УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УНИВЕРСИТЕТ УПРАВЛЕНИЯ «ТИСБИ»

Кафедра информационных технологий

Утверждаю
Зав. кафедрой
О.В. Федорова

Протокол заседания
кафедры №12
от 30.04.2025

Рабочая программа дисциплины

Наименование дисциплины	Программирование
По направлению подготовки	27.03.02. «Управление качеством»
Профиль подготовки	Управление качеством в бизнес-системах
Год набора	2025

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
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 27.03.02 «Управление качеством».

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 программных продуктов;

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 для обработки данных; 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 прикладных задач.

2. Место дисциплины в структуре ОПОП

Данная дисциплина относится к обязательной части Блока 1 учебного плана. До начала изучения дисциплины «Программирование» у студента должны быть сформированы компоненты компетенций, полученных в результате изучения дисциплин Цифровые технологии и основы искусственного интеллекта и Практикум по цифровым технологиям. Дисциплина находится во взаимосвязи с дисциплинами согласно схеме:

Обеспечивающие учебные дисциплины

↓ **Цифровые технологии и основы искусственного интеллекта**
Практикум по цифровым технологиям

Программирование

Обеспечиваемые учебные дисциплины

↓ **Сертификация продукции и услуг**
Инновации в управлении качеством

3. Требования к результатам освоения дисциплины

Дисциплина участвует в формировании следующих компетенций в соответствии с ФГОС ВО по направлению 27.03.02. «Управление качеством»:

ОПК-6 - Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

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 современные программные среды разработки компьютерных программ для решения прикладных задач	**ОПК-6.1 З.1** Знает существующие алгоритмы работы с данными и основные языки программирования, аспекты их практического применения

	ОПК-6.1 У.1 Умеет разрабатывать алгоритмы и компьютерные программы для обработки данных, применять языки программирования для решения прикладных задач **ОПК-6.1 В.1** владеет практическими навыками программирования и откладки компьютерных программ, а также принципами разработки программных продуктов

4. Структура и содержание дисциплины.

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

Общая трудоемкость дисциплины составляет 7 зачетных единиц (252 академических часа).

Модульная разбивка учебной дисциплины					
Направление: Управление качеством Дисциплина «Программирование»					
Наименование модулей	Количество ауд. часов		Самостоятельная работа Очная/заочная	Всего часов Очная/заочная	Индикаторы компетенции
	лекции очная/заочная	Практика очная/заочная			
Модуль 1:					
Тема 1: Основные этапы решения задач с помощью компьютера. Понятие об алгоритмах.	4/1	-/-	18/17	22/17	ОПК-6.1
Тема 2: Общая структура программы на языке высокого уровня. Основные типы данных.	4/1	4/1	14/17	22/19	
Модуль 2:					
Тема 1: Инструкции присваивания, ввода, вывода	2*/1	4/1	10/17	16/19	ОПК 6.1
Тема 2: Основные инструкции	4*/1	4/1	10/20	18/22	

* Данная тема изучается с элементами интерактивных методов обучения

проверки условий: простая инструкции проверки условий и множественная инструкция проверки условий					
Модуль 3:					
Тема 1: Инструкции цикла: цикл с известным числом повторений (цикл с счетчиком)	4*/2	4/2	12/19	20/23	ОПК 6.1
Тема 2: Инструкции цикла с неизвестным числом повторений	4*/2	4/2	17/18	25/22	
Модуль 4:					
Тема 1: Массивы: одномерные и многомерные	4/2	4*/1	13/18	21/21	ОПК 6.1
Тема 2: Строки и записи/структуры.	4/1	4*/2	11/21	19/24	
Модуль 5:					
Тема 1: Файлы: назначение, разновидности, общие принципы использования	4/1	4/2	15/18	21/20	ОПК 6.1
Тема 2: Понятие о подпрограммах: функции	4/1	4/1	11/17	19/19	
Тема 3: Рекурсивные подпрограммы.	2/1	4/1	19/19	25/21	ОПК 6.1
Подготовка к зачету			**6/6**	**6/6**	
Подготовка к экзамену			**18/18**	**18/18**	
ИТОГО:	**40/14**	**40/14**	**172/224**	**252/252**	

