в соответствии с современными требованиями; тестировать, отлаживать и просчитывать на компьютере разработанные программы; интерпретировать полученные результаты.

Владеть: теоретическими и практическими навыками разработки алгоритмов и программ; навыками чтения и понимания исходного кода; навыками применения современных информационных технологий для решения профессиональных задач.

2. Место дисциплины в структуре ОПОП

Данная дисциплина относится к обязательной части Блока 1 учебного плана. До начала изучения дисциплины «Методы программирования» у студента должны быть сформированы компоненты компетенций, полученных в результате изучения дисциплины Вычислительная математика. Дисциплина находится во взаимосвязи с дисциплинами согласно схеме:

Обеспечивающие учебные дисциплины

↓ **Вычислительная математика**

Методы программирования

Обеспечиваемые учебные дисциплины

↓ **Основы операционных систем**
Введение в системы баз данных
Объектно-ориентированное программирование

3. Требования к результатам освоения дисциплины

Дисциплина участвует в формировании следующих компетенций в соответствии с ФГОС ВО по направлению «Программная инженерия»:

ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов

После освоения дисциплины студент должен получить следующие образовательные результаты соотнесенные с индикаторами достижения компетенций

индикаторы	**Результаты обучения по дисциплине**
Компетенция ОПК-6	
ОПК-6.1. Применяет языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для	**ОПК-6.1. З.1.** Знает основные языки программирования, современные программные среды **ОПК-6.1. У.1.** Умеет разрабатывать алгоритмы и

автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных	программы, пригодные для практического использования
ОПК-6.2. Программирует, отлаживает и тестирует программные продукты	**ОПК-6.2. В.1.** Имеет навыки программирования, откладки и тестирования программ

4. Структура и содержание дисциплины.

4.1. Модульно-тематический план и пояснительная записка с указанием этапов формирования компетенций

Общая трудоемкость дисциплины составляет 6 зачетных единиц (216 академических часа).

Модульная разбивка учебной дисциплины					
Направление: Программная инженерия Дисциплина «Методы программирования»					
Наименование модулей	Количество ауд. часов		Самостоятельная работа Очная	Всего часов	Индикаторы компетенции
	лекции очная	Практика очная			
<u>Модуль 1:</u>					
<u>Тема 1:</u> Основные этапы решения задач с помощью компьютера. Понятие об алгоритмах.	2	-	8	10	ОПК-6.1
<u>Тема 2:</u> Общая структура программы на языке высокого уровня. Основные типы данных.	4	4	8	16	
<u>Модуль 2</u>:					
<u>Тема 1</u>: Инструкции присваивания, ввода, вывода	2*	4	8	14	ОПК 6.1 ОПК 6.2
<u>Тема 2</u>: Основные инструкции проверки условий: простая инструкции проверки условий и множественная инструкция проверки условий	4*	4	10	18	

* Данная тема изучается с элементами интерактивных методов обучения

Модуль 3:					
Тема 1: Инструкции цикла: цикл с известным числом повторений	3*	3	8	14	ОПК 6.1 ОПК 6.2
Тема 2: Инструкции цикла с неизвестным числом повторений	3*	3	8	14	
Модуль 4:					
Тема 1: Массивы: одномерные и многомерные	4	4*	8	16	ОПК 6.1 ОПК 6.2
Тема 2: Строки и записи/структуры	6	6*	8	20	
Модуль 5:					
Тема 1: Файлы: назначение, разновидности, общие принципы использования	4	4	8	16	ОПК 6.1 ОПК 6.2
Тема 2: Понятие о подпрограммах: функции	4	4	8	16	
Тема 3: Рекурсивные подпрограммы.	4	4	8	16	
Подготовка к зачету			**10**	**10**	
Подготовка к экзамену			**36**	**36**	
ИТОГО:	**40**	**40**	**136**	**216**	

Пояснительная записка
с этапами формирования компетенций

Данный курс разбит на пять логически завершенных и взаимосвязанных между собой модуля, которые охватывают весь материал дисциплины, обеспечивают приобретение образовательных результатов в соответствии с федеральными государственными образовательными стандартами. Порядок освоения модулей выстраивает траекторию и этапы формирования заявленных компетенций (или их составляющих).

