Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 13.11.2023 15:34:36
Уникальный программный ключ:
8db180d1a3f02ac9e60521a5672742735c18b1d6

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДЕНО
Декан факультета
Информационных технологий
_____ / Д.Г. Демидов /
«16» ___02___ 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Программирование и алгоритмизация на языках высокого уровня»

Направление подготовки
09.03.01 “Информатика и вычислительная техника”

Образовательная программа (профиль подготовки)
«Киберфизические системы»

Квалификация (степень) выпускника
Бакалавр

Форма обучения
Очная

Год набора: 2023

Москва 2023 г.

Программа дисциплины «**Программирование и алгоритмизация на языках высокого уровня**» составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению **09.03.01 “Информатика и вычислительная техника”** профилю подготовки **«Киберфизические системы».**

Программа актуализирована в 2023 году в связи с актуализацией учебного плана.

Разработчик:

к. ф.-м. н., доцент кафедры  / Т.Т. Идиатуллов /

Согласовано:
Заведующий кафедрой «СМАРТ-технологии»,
к.т.н., доцент  / Е.В. Петрунина /

1. Цели освоения дисциплины

Целью освоения дисциплины «Программирование и алгоритмизация на языках высокого уровня» является формирование системы знаний, умений и навыков в области основ алгоритмизации и прикладного программирования.

Задачи дисциплины: изучение принципов построения алгоритмов, изучение основ алгоритмических конструкций, изучение объектно-ориентированного языка программирования C#, изучение методов построения приложений, алгоритмов и структур данных, используемых при решении прикладных задач в различных предметных областях с применением электронно-вычислительных машин.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина «Программирование и алгоритмизация на языках высокого уровня» относится к дисциплинам базовой части (Блока 1) Б.1.1.8.9. основной образовательной программы бакалавриата; изучается во 2 семестре.

Дисциплина базируется на следующих, пройденных дисциплинах:

- Информационные технологии;
- Линейная алгебра;
- Математический анализ;
- Программирование и основы алгоритмизации.

3.Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Код компетенции	**В результате освоения образовательной программы обучающийся должен обладать**	**Перечень планируемых результатов обучения по дисциплине**
ОПК-8	способность разрабатывать алгоритмы и программы, пригодные для практического применения	**Знать:** - основы алгоритмизации (свойства алгоритмов, область применения алгоритмов); - методы построения алгоритмов; - структуры данных; - синтаксис и семантику универсального алгоритмического языка программирования высокого уровня; - основные принципы и методологию разработки прикладного программного обеспечения; - типовые способы организации программных данных; - подходы к построению программных алгоритмов. **Уметь:** - строить блок-схемы алгоритмов; - проводить анализ эффективности алгоритмов; - описывать алгоритмы с

		использованием графических языков моделирования. **Владеть:** - навыками составления алгоритмов и их представления в виде блок-схем; - навыками формализации прикладных задач.
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение	**Знать:** - инструменты разработки программного обеспечения; - виды структур данных; - диаграммы проектирования программного обеспечения; - стадии разработки программного обеспечения. **Уметь:** - уметь согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения; - получать программные реализации полученных решений на объектно-ориентированных языках программирования. **Владеть:** - навыками использования инструментальных программных средств в процессе разработки программного обеспечения; - навыками построения проектирования программного обеспечения; - навыками разработки программного обеспечения с использованием объектно-ориентированных подходов.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, 144 академических часов (из них 72 часа – аудиторная работа, в том числе 18 часов лекций, 54 часа лабораторных занятий, и 72 часа самостоятельной работы студента).

Дисциплина изучается во 2 семестре (1 курс). Форма контроля – экзамен, курсовой проект

Структура и содержание дисциплины «Программирование и алгоритмизация на языках высокого уровня» по срокам и видам работы отражены в приложении.

Содержание разделов дисциплины

Тематика лекционных занятий:

Тема 1. Введение. Предпосылки возникновения ООП и КСИ. Среда исполнения .NET. Введение в объектно-ориентированное программирование.

Тема 2. Структура языка C#. Управляющие конструкции языка высокого уровня. Блок (составной оператор).Пустой оператор. Операторы ветвления. Операторы цикла.