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 так же 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 при

разработке программных продуктов

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 при обработке данных.
2. Уметь использовать циклические операторы при написании программ
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 ти-

пов 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. 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. Компиля-

ция, 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. *Нагаева, И. А.* Программирование: Delphi : учебное пособие для вузов / И. А. Нагаева, И. А. Кузнецов ; под редакцией И. А. Нагаевой. — Москва :

Издательство Юрайт, 2024. — 302 с. — (Высшее образование). — ISBN 978-5-534-07098-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/540416

Дополнительная литература

4. Таренко Л.Б. Методика обучения программированию. Часть 1. Основные инструкции. Учебно-методическое пособие. Казань: ИЦ Университета управления «ТИСБИ», 2021. – 90с.
5. Таренко Л.Б. Методика обучения программированию. Часть 2. Базовые структуры. Учебно-методическое пособие. Казань: УВО «Университет управления «ТИСБИ», 2022. – 100с.
6. *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
7. *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
8. *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.

- Lazarus. Бесплатная среда по разработке программного обеспечения (https://lazarus-rus.ru/)

- Visual Studio. Бесплатная интегрированная среда разработки программного обеспечения

- Delphi 10.1 Berlin Professional Academic Concurrent. Среда разработки программного обеспечения. Акт предоставления прав № Tr044490 от 14.09.2016. Лицензия № 457049

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 компьютерные программы, пригодные для практического применения»

в части дисциплины «Программирование»

№ 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 зачет экзамен
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 - Владеет навыками написания программ по образцу с использованием основных операторов	решение практических задач, творческие задания, комп. тестирование зачет экзамен

		(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 зачет экзамен

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 Программирование, а именно: Цифровые технологии и основы искусственного интеллекта, Практикум по цифровым технологиям.

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.

Приложение 2

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«УНИВЕРСИТЕТ УПРАВЛЕНИЯ «ТИСБИ»
Кафедра информационных технологий

Фонд оценочных средств
для проведения текущей и промежуточной аттестации
по дисциплине
«Программирование»

направление подготовки: Управление качеством

профиль подготовки: Управление качеством в бизнес-системах

Казань

Содержание

1. Паспорт фонда оценочных средств
2. Наполнение фонда оценочных средств по формам контроля
   2.1 Фонд оценочных средств и шкала оценивания для текущего контроля.
      2.1.1 Компьютерное тестирование
      2.1.2 Тестовый опрос
      2.1.3 Выполнение практического задания
      2.1.4 Творческая задача

   2.2 Фонд оценочных средств и шкала оценивания для промежуточного контроля.
      2.2.1 Фонд оценочных средств для проверки знаний и умений (вопросы к зачету и экзамену)
      2.2.2 Фонд оценочных средств для проверки сформированности навыков (задачи к зачету и экзамену)

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

Формы контроля / формируемые компетенции и их индикаторы	ОПК-6
	ОПК-6.1
Формы текущего контроля	
Компьютерное тестирование	З1
Тестовый опрос	З1
Выполнение практических заданий	З1 У1 В1
Творческие задания	З1 У1 В1
Формы промежуточного контроля	
зачет	З1 У1В1
экзамен	З1 У1В1

2. Наполнение фонда оценочных средств по формам контроля.

2.1.Фонд оценочных средств и шкала оценивания для текущего контроля

2.1.1.Компьютерное тестирование

Для текущего контроля успеваемости студентов введено компьютерное тестирование по пройденным темам. Преподаватель формирует сценарий контрольного опроса, который содержит наименование тем и количество задаваемых вопросов по каждой теме. Рекомендуется охватывать 2-3 темы по 10-15 вопросов.

Для охвата всего курса «Программирование» предлагается сформировать сценарий из восьми тем по 10 вопросов. Тестирование помогает определить, владеет ли студент основными понятиями, а также выяснить, как полученные знания помогают ему при написании конкретных программ.

Образцы заданий для компьютерного тестирования

1. Что такое трансляция (компиляция) программы?

- перевод текста программы с языка программирования в набор машинных команд
- перевод текста программы с английского языка на русский
- поиск ошибок в программе

- перевод программы с машинного языка на язык высокого уровня

2. Какие ошибки характерны для процесса разработки новой программы?