Каждый модуль состоит из 2-3 тем, содержащих определенный раздел учебного материала, и представляет собой законченный блок информации. По каждой теме в соответствии с учебным планом проводятся лекции и практические занятия. Предусмотрена индивидуальная самостоятельная работа, состоящая из подготовки к разделам, выделенным для самостоятельного изучения, подготовки к практическим занятиям по соответствующим темам с использованием лекционного материала, учебных пособий, рабочих программ дисциплин, Internet-ресурсов, а также рекомендованной дополнительной литературы.

После прохождения первого модуля, включающего в себя две темы, будут получены следующие образовательные результаты:

1. Студент должен владеть такими понятиями, как структурное программирование, алгоритмизация, типизация данных
2. Уметь использовать основные стандартные типы данных
3. Знать общую структуру программы

Уровень освоения проверяется компьютерным тестированием и выполнением практических задач с использованием программных средств в соответствии с темами изучаемого модуля.

Второй модуль предполагает приобретение навыков по следующим темам: основные инструкции, проверка условий.

Планируемые образовательные результаты:

1. Знать формат и правила написания операторов ввода-вывода и проверки условия.
2. Понимать назначение и иметь практические навыки применения инструкций ввода и вывода при решении практических задач

3. Понимать назначение использования инструкций проверки условий

Уровень освоения компетенций проверяется компьютерным тестированием и решением практических задач с использованием программных средств в соответствии с темами изучаемого модуля.

Третий модуль охватывает две темы, рассматривающие возможность организации циклических вычислений. Изучив третий модуль, студенты должны получить следующие образовательные результаты:

1. Знать назначение циклических структур.
2. Уметь использовать циклические операторы при написании программ и чтении исходного текста
3. Иметь практические навыки использования циклических структур

Уровень освоения компетенций проверяется компьютерным тестированием и решением практических задач с использованием программных средств в соответствии с темами изучаемого модуля.

Четвертый модуль охватывает две темы, посвященные базовым структурам данных. Изучив четвертый модуль, студенты должны получить следующие образовательные результаты:

1. Знать структуру «массив» и способы его описания в программах
2. Уметь обрабатывать массивы, как совокупность однотипной информации
3. Владеть навыком использования при разработке программ структуры запись, как структуры, содержащей разнотипную информацию

Уровень освоения компетенций проверяется компьютерным тестированием, и написанием и выполнением практических задач с использованием программных средств в соответствии с темами изучаемого модуля.

Пятый модуль дает знания и навыки по использованию файлов и подпрограмм, по использованию модулей и рекурсивных подпрограмм Изучив пятый модуль, студенты должны получить следующие образовательные результаты:

1. Знать структуру файлов и подпрограмм, иметь практические навыки создания и подключения модулей
2. Уметь использовать файлы для обработки и хранения информации
3. Понимать назначение и иметь навыки оформления подпрограмм, уметь

использовать рекурсивные подпрограммы

Уровень освоения компетенций проверяется компьютерным тестированием и решением практических задач с использованием программных средств в соответствии с темами изучаемого модуля.

Данное деление дисциплины на модули активизирует самостоятельную работу студентов, повышает интенсивность и системность учебной работы, регулирует контроль учебной деятельности студентов в течении семестров, усиливает мотивацию студентов к изучению учебного материала.

4.2. Содержание дисциплины по темам (разделам)

Тема 1. Основные этапы решения задач с использованием компьютера. Способы записи алгоритма.

Этапы создания программного обеспечения. Основные методы программирования: понятие структурного, процедурного и модульного программирования. Принципы процедурного программирования. Основные этапы разработки программ с использованием процедурного подхода.