Тема 3. Объектно-ориентированное программирование на языке C#. Понятие объектно-ориентированного проектирования. Понятие объекта. Абстракция и инкапсуляция. Наследование. Полиморфизм. Достоинства и недостатки ООП. UML в разработке ООП-приложений.

Тема 4. Синтаксис описания класса. Спецификаторы описания. Элементы описания класса. Описание объекта. Присваивание и сравнение объектов. Данные: поля и константы. Методы. Способы передачи аргументов в метод. Методы с переменным количеством аргументов. Рекурсивные методы. Перегрузка методов. Свойства. Операции класса.

Тема 5. Перечисления и массивы. Перечислимый тип данных. Создание массива. Размещение массивов в памяти. Размерность массива. Действия с массивами. Оператор foreach. Сортировка массивов. Сортировка выбором. Класс Array. Использование методов класса Array. Ступенчатые массивы. Передача массивов как параметров метода.

Тема 6. Наследование. Синтаксис реализации наследования в описании класса. Конструкторы и наследование. Наследование полей и методов. Совместимость типов при наследовании. Раннее и позднее связывание. Полиморфизм. Применение виртуальных методов. Абстрактные классы. Бесплодные (финальные) классы. Класс object. Наследование и вложение.

Тема 7. Делегаты. Общие сведения об интерфейсе. Отличия интерфейса от абстрактного класса. Реализация интерфейса. Обращение к реализованному методу через объект типа интерфейса. Операция is. Операция as. Особенности реализации интерфейсов. Стандартные интерфейсы .NET. Сравнение объектов – интерфейс IComparable. Параметризованные интерфейсы. Клонирование объектов. Виды клонирования.

Тема 8. Структуры. Синтаксис описания структур. Реализация перечислений. Определение делегатов. Использование делегатов. Реализация обратного вызова (callback). Передача делегата через список параметров. События. Определение события. Механизм событий. Синтаксис описания события. Реализация подписки на событие. Применение делегатов и событий.

Тема 9. Прикладные аспекты разработки приложений на языке C# в среде Visual Studio. Работа с файлами на языке C#. Классы .NET для работы с потоками. Уровни обмена с внешними устройствами. Доступ к файлам. Чтение текстового файла. Чтение чисел из текстового файла. Работа с БД Access. Подключение к БД Access. Считывание данных и внесение изменений. Работа с COM-портом. Работа с изображениями.

Тема 10. *Графический пользовательский интерфейс. События. Таймеры*. Событийно-ориентированное программирование. Компонентно-ориентированное программирование и конструирование графического пользовательского интерфейса. Диаграммы последовательностей действий в описании функционирования интерфейса пользователя. Реализация элементов интерфейса с помощью визуального проектирования и программными средствами. Проблема реализация продолжительных операций в приложениях. Применение таймеров.

Тема 11. *Практические приемы реализации приложений в WinForms*. Построение приложений с использованием WinForms. Использование диалоговых окон выбора файла. Формирование отчетов в HTML. Вывод документа в WebBrowser. Компонент для отображения графиков. Извлечение изображения из Chart для размещения в документах и сохранения.

Тематика лабораторных работ

Лабораторно-практические работы

Лабораторно-практическая работа № 1. Создание приложений с графическим пользовательским интерфейсом на языке C# с использованием WinForms

Лабораторно-практическая работа № 2. Табличное представление данных на языке C#

Лабораторно-практическая работа № 3. Работа с базами данных в приложениях на языке C#

Лабораторно-практическая работа № 4. Создание приложений для работы с файлами на языке C#

Лабораторно-практическая работа № 5. Использование графических возможностей языка C#

Лабораторные работы (с индивидуальными заданиями)

Лабораторная работа № 1. Разработка консольных приложений на C#.

Лабораторная работа № 2. Использование циклических алгоритмов для обработки данных и для формирования консольных интерфейсов пользователя.

Лабораторная работа № 3. Построение GUI-приложений на C#. Элементы управления форм и функции.

Лабораторная работа № 4. Работа с базами данных средствами в приложениях на C#.

Лабораторная работа № 5. Использование графических возможностей приложении на C#.

Лабораторная работа № 6. Создание интерактивных приложений.

Для проведения лабораторных работ требуется компьютерный класс. объеденный в локальную сеть с выходом в Интернет. Компьютеры должны быть объединены локальной сетью. Необходим выход в глобальную сеть Интернет. Требуемое программное обеспечением: компилятор Visual Studio, платформа .NET или Mono, текстовый редактор, офисный пакет LibreOffice. Компьютерный класс должен иметь возможность обновления и установки дополнительного свободно распространяемого программного обеспечения.