- ошибки в написании исходного текста программы
- логические ошибки
- ошибки, связанные с объединением отдельных программных модулей
- ошибки при вводе исходных данных пользователем

3. Что включает в себя описание переменных в разделе описаний?

- определение имен-идентификаторов переменных
- указание типа каждой переменной
- указание доступности каждой переменной
- указание способа использования переменной

4. Основные особенности текстовых файлов: текстовый файл - это последовательность текстовых строк; каждая строка может иметь свою байтовую длину. Что еще?

a)каждая строка заканчивается специальным признаком конца строки

b)текстовые файлы можно обрабатывать не только последовательно, но и в любом порядке

5. При каких значениях переменных i и j цикл While (i>0) AND (j<0) do ... заканчивает свою работу?

- если i<=0
- если j>=0
- если одновременно i>0 и j<0
- условие цикла записано неправильно

6. Что такое алгоритм?

a)Четко определенная последовательность шагов, приводящая к получению необходимого результата

b)любая последовательность шагов

c)формализованное описание постановки задачи

d)программа на языке программирования

7. Какие управляющие структуры являются основными при описании алгоритмов?

a)линейная последовательность шагов

b)проверка условия

c)циклическое повторение действий

d)структура ввода/вывода данных

8. Какие ошибки называются синтаксическими?

a)ошибки в неправильном оформлении текста программы

b)ошибки в алгоритме программы

c)ошибки при вводе исходных данных

d)любые возможные типы ошибок

9. Какие ошибки характерны для процесса разработки новой программы?

a)ошибки в написании исходного текста программы

b)логические ошибки

c)ошибки, связанные с объединением отдельных программных модулей

10. Дан двухмерный массив M символов 10 на 20. На каком расстоянии от начала массива будет находиться элемент M[5, 5]?

a)на расстоянии 84 байтов

b)на расстоянии 105 байтов

c)на расстоянии 55 байтов

d)на расстоянии 80 байтов

11. Какое значение имеет логическое выражение (((x>10) AND (y=0)) OR (z<5)) при x=11, y=0 и z=8?

a)истина

b)ложь

c)неопределенное

d)выражение синтаксически неверно

12. Какая операция не относится к операциям с логическими переменными?

a)логическое сложение

b)логическое умножение

c)отрицание

d)логическое вычитание

13. Какие из следующих логических условий являются истинными?

a)(true OR false) = true

b)(true AND true) = true

c)(false OR true) = false

d)(true AND true) = false

14. Какая подпрограмма называется рекурсивной?

a)если она вызывает сама себя

b)если она имеет внутренние подпрограммы

c)если она оформлена в виде функции

d)если она имеет формальные параметры

15. Что происходит при выполнении рекурсивных подпрограмм?

a)создается вложенная последовательность вызовов и возвратов в обратном порядке

b)создается множество одновременно выполняемых программ

c)создается произвольная последовательность вызовов в случайном порядке

d)создается линейная последовательность вызовов без возвратов

Критерии оценивания

Результат компьютерного тестирования выдается в процентах. Порог прохождения компьютерного тестирования – 60%. 1% приравнивается к 1 баллу. Правильно ответив на все вопросы теста, студент может набрать 100 баллов.

2.1.2. Тестовый опрос

Тестовый опрос используется для текущего контроля умений и навыков студентов. В целях освоения компетенции для выполнения тестового опроса студенты изучают необходимый теоретический материал по теме.

Тестовый опрос выполняется на лекционных занятиях в расчете по времени 10-15 минут от пары и содержит 20 небольших вопросов.

Примерные вопросы для тестового опроса

Вопросы включают оценку закрепления материала, пройденного на лекциях, а также вопросы, направленные на выявление уровня понимания студентом сути использования различных операторов языка программирования при написании программ.

1. Какие ошибки называют синтаксическими?
2. Что такое алгоритм?
3. Что такое вложенные условные инструкции?
4. Для чего используется инструкция множественного выбора?
5. Как записывается инструкция выбора?
6. Чем отличается использование условной инструкции и инструкции выбора?
7. Какие служебные слова используются в программе для описания целочисленных переменных?
8. Когда цикл с известным числом повторений завершает свою работу?