Критерии качества программы. Жизненный цикл программы. Модификация программного обеспечения. Постановка задачи и спецификация программы. Анализ программы. Корректность программы.

Языки программирования низкого уровня. Хронология возникновения и развития языков программирования высокого уровня. Инструментальные средства для разработки программ в среде Windows.

Понятие об алгоритмах. Линейная, разветвленная и циклическая структуры. Требования, предъявляемые к алгоритму. Формы описания алгоритмов: словесная, графическая (блок-схемы), псевдокод.

Тема 2. Программа на языке высокого уровня. Основные типы данных.

Понятие синтаксиса и семантики языка программирования. Алфавит языка. Служебные (ключевые) слова. Имена (идентификаторы) переменных. Общая структура программы. Заголовок, раздел описания, тело программы.

Принцип типизации данных. Понятие скалярных и структурированных

типов данных. Основные типы, входящие в группу стандартных типов данных: целый; вещественный; логический; символьный; строковый. Выражения. Операции, допустимые для каждого типа данных, их приоритетность. Стандартные функции, применяемые для перечисленных типов данных.

Тема 3. Инструкции присваивания, ввода, вывода.

Инструкция присваивания. Основные правила использования инструкции присваивания. Инструкции ввода. Ввод данных различного типа с помощью оператора ввода. Оператор ввода без параметров. Инструкция вывода. Формат вывода для числовых переменных. Чередование в списке вывода текстовых констант и выводимых переменных.

Тема 4. Управляющие операторы языка.

Простая инструкция проверки условий

Представление основных управляющих структур программирования. Оператор условной передачи управления. Инструкция проверки условий: простая и множественная (инструкция выбора). Общий вид простой инструкции проверки условия. Принцип работы. Вложенные условные инструкции, глубина вложенности.

Множественная инструкция проверки условий

Общий вид инструкции выбора. Селектор: тип, допустимые значения. Принцип работы.

Тема 5. Управляющие операторы языка.

Цикл с известным числом повторений.

Инструкции цикла. Цикл с известным числом повторений. Принцип работы. Счетчик или параметр цикла. Тело цикла. Вложенные циклы.

Тема 6. Управляющие операторы языка.

Цикл с неизвестным числом повторений.

Циклы с неизвестным числом повторений. Две разновидности: с предусловием и с постусловием. Особенности использования. Конструирование условий выхода из цикла. Использование циклов с неизвестным числом повторений для организации многократного диалогового повторения программы.

Тема 7. Массивы: одномерные и многомерные

Утверждения о массивах. Массив как составная структура данных. Представление массивов в памяти. Порядковые номера (индексы) элементов. Вычисление адреса элемента массива. Одномерные массивы. Общий вид описания массива. Типичные задачи обработки массивов.

Многомерные массивы. Представление двумерных массивов в памяти. Вычисление адреса элемента массива. Общий вид описания массива. Типичные задачи обработки массивов. Использование циклов для работы с массивами. Преимущества и недостатки массивов.

Тема 8. Строки и записи.

Объявление переменных строкового типа. Основные операции со строками.

Составная структура данных (запись). Понятие поля записи. Объявление переменных типа "запись" в программе. Использование записей. Операции над записями. Доступ к полям записи. Массивы в качестве полей записи. Использование специального оператора присоединения при работе с записями. Вложенные записи. Особенности обработки и использования.

Тема 9. Файлы: назначение, разновидности, общие принципы использования

Общие понятия об использовании файлов. Описание файлового типа данных. Объявление файловых переменных. Инициализация файла. Открытие файла для чтения или записи. Обработка компонентов файла. Закрытие файла. Использование типизированных файлов. Работа с файлом записей.

Описание файлового типа данных. Объявление файловых переменных. Инициализация файла. Открытие текстового файла для чтения или записи. Обработка компонентов текстового файла. Закрытие файла. Текстовые файлы: особенности обработки. Использование текстовых файлов для хранения числовой информации.

Тема 10. Понятие о подпрограммах: функции.