Тематика вопросов для самостоятельного изучения

- Изучение тенденции применения различных языков программирования при решении разных практических задач.
- Изучение сред разработки, систем управления версиями.
- Изучение методов коллективной разработки.
- Изучение средств автоматизированного тестирования приложений.

5. Образовательные технологии

Методика преподавания дисциплины «Программирование и алгоритмизация на языках высокого уровня» и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения аудиторных и внеаудиторных занятий:

– аудиторные занятия: лекции, лабораторные работы, тестирование;

– внеаудиторные занятия: самостоятельное изучение отдельных вопросов, подготовка к лабораторным работам.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоении дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В процессе обучения в течение семестра используются оценочные средства текущего контроля успеваемости и промежуточных аттестаций. Применяются следующие оценочные средства: тест, защита лабораторных работ, зачет, экзамен.

Образцы тестовых заданий и вопросов к экзамену и зачету приведены в приложении 2.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю).

6.1.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

В результате освоения дисциплины (модуля) формируются следующие компетенции:

Код компетенции	**В результате освоения образовательной программы обучающийся должен обладать**
ОПК-8	способность разрабатывать алгоритмы и программы, пригодные для практического применения
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение.

В процессе освоения образовательной программы данная компетенция, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.2. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины (модуля), описание шкал оценивания.

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине (модулю).

ОПК-8 способность разрабатывать алгоритмы и программы, пригодные для практического применения				
Показатель	**Критерии оценивания**			
	2	**3**	**4**	**5**
<u>Знать:</u> Знать: - основы алгоритмизации (свойства алгоритмов,	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие	Обучающийся демонстрирует неполное соответствие следующих знаний:основ	Обучающийся демонстрирует частичное соответствие следующих знаний: основ	Обучающийся демонстрирует полное соответствие следующих знаний: основ

область применения алгоритмов); - методы построения алгоритмов; - структуры данных; - синтаксис и семантику универсального алгоритмического языка программирования высокого уровня; - основные принципы и методологию разработки прикладного программного обеспечения; - типовые способы организации программных данных; - подходы к построению программных алгоритмов.	следующих знаний: основ алгоритмизации; методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмического языка программирования высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов.	алгоритмизации; методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмического языка программирования высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	алгоритмизации; методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмического языка программирования высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов. Допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	алгоритмизации; методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмического языка программирования высокого уровня; основ принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов. Свободно оперирует приобретенными знаниями.
Уметь:	Обучающийся	Обучающийся	Обучающийся	Обучающийся

- строить блок-схемы алгоритмов; - проводить анализ эффективности алгоритмов; - описывать алгоритмы с использованием графических языков моделирования.	не умеет или в недостаточной степени умеет: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).	демонстрирует неполное соответствие следующих умений: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).	демонстрирует частичное соответствие следующих умений: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).	демонстрирует полное соответствие следующих умений:осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности алгоритмов; описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).
Владеть: - навыками составления алгоритмов и их представления в виде блок-схем; - навыками формализации прикладных задач.	Обучающийся не владеет или в недостаточной степени владеет навыками составления алгоритмов и их представления в виде блок-схем; навыками формализации прикладных задач.	Обучающийся владеет навыками составления алгоритмов и их представления в виде блок-схем; навыками формализации прикладных задач.	Обучающийся частично владеет навыками составления алгоритмов и их представления в виде блок-схем; навыками формализации прикладных задач.	Обучающийся в полном объеме владеет навыками составления алгоритмов и их представления в виде блок-схем; навыками формализации прикладных задач.
ПК-3 Способен разрабатывать требования и проектировать программное обеспечение.				
Знать: - инструменты разработки	Обучающийся демонстрирует	Обучающийся демонстрирует	Обучающийся демонстрирует	Обучающийся демонстрирует