9. Каким служебным словом в программе обозначается структура массив?
10. При каком условии завершает работу цикл While?
11. Какие типы данных можно использовать в качестве элементов массива?
12. Чем определяется размер памяти, выделяемой для хранения элементов массива?
13. Какие служебные слова используются при описании структуры запись?
14. Сколько резервируется памяти в компьютере для структуры «запись» в целом?
15. В чем состоят отличия функций от процедур?
16. Что может включать в себя заголовок процедуры?
17. Что должен включать в себя заголовок функции?
18. Переменные каких типов можно использовать в качестве счетчика повторений в циклах for_to_do?
19. Когда цикл с известным числом повторений заканчивает свою работу?

Критерии оценивания

Результат выполнения тестового опроса студентом оценивается в баллах. Оценка письменного ответа студента по результатам опроса производится по 100 балльной системе. Каждый правильный ответ оценивается в 5 баллов. Ответив верно на все задания тестового опроса. студент может набрать 100 баллов. Менее 60 баллов – опрос не пройден.

2.1.3. Выполнение практических заданий по написанию программ

По каждой теме модуля в курсе разработано по 3-5 практических задания, которые заключаются в написании студентами программ для их выполнения. Текущий контроль выполняется на каждом практическом занятии и за-

ключается в проверке преподавателем правильности работы написанной студентом программы. Для этого программа выполняется на нескольких тестовых примерах с анализом полученных результатов. Обнаруженные ошибки исправляются студентом.

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.

4. 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.

5. 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.

6. 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, чте-

ние исходного кода, исправление синтаксических ошибок. Отладка и тестирование программ.

7. 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.

8. 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.

9. 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.

10. 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. Текстовые файлы с числовой информацией

Написать программу для создания и чтения текстового файла с числами, хранящимися в файле разбитыми на строки. Компиляция, чтение исходного

кода, исправление синтаксических ошибок. Отладка и тестирование программ.

16. 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.

17. Рекурсивные подпрограммы

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.

Критерии оценивания выполнения практических заданий

Результат выполнения практического задания студентом оценивается в баллах. За каждое правильно выполненное задание ставятся баллы

Результат	Балл
Решение задачи логично, разработан наиболее эффективный алгоритм, программа демонстрирует авторскую позицию студента. При написании программы студент продемонстрировал применение знаний к реальным практическим ситуациям, объясняет решение программы и дает свою оценку решения проблемы. Студент не затрудняется с ответом при видоизменении задания и исходных данных, и правильно обосновывает принятое решение, владеет разносторонними навыками и приемами написания исходного текста программ, тестирования и отладки, чтения исходного текста.	85-100
Разработан не самый эффективный алгоритм решения задачи. При написании программы студент продемонстрировал применение знаний к реальным практическим ситуациям, объясняет решение программы. Студент не затрудняется с ответом при видоизменении исходных данных, и правильно обосновывает принятое решение, владеет навыками и приемами написания программ и написания и чтения исходного текста.	71-84
Алгоритм решения задачи разработан по образцу. Студент затрудняется с ответом при видоизменении исходных данных, владеет элементарными навыками и приемами написания исходного текста программ, тестирования и отладки.	60-70

Не разработан алгоритм решения задачи, программа не написана. Студент демонстрирует непонимание задания.	Менее 60

2.1.4.Творческие задания

Для решения проблемно-ориентированной, творческой задачи студенты делятся на группы и распределяют между собой роли: ответственный за презентационный материал, ответственный за текст доклада, ответственный за поиск информации, за разработку программ. Информацию, необходимую для выполнения задания, участники группы получают из книг, лекционного материала и из Интернета. Промежуточные обсуждения полученных результатов проводятся по мере выполнения задания.

Завершается работа защитой проекта на лекционном занятии, в рамках которой студенты представляют:

- текст выступления по заданной теме;
- отлаженные, протестированные, работающие программы (программы, отражающие работу основных операторов или применение основных структур языка программирования)
- листинг программного кода с комментариями
- презентацию.