Понятие о подпрограммах. Преимущества использования подпрограмм.

Вызывающая и вызываемая подпрограммы и их взаимодействие. Описание функции. Передача и обработка данных. Глобальные и локальные переменные. Правило видимости имен. Формальные и фактические параметры. Параметры-переменные и параметры-значения.

Тема 11. Рекурсивные подпрограммы.

Понятие о рекурсивных подпрограммах. Рекурсивные определения и алгоритмы. Прямая и косвенная рекурсия. Программирование рекурсивных алгоритмов. Преимущества и недостатки рекурсивной реализации в сравнении с циклической реализацией. Механизм вложенных вызовов и возвратов. Стековый принцип работы рекурсивного механизма. Сохранение и восстановление значений локальных переменных. Использование рекурсивных программ: метод двоичного поиска.

4.3. Планы практических и семинарских занятий

Тема: Основные инструкции

Основные вопросы темы

1. Знакомство с инструментом разработки программного обеспечения

Запуск системы программирования (Lazarus, Visual Studio). Управление системой с помощью меню и мыши. Работа с файлами исходных текстов (сохранение и открытие). Освоение встроенного текстового редактора.

2. Создание простейших программ

Ввод исходного текста программы (Pascal Lazarus / C++). Компиляция на язык машины. Чтение исходного кода, понимание, исправление синтаксических ошибок. Выполнение программы. Отладка и тестирование создаваемых программ.

3. Использование инструкций присваивания, ввода и вывода

Освоение инструкции присваивания. Создание программ (программного кода), использующих операторы ввода/вывода. Ввод исходного кода программы. Вывод полученных результатов, используя форматы вывода для числовых переменных. Вывод текстовых сообщений. Отладка и тестирование программ.

Тема: Управляющие операторы языка

Основные вопросы темы

1. Освоение простой инструкции проверки условий

Создание программы вычисления значения функции, заданной разными формулами на двух интервалах. Использование вложенной условной инструкции при вычислении корней квадратного уравнения. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

2. Освоение множественной инструкции проверки условий

Создание программ с использованием инструкции выбора. Использование в теле инструкции на месте значения селектора одиночных допустимых значений селектора, списки допустимых значений и интервалы допустимых значений. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

3. Использование циклов с известным числом повторений

Создание программ, позволяющих многократно выполнять одни и те же действия заданное число раз с разными данными. Построение треугольной таблицы из символов с использованием вложенных циклов. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

4. Освоение циклов с неизвестным числом повторений

Создание в программах внешнего диалогового цикла для возможности многократного выполнения программы по желанию пользователя, используя цикл с постусловием. Использование цикла с предусловием для работы с заранее неизвестным количеством вводимых пользователем чисел. Конструирование условия выхода из цикла для корректного завершения работы цикла. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

Тема: Массивы

Основные вопросы темы

1. Одномерные массивы

Запись случайно сгенерированных чисел в массив. Вывод их на экран в обратном порядке при помощи инструкции цикла с шагом (-1). Ввод с

клавиатуры некоторого набора чисел, запоминание их в массиве, суммирование элементов массива с выводом результата на экран. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

2. Многомерные массивы

Генерация двухмерного массива случайных чисел. Поиск в массиве максимального и минимального числа, подсчет среднего значения элементов по строкам и столбцам. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

Ввод числовой матрицы с клавиатуры и вывод на экран транспонированной матрицы.

Вычисление произведения двух матриц. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

Тема: Строки и записи

Основные вопросы темы

1. Текстовые строки

Ввод с клавиатуры произвольной текстовой строки, вывод ее на экран в обратном порядке, подсчет появлений заданных пользователем символов в строке. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

2. Использование структуры данных “запись”

Создание программы обработки данных о студенческой группе, хранящей в виде массива записей индивидуальные сведения для каждого студента. Организация возможности добавления нового студента, поиска заданного студента и изменения для него некоторых данных. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