программного обеспечения; - виды структур данных; - диаграммы проектирования программного обеспечения; - стадии разработки программного обеспечения.	полное отсутствие или недостаточное соответствие следующих знаний: инструментов разработки программного обеспечения; видов структур данных; диаграмм проектирования программного обеспечения; стадий разработки программного обеспечения.	неполное соответствие следующих знаний:инструментов разработки программного обеспечения; видов структур данных; диаграмм проектирования программного обеспечения; стадий разработки программного обеспечения. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	частичное соответствие следующих знаний:инструментов разработки программного обеспечения; видов структур данных; диаграмм проектирования программного обеспечения; стадий разработки программного обеспечения. Допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	полное соответствие следующих знаний:инструментов разработки программного обеспечения; видов структур данных; диаграмм проектирования программного обеспечения; стадий разработки программного обеспечения. Свободно оперирует приобретенными знаниями.
<u>Уметь:</u> - уметь согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения; - получать	Обучающийся не умеет или в недостаточной степени умеет согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании	Обучающийся демонстрирует неполное соответствие следующих умений: согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при	Обучающийся демонстрирует частичное соответствие следующих умений: согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при	Обучающийся демонстрирует полное соответствие следующих умений:согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при

программные реализации полученных решений на объектно-ориентированных языках программирования.	прикладного программного обеспечения; получать программные реализации полученных решений на языке программирования C#.	создании прикладного программного обеспечения; получать программные реализации полученных решений на языке программирования C#. Обучающийся испытывает значительные затруднения при оперировании умениями при их переносе на новые ситуации.	создании прикладного программного обеспечения; получать программные реализации полученных решений на языке программирования C#. Умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	создании прикладного программного обеспечения; получать программные реализации полученных решений на языке программировани я C#. Свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
Владеть: - навыками использования инструментальных программных средств в процессе разработки программного обеспечения; - навыками построения проектирования программного обеспечения; - навыками разработки программного обеспечения с использованием объектно-	Обучающийся не владеет или в недостаточной степени владеет навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками	Обучающийся владеет навыками навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками разработки программного обеспечения.	Обучающийся частично владеет навыками навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками разработки программного обеспечения, но	Обучающийся в полном объеме владеет навыками навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками разработки программного обеспечения,

ориентированн ых подходов.	разработки программного обеспечения.	Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	свободно применяет полученные навыки в ситуациях повышенной сложности.

Шкалы оценивания результатов промежуточной аттестации и их описание:

Форма промежуточной аттестации: экзамен.

Обязательными условиями подготовки студента к промежуточной аттестации является выполнение и защита студентом лабораторных работ, предусмотренных рабочей программой.

Шкала оценивания	**Описание**
Отлично	*Выполнены все виды учебной работы, предусмотренные рабочей программой. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.*
Хорошо	*Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.*
Удовлетворительно	*Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.*

Неудовлетворительно	*Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.*

Фонд оценочных средств представлен в приложении 2 к рабочей программе.

7. Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Разработка приложений для Windows 8 на языке C# / С. В. Пугачев, А. М. Шериев, К. А. Кичинский. — СПб.: БХВ-Петербург, 2013. — 416 с.: ил. — (Профессиональное программирование)

2. C# для школьников: Учебное пособие / М. Дрейер. Перевод с англ. под ред. В. Биллига — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009. — 128 с.: ил., табл. — (Лицей информационных технологий).

б) Дополнительная литература:

1. Троелсен. Э., C# и платформа .NET. Библиотека программиста. — СПб.: Питер, 2004. —796 с.: ил.

в) Электронные образовательные ресурсы:

1. Платформа цифрового образования Мосполитеха (ЭОР):
   – Программирование и алгоритмизация на языках высокого уровня
   https://online.mospolytech.ru/local/crw/course.php?id=13495
2. https://habrahabr.ru/
3. https://tproger.ru/tag/c-language/
4. https://prog-cpp.ru/c/

8. *Материально-техническое обеспечение дисциплины.*

Для проведения лабораторных работ требуется компьютерный класс. объеденный в локальную сеть с выходом в Интернет. Компьютеры должны быть объединены локальной сетью. Необходим выход в глобальную сеть Интернет. Требуемое программное обеспечением: компилятор VisualStudio, .NET или Mono, текстовый редактор, офисный пакет LibreOffice. Компьютерный класс должен иметь возможность обновления и установки дополнительного свободно распространяемого программного обеспечения.