Темы для задания:
Основные типы данных.
Инструкции присваивания, ввода, вывода
Простая инструкции проверки условий
Множественная инструкция проверки условий
Инструкция цикла с известным числом повторений
Инструкции цикла с неизвестным числом повторений
Массивы: одномерные
Массивы: многомерные
Строки, записи и множества.
Типизированные файлы
Текстовые файлы
Процедуры и функции
Модульное программирование
Рекурсивные подпрограммы

Критерии оценивая творческих заданий

Задания оцениваются в баллах. Степень раскрытия сущности темы и правильно примененные теоретические знания при написании, тестировании и отладке программы – наиболее важный критерий оценки работы студента над заданием.

Критерии оценивания	**Баллы**
С использованием компьютера осуществлены поиск и систематизация необходимой информации. Проанализированы отобранные материалы и выбран оптимальный способ решения проблемы. Спроектирована структура доклада. Разработан собственный эффективный алгоритм программы, демонстрирующей работу заданного в теме оператора языка программирования. Представлена отлаженная, правильно оформленная с использованием программных средств и протестированная при различных исходных данных программа. Оформлены результаты: обоснован выбор решения проблемы, сформулированы выводы, подготовлена презентация и демонстрационный материал для защиты программного продукта, соблюдены требования к внешнему оформлению.	86-100
Основные требования к заданию выполнены, но при этом допущены недочёты. В частности, разработан не самый эффективный алгоритм решения; для демонстрации работы основных операторов языка программирования взята шаблонная задача, имеются упущения в оформлении.	71-85
В работе имеются отступления от требований к выполнению задания, тема освещена лишь частично; работа операторов представлена на одном примере, взятом из материалов лекций, не разработан собственный алгоритм решения проблемы; отсутствуют выводы, слабое оформление презентации.	66-70
Задача представлена, но тема не раскрыта, обнаруживается существенное непонимание использования программных средств для решения практических задач.	60-65
Задача не представлена	<60

2.2. Фонд оценочных средств и шкала оценивания для промежуточного контроля

2.2.1. Фонд оценочных средств для проверки знаний/умений

Вопросы на зачете

1. Понятие о типах данных. Целые типы данных: обозначение, диапазон значений, длина внутреннего представления
2. Понятие о типах данных. Операции, допустимые при работе с целыми и вещественными типами данных (рассмотреть на примерах)
3. Понятие о типах данных. Вещественные типы данных (обозначения, длина внутреннего представления, допустимые операции)
4. Основные логические операции. Правила выполнения логических операций
5. Инструкция присваивания. Основные требования.
6. Инструкции вывода для целочисленных и вещественных переменных
7. Инструкции ввода и вывода (общий вид и примеры)
8. Простая инструкция проверки условий (общий вид, примеры)
9. Множественная инструкция проверки условия (общий вид и примеры)
10. Цикл с известным числом повторений (общий вид, примеры).
11. Вложенные циклы с известным числом повторений (общий вид, примеры).
12. Цикл с предусловием (общий вид, примеры)
13. Цикл с постусловием (общий вид, примеры)
14. Одномерные массивы: назначение, представление в памяти, особенности обработки, описание
15. Двухмерные массивы: назначение, представление в памяти, особенности обработки, описание
16. Типичные задачи обработки массивов (перечислить). Описание двухмерного массива (несколькими способами)
17. Цикл с известным числом повторений и его использование для обработки массивов
18. Циклы с неизвестным числом повторений (общий вид). Сравнить.

Вопросы на экзамене

1. Общая структура программы. Синтаксис и семантика языка программирования.
2. Понятие о типах данных. Существующие типы данных.

3. Строковые типы данных. Особенности обработки
4. Оператор множественного выбора Case – Of: общий вид, принцип работы.
5. Цикл с известным числом повторений: общий вид, принцип работы. Использование циклов в программах для обработки массивов.
6. Циклы с неизвестным числом повторений: общий вид, принцип работы, примеры использования циклов в программах.
7. Одномерные массивы: назначение, описание, особенности применения в программах.
8. Двумерные массивы: назначение, представление в памяти, особенности обработки.
9. Текстовые файлы: объявление, особенности обработки в программе.
10. Типизированные файлы: объявление, особенности обработки
11. Функции как разновидности подпрограмм: определение, общий вид, принцип работы, преимущества.
12. Понятие о подпрограммах: назначение, использование
13. Подпрограммы: локальные и глобальные параметры, правило видимости имен
14. Структуры данных типа “запись”: назначение, описание, особенности обработки
15. Подпрограммы: входные и выходные параметры, два способа передачи параметров
16. Файлы: назначение, описание, разновидности.
17. Простая инструкция проверки условий: описание, общий принцип работы.
18. Логические типы данных. Правила выполнения логических операций.
19. Целые типы данных. Допустимые операции
20. Символьные типы данных. Особенности использования в программах.
21. Массивы: представление в памяти, особенности обработки.
22. Основные инструкции: присваивание, ввод и вывод. Примеры использования.