Тема: Файлы

Основные вопросы темы

1. Использование простых типизированных файлов

Создание программ, осуществляющих следующие действия: создание

файла с целыми числами; чтение данных из файла; копирование файла; объединение файлов. Организовать возможность задания имен файлов пользователем и многократный выбор необходимого действия из предлагаемого меню. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

2. Использование типизированных файлов с компонентами-записями

Создание программы, позволяющей сохранять массив записей о студентах в файле, читать из файла и объединять два файла. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

3. Обработка текстовых файлов

Написать программу, обеспечивающую следующие возможности: создание текстового файла; чтение файла и вывод его содержимого на экран; копирование файлов; объединение файлов друг за другом и последовательными парами строк. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

Тема: Функции

Основные вопросы темы

1. Использование функций

Создание программ, которые с помощью внутренних подпрограмм-функций осуществляют поиск максимального числа и поиск первого появления заданного числа в массиве. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

Тема: Рекурсивные подпрограммы

Основные вопросы темы

Создание программы рекурсивного вычисления факториала. Реализация рекурсивной программы двоичного поиска в упорядоченном массиве целых чисел. Компиляция, чтение исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

4.4. Планы практической подготовки/лабораторных занятий

Не предусмотрено учебным планом.

5. Учебно-методическое обеспечение самостоятельной работы студентов

В процессе самостоятельного изучения студент обязан проработать перечисленные ниже темы, для углубления теоретических знаний и практических навыков, на основании следующих методических рекомендаций по самостоятельной работе.

Самостоятельная работа студентов регламентируется Положением об организации самостоятельной работы студентов.

Основными видами учебных занятий для студентов по данному курсу учебной дисциплины являются: лекции, практические занятия и самостоятельная работа студентов. Самостоятельная работа студентов является составной частью их учебной работы и имеет целью закрепление и углубление полученных знаний, умений и навыков, поиск и приобретение новых знаний.

В соответствии со спецификой предмета, самостоятельная работа включает в себя:

1. самостоятельное изучение материала по заданным преподавателем темам;
2. чтение/изучение исходного программного кода для понимания назначения операторов;
3. подготовку к практическим занятиям по темам;
4. работу с литературой и учебно-методическими пособиями.

Самостоятельная подготовка к практическим занятиям выполняется студентом в свободное от занятий время и включает в себя:

- проработку лекционного материала по указанной теме;
- подготовку алгоритмов решения задач по теме;
- написание программного кода в соответствии с разработанным алгоритмом;
- проверку работоспособности программ, как в домашних условиях, так и в компьютерных классах института;
- предоставление результатов самостоятельной работы

преподавателю.

Темы для самостоятельного освоения

1. Теорема структуры и структурное программирование.
2. Утверждения о программах.
3. Нелинейные структуры данных.
4. Инвариантные утверждения.
5. Способы конструирования и верификации программ.

Темы практических занятий

1. Использование инструкции вывода для чередования в списке вывода текстовых констант и выводимых переменных.

2. Изучение множественной инструкции проверки условий.

3. Изучение циклов (с известным и неизвестным числом повторений)

4. Комментарии в листинге программного кода

5. Освоение многомерных массивов

6. Изучение записей

7. Изучение файлов

8. Использование процедур и функций

9. Использование модулей в программах

10. Освоение рекурсивных подпрограмм

6. Учебно-методическое и информационное обеспечение дисциплины

Основная:

1. *Кувшинов, Д. Р.* Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2024. — 104 с. — (Высшее образование). — ISBN 978-5-534-07559-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/540199
2. *Огнева, М. В.* Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2024. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/539713

3. Литвиненко, В. А. Основы программирования в C++Builder CE : учебное пособие / В. А. Литвиненко. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2024. — 120 с. — ISBN 978-5-9275-4800-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/149676.html. — Режим доступа: для авторизир. Пользователей
4. Биллиг, В. А. Основы объектного программирования на C# (C# 3.0, Visual Studio 2008) : учебник / В. А. Биллиг. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. — 409 с. — ISBN 978-5-4497-0880-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: https://www.iprbookshop.ru/146364.html. — Режим доступа: для авторизир. Пользователей