23.Вещественные типы данных. Допустимые операции

2.2.2 Фонд оценочных средств для проверки умений/навыки

Задачи для проверки на зачете сформированности умений и навыков студентов

1. Как определяется общий объем резервируемой памяти для массивов. Рассчитать для одномерного и двухмерного массивов (на различных примерах)
2. Формула вычисления адреса элемента одномерного и двухмерного массивов (общий вид и расчет адреса элемента на собственном примере)
3. Сгенерировать одномерный массив из 100 целых чисел в диапазоне от 0 до 50 и вывести его на экран в обратном порядке
4. Сгенерировать массив целых чисел в диапазоне от 0 до 9 размерностью 10 на 15 и вывести его на экран в виде таблицы
5. Написать текст программы, реализующей поиск первого появления заданного числа в одномерном массиве
6. Написать текст программы, реализующей поиск всех появлений заданного числа в одномерном массиве
7. Сгенерировать одномерный массив из 50 целых чисел в диапазоне от 0 до 100 и вывести его на экран в обратном порядке
8. Сгенерировать одномерный массив из 50 целых чисел в диапазоне от -100 до 100 и вывести его на экран
9. Написать фрагмент программы: ввести с клавиатуры массив из 15 целых чисел и вывести их на экран

Задачи для проверки на экзамене сформированности умений и навыков студентов

1. Написать программу вычисления значения функции:

$$y = \begin{cases} x\cos(2x), x < 5 \\ \dfrac{8x+12}{x}, x \geq 5 \end{cases}$$

2. Используя инструкцию выбора Case – of написать программу перевода номера дня недели в его наименование
3. Написать фрагмент программы, позволяющей сгенерировать и вывести на экран одномерный массив из 50 целых чисел в диапазоне от 0 до 150.
4. Описать двумерный массив 30 на 45 целых чисел. Найти общий объем, занимаемый в памяти элементами массива.
5. Описать двумерный массив 4 на 10 вещественных чисел. Найти общий объем, занимаемый в памяти элементами массива.
6. Написать фрагмент программы, позволяющей подсчитать сумму всех элементов в двумерном массиве размерности 13 на 25 целых чисел, сгенерированных в диапазоне от 0 до 50.
7. Описать программно запись о сотрудниках организации, содержащую следующую информацию: номер сотрудника, ФИО, адрес, должность, оклад, дата приема
8. Найти, чему будет равно значение логических операций при следующем наборе исходных данных: x=22, y=0, z=-90, v=true.

(x<100) and (z>13) or (y<40)

((not v) or (x+12)<35) or (z>0)

9. Написать программу создания типизированного файла с целыми числами. Количество чисел 150. Числа генерируются случайным образом в интервале от -60 до +60.
10. Написать программу чтения (с выводом на экран) из существующего типизированного файла с целыми числами.

Критерии оценки уровня усвоения знаний, умений и навыков по результатам экзамена

Характеристика ответа	Европейская оценка	Рубежные баллы	Оценка	Уровень сформированности компетенций
Дан полный, развернутый ответ на поставленный теоретический вопрос экзаменационного билета, показана совокупность осознанных знаний об объекте, проявляющаяся в свободном ориентировании понятиями и определениями, умении использовать при написании программ. Ответ формулируется в терминах науки, изложен профессиональным языком, логичен, демонстрирует авторскую позицию студента. При ответе на экзаменационный билет студент демонстрирует применение знаний к реальным профессиональным ситуациям, объясняет решение задачи (программы) и дает свою оценку решения проблемы. Причем студент не затрудняется с ответом при видоизменении задания и правильно обосновывает принятое решение, владеет разносторонними навыками и приемами написания и чтения исходного текста программ, методами разработки.	**A**	**100-96**	**5+**	**Повышенный уровень сформированности компетенций**
Дан полный, развернутый ответ на поставленный теоретический вопрос, показана совокупность осознанных знаний об объекте, раскрыты основные положения темы; в ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий. Знание об объекте демонстрируется на фоне понимания его в системе данной науки и междисциплинарных связей. Умеет тесно увязывать теорию с практикой. Программная реализация задачи выполнена правильно и с обоснованием принятого решения. Ответ изложен в терминах науки. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа.	**A**	**95-91**	**5**	
Дан полный, развернутый ответ на поставленный вопрос; в ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий. Студент владеет разносторонними навыками и приемами написания программ для решения практических задач. Ответ изложен в терминах науки. В ответе допущены недочеты, исправленные студентом с помощью преподавателя.	**A**	**90-86**	**5-**	
Дан полный, развернутый ответ на поставленный вопрос. Ответ четко структурирован, логичен, изложен в терминах науки. Студент владеет разносторонними навыками и приемами написания программ для решения практических задач. Могут быть допущены недочеты или незначительные	**B**	**85-81**	**4+**	**Базовый уровень сформированности компетен-**

ошибки, исправленные студентом с помощью преподавателя.				
Дан полный, развернутый ответ на поставленный вопрос. Ответ четко структурирован, логичен, изложен в терминах науки. Программная реализация задачи выполнена верно. Ответы на дополнительные вопросы логичны, изложены в терминах науки, однако допущены незначительные ошибки или недочеты, исправленные студентом с помощью "наводящих" вопросов преподавателя.	**C**	**80-76**	**4**	
Студент демонстрирует достаточные теоретические и практические знания. Дан полный, но недостаточно последовательный ответ на поставленный вопрос, но при этом показано умение применять различные операторы языка при решении практических задач. Ответ логичен и изложен в терминах науки. Могут быть допущены 1-2 ошибки в определении основных понятий или синтаксические ошибки в написании программы, которые студент затрудняется исправить самостоятельно.	**C**	**75-71**	**4-**	
Дан недостаточно полный и развернутый ответ. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. Студент может конкретизировать обобщенные знания, доказав на примерах их основные положения только с помощью преподавателя. Студент испытывает затруднения при написании программы и не может связать теорию с практикой.	**D**	**70-66**	**3+**	**Пороговый уровень сформированности компетенций**
Дан неполный ответ, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий вследствие непонимания студентом их возможности использования при решении практических задач. В ответе отсутствуют выводы. Испытывает затруднения при выполнении практических задач, допускает логические ошибки при написании программ.	**E**	**65-61**	**3**	
Дан неполный ответ. Присутствует нелогичность изложения. Студент затрудняется с доказательностью. Допускает существенные ошибки в определениях терминов и понятий. В ответе отсутствуют выводы. Речь неграмотна. Отсутствует связь теории с практикой. При ответе на дополнительные вопросы студент начинает понимать связь между знаниями только после подсказки преподавателя.	**E**	**60**	**3-**	

Студент испытывает значительные трудности в ответе на вопросы. Присутствует масса существенных ошибок в определениях терминов, структур, характеристике инструкций, описания операторов языка программирования. Речь неграмотна. На дополнительные вопросы студент не отвечает. Задача не решена.	**F**	**Менее 60**	**2**	Компетенции не сформированы