Дополнительная литература

5. *Черпаков, И. В.* Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — 2-е изд. — Москва : Издательство Юрайт, 2024. — 196 с. — (Высшее образование). — ISBN 978-5-534-18759-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/545506
6. *Федоров, Д. Ю.* Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 227 с. — (Высшее образование). — ISBN 978-5-534-17323-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/539651
7. *Гниденко, И. Г.* Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 248 с. — (Высшее образование). — ISBN 978-5-534-18130-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/536903

Интернет-ресурсы и перечень ежегодно обновляемых современных профессиональных баз данных и информационных справочных систем:

1. www.iprbookshop.ru Цифровой образовательный ресурс IPR SMART
2. https://urait.ru/ Образовательная платформа Юрайт
3. http://citforum.ru/
4. https://habr.com/ru/
5. https://stackoverflow.com/
6. https://social.technet.microsoft.com/Forums/ru-RU/home

7. Материально-техническое обеспечение дисциплины

Учебная аудитория, кабинет для самостоятельной работы студентов, читальный зал и библиотека, видеопроекционное оборудование, компьютер, оснащенный типовым пакетом системного и офисного ПО, в соответствии с Реестром материально-технического обеспечения аудиторного фонда Университета управления "ТИСБИ".

Комплект презентационного оборудования (компьютер с выходом в интернет, проектор, экран).

Персональный компьютер с выходом в интернет в компьютерных классах Университета для каждого студента на практических занятиях.

Пакет лицензионного системного и офисного ПО:

-Microsoft Windows 10 Pro -.

- Microsoft Office 2013 -.

Программное обеспечение, входящее в пакет, получает обновление в автоматическом, установленном разработчиком (компанией Microsoft) порядке, посредством сети Интернет.

- Visual Studio. Бесплатная интегрированная среда разработки программного обеспечения

- NetBeans IDE. Бесплатная интегрированная среда разработки с открытым исходным кодом для разработчиков программного обеспечения

- MS Visual C# Express. Бесплатный программный пакет для создания Windows приложений на базе ПК

8. Оценка компетенций по изучаемой дисциплине

Для оценки компетентности рекомендуется использовать рейтинговую оценку знаний, умений и навыков студента по окончанию изучения каждого Модуля в соответствии с Положением о модульно-рейтинговой системе организации образовательного процесс. Итоговая оценка (в баллах) складывается из баллов, набранных по каждому Модулю (семестровая оценка) и баллов, набранных, непосредственно на экзамене (зачете).

Расчет набранных баллов по дисциплине осуществляется в следующей последовательности:

$C = \frac{M_1 + M_2 + ...M_n}{n} \times 0{,}6$, где М – количество баллов по модулю; n – количество модулей

$З = К \times 0{,}4$, где К - количество баллов на экзамене (зачете);

$И = С + З + П$, где П – поощрительные баллы (от 1 до 5).

Уровень сформированности компетенций и их основные признаки оцениваются по следующим таблицам:

Оценка уровня сформированности компетенции ОПК-6

«Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов»

в части дисциплины «Методы программирования»

№ п/п	Уровни сформированности компетенции	Основные признаки уровня	Инструменты оценки сформированности уровня
1	2	3	4
1	**Пороговый уровень (как минимально допустимый)** (обязательный для всех студентов-выпускников вуза по завершении освоения ОПОП ВО) (от 60 до 70 баллов)	- Знает определения и способы описания основных структур для хранения и обработки информации - знает основные языки программирования, современные программные среды - Умеет составлять алгоритмы решения задач.	тестовый опрос, комп. тестирование, практические задания Экзамен (зачет)
2	**Базовый уровень (относительно порогового уровня)** **(**От 71 до 85 баллов**)**	- Знает определения и способы описания основных структур для хранения и обработки информации - знает основные языки программирования, современные программные среды - Умеет составлять алгоритмы решения задач. - Умеет использовать программные средства для решения практических задач - Умеет описывать основные структуры на языке программирования. - умеет разрабатывать алгоритмы и программы, пригодные для практического применения	Решение практических задач, творческие задания, комп. тестирование Экзамен (зачет)