Критерии оценки уровня усвоения знаний, умений и навыков по результатам зачета

Характеристика ответа	Европейская оценка	Рубежные баллы	Оценка	Уровень сформированности компетенций
Дан полный, развернутый ответ на поставленный теоретический вопрос зачета, показана совокупность осознанных знаний об объекте, проявляющаяся в свободном ориентировании понятиями, умении выделить существенные и несущественные его признаки, причинно-следственные связи. Знание об объекте демонстрируется на фоне понимания его в системе данной науки и междисциплинарных связей. Ответ формулируется в терминах науки, изложен литературным языком, логичен, доказателен, демонстрирует авторскую позицию студента. При ответе на экзаменационный билет студент демонстрирует применение знаний к реальным профессиональным ситуациям, объясняет решение задачи на уровне анализа, синтеза и дает свою оценку решения проблемы. Причем студент не затрудняется с ответом при видоизменении задания и правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.	A	100-96	зачтено	Повышенный уровень сформированности компетенций
Дан полный, развернутый ответ на поставленный теоретический вопрос, показана совокупность осознанных знаний об объекте, доказательно раскрыты основные положения темы; в ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Знание об объекте демонстрируется на фоне понимания его в системе данной науки и междисциплинарных связей. Умеет тесно увязывать теорию с практикой. Задача решена правильно и с обоснованием принятого решения. Ответ изложен литературным языком в терминах науки. Могут быть допущены недочеты в определении понятий, исправленные студентом самостоятельно в процессе ответа.	A	95-91	зачтено	

Дан полный, развернутый ответ на поставленный вопрос, доказательно раскрыты основные положения темы; в ответе прослеживается четкая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Задача решена верно, правильно обосновывает принятую методику решения задачи. Ответ изложен литературным языком в терминах науки. В ответе допущены недочеты, исправленные студентом с помощью преподавателя.	A	90-86	зачтено	
Дан полный, развернутый ответ на поставленный вопрос, показано умение выделить существенные и несущественные признаки, причинно-следственные связи. Ответ четко структурирован, логичен, изложен литературным языком в терминах науки. Студент владеет разносторонними навыками и приемами выполнения практических задач. Могут быть допущены недочеты или незначительные ошибки, исправленные студентом с помощью преподавателя.	B	85-81	зачтено	Базовый уровень сформированности компетенций
Дан полный, развернутый ответ на поставленный вопрос, показано умение выделить существенные и несущественные признаки, причинно-следственные связи. Ответ четко структурирован, логичен, изложен в терминах науки. Студент владеет разносторонними навыками и приемами выполнения практических задач. Ответы на дополнительные вопросы логичны, изложены в терминах науки, однако допущены незначительные ошибки или недочеты, исправленные студентом с помощью "наводящих" вопросов преподавателя.	C	80-76	зачтено	
Студент демонстрирует достаточные теоретические и практические знания. Дан полный, но недостаточно последовательный ответ на поставленный вопрос, но при этом показано умение выделить существенные и несущественные признаки и причинно-следственные связи. Ответ логичен и изложен в терминах науки. Могут быть допущены 1-2 ошибки в определении основных понятий или решении практической задачи, которые	C	75-71	зачтено	

студент затрудняется исправить самостоятельно.				
Дан недостаточно полный и развернутый ответ. Логика и последовательность изложения имеют нарушения. Допущены ошибки в раскрытии понятий, употреблении терминов. Студент не способен самостоятельно выделить существенные и несущественные признаки и причинно-следственные связи. Студент может конкретизировать обобщенные знания, доказав на примерах их основные положения только с помощью преподавателя. Речевое оформление требует поправок, коррекции. Студент испытывает затруднения при выполнении практической задачи и не может связать теорию с практикой.	D	70-66	зачтено	Пороговый уровень сформированности компетенций
Дан неполный ответ, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий, теорий, явлений, вследствие непонимания студентом их существенных и несущественных признаков и связей. В ответе отсутствуют выводы. Умение раскрыть конкретные проявления обобщенных знаний не показано. Испытывает затруднения при выполнении практических задач. Речевое оформление требует поправок, коррекции.	E	65-61	зачтено	
Дан неполный ответ. Присутствует нелогичность изложения. Студент затрудняется с доказательностью. Масса существенных ошибок в определениях терминов, понятий, характеристике фактов, явлений. В ответе отсутствуют выводы. Речь неграмотна. При ответе на дополнительные вопросы студент начинает понимать связь между знаниями только после подсказки преподавателя	E	60	зачтено	
Студент испытывает значительные трудности в ответе на зачетные вопросы. Присутствует масса существенных ошибок в определениях терминов, понятий, характеристике фактов, явлений экономической теории. Речь неграмотна. На дополнительные вопросы студент не отвечает. Задача не решена.	F	Менее 60	Не зачтено	Компетенции не сформированы