		- Владеет навыками написания программ по образцу с использованием основных операторов (инструкций) языка программирования и программных средств	
3	**Повышенный уровень (относительно порогового уровня)** (От 86 до 100 баллов)	- Знает определения и способы описания основных структур для хранения и обработки информации - знает основные языки программирования, современные программные среды - Умеет составлять алгоритмы решения задач. - Умеет использовать программные средства для решения практических задач - Умеет описывать основные структуры на языке программирования. - умеет разрабатывать алгоритмы и программы, пригодные для практического применения - Владеет навыками написания программ по образцу с использованием основных операторов (инструкций) языка программирования - Владеет навыками использования программных средств для решения практических задач; - имеет навыки программирования, отладки и тестирования программ	тестовый опрос, комп. Тестирование, решение практических задач, творческие задания Экзамен (зачет)

Приложение 1.

Методические рекомендации для обучающихся по освоению дисциплины

Студентам на первом занятии необходимо ознакомиться с Рабочей программой дисциплины, где прописаны цели, задачи и трудоемкость дисциплины. Перед началом изучения дисциплины необходимо повторить учебный материал обеспечивающих учебных дисциплин предшествующих курсов, которые дают основу для изучения дисциплины Методы программирования, а именно: Математика (Алгебра и геометрия) и Информатика.

Затем необходимо ознакомиться с порядком изучения дисциплины, т.е. модульно-тематическим планом и пояснительной запиской с указанием этапов формирования заявленных компетенций.

И, наконец, ознакомиться с порядком оценивания результатов обучения, для чего необходимо изучить следующие документы: Положение о модульно-рейтинговой системе оценивания и Принципы оценки уровня знаний, умений и навыков (характеристика ответа).

Студент должен внимательно изучить перечень основной (дополнительной) литературы и взять необходимые учебники в библиотеке.

Организация учебного процесса изучения данной дисциплины имеет следующие основные особенности.

1. Чтение лекций по всему курсу сопровождается использованием компьютерной презентации (лекция – визуализация), включающей в себя слайды по всем разделам курса. Это экономит время, обычно затрачиваемое преподавателем на рисование сложных схем и алгоритмов, дает возможность оперативно обращаться к ранее пройденному материалу, повышает наглядность изучаемого материала, позволяет проводить занятия в интерактивной форме.

2. Для проведения занятий в интерактивной форме используются оригинальные проблемно-ориентированные задания по основным разделам курса.

3. Существенной особенностью дисциплины является ее практическая направленность, связанная с написанием большого количества небольших программ на практических занятиях. Поэтому важную роль в курсе имеет комплекс лабораторных работ, главной задачей которого является обучение студентов в процессе их самостоятельной работы на компьютерах, получение навыков применения современных информационных технологий для решения профессиональных задач и формирования компетенций.

При изучении данного курса используются интерактивные методы обучения, что помогает эффективнее сформировать заявленные компетенции.

На занятиях студентам предлагается разделиться на группы и выполнить творческие, проблемно-ориентированные задания по темам модуля, в которых необходимо:

1. Составить график, который позволит оптимизировать процесс выполнения работы в условиях ограниченного времени.
2. Осуществить поиск и систематизацию необходимой информации. Проанализировать отобранные материалы и выбрать оптимальный способ решения проблемы.
3. Приступить к проектированию структуры доклада, руководствуясь идеями и алгоритмами, почерпнутыми из материалов лекций и собственными идеями, и наработками.
4. Оформить результаты: обосновать выбор решения проблемы и подготовить презентацию и демонстрационный материал для защиты проекта

В результате выполнения совместной проектной работы осуществляется личностное включение студентов в процесс обучения, создаются условия для проявления творческих сил студентов.