9. Методические рекомендации для самостоятельной работы студентов

Самостоятельная работа студентов направлена на решение следующих задач:

Самостоятельная работа является одним из видов учебных занятий. Цель самостоятельной работы – практическое усвоение студентами вопросов автоматизации управления жизненным циклом изделия, рассматриваемых в процессе изучения дисциплины.

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Задачи самостоятельной работы студента:

- развитие навыков самостоятельной учебной работы;
- освоение содержания дисциплины;
- углубление содержания и осознание основных понятий дисциплины;
- использование материала, собранного и полученного в ходе самостоятельных занятий для эффективной подготовки к зачету.

Виды внеаудиторной самостоятельной работы:

- самостоятельное изучение отдельных тем дисциплины;
- подготовка к лекционным занятиям;
- подготовка к семинарам и практическим занятиям;
- оформление отчетов по выполненным лабораторным работам и подготовка к их защите;
- выполнение расчетно-графической работы.

Для выполнения любого вида самостоятельной работы необходимо пройти следующие этапы:

- определение цели самостоятельной работы;
- конкретизация познавательной задачи;
- самооценка готовности к самостоятельной работе;
- выбор адекватного способа действия, ведущего к решению задачи;
- планирование работы (самостоятельной или с помощью преподавателя) над заданием;
- осуществление в процессе выполнения самостоятельной работы самоконтроля (промежуточного и конечного) результатов работы и корректировка выполнения работы;
- рефлексия;
- презентация работы.

10. Методические рекомендации для преподавателя

На первом занятии по дисциплине необходимо ознакомить студентов с порядком ее изучения (темами курса, формами занятий, текущего и промежуточного контроля), раскрыть место и роль дисциплины в системе наук, ее практическое значение, довести до студентов требования к форме отчетности и применения видов контроля. Выдаются задания для подготовки к семинарским занятиям.

При подготовке **к семинарскому занятию** по перечню объявленных тем преподавателю необходимо уточнить план их проведения, продумать формулировки и содержание учебных вопросов, выносимых на обсуждение, ознакомиться с перечнем вопросов по теме семинара.

В ходе семинара во вступительном слове раскрыть практическую значимость темы семинарского занятия, определить порядок его проведения, время на обсуждение каждого учебного вопроса. Применяя фронтальный опрос дать возможность выступить всем студентам, присутствующим на занятии.

Целесообразно в ходе защиты лабораторных работ задавать выступающим и аудитории дополнительные и уточняющие вопросы с целью выяснения их позиций по существу обсуждаемых проблем.

Следует предоставить возможность выступления с места в виде кратких сообщений по подготовленному заранее вопросу.

В заключительной части семинарского занятия следует подвести его итоги: дать оценку выступлений каждого студента и учебной группы в целом. Раскрыть положительные стороны и недостатки проведенного семинарского занятия. Ответить на вопросы студентов. Выдать задания для самостоятельной работы по подготовке к следующему занятию.

Приложение 1.

Структура и содержание дисциплины «Программирование и алгоритмизация на языках высокого уровня» по направлению подготовки **09.03.01 «Информатика и вычислительная техника»** и профилю подготовки **«Киберфизические системы»**

n/n	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов					Формы аттестации	
				Л	П/С	Лаб	СРС	КСР	ПЛР*	СИ**	Т	Реферат	К/р	Э	З
	Тема 1. Введение. Предпосылки возникновения ООП и КСИ. Среда исполнения .NET. Введение в объектно-ориентированное программирование.	4	1	1			1								
	Тема 2. Структура языка C#. Управляющие конструкции языка высокого уровня. Блок (составной оператор). Пустой оператор. Операторы ветвления. Операторы цикла.	4	1	1			1								
	Лабораторная работа № 1. Разработка консольных приложений на C#.	4		1			1								
	Тема 3. Объектно-ориентированное программирование на языке C#. Понятие объектно-ориентированного проектирования. Понятие объекта. Абстракция и инкапсуляция. Наследование. Полиморфизм. Достоинства и недостатки ООП. UML в разработке ООП-приложений.	4				8	8								
	Тема 4. Синтаксис описания класса. Спецификаторы описания. Элементы описания класса. Описание объекта. Присваивание и сравнение объектов. Данные: поля и константы. Методы. Способы передачи аргументов в метод. Методы с переменным количеством аргументов. Рекурсивные методы.	4				8	8								

	Перегрузка методов. Свойства. Операции класса.														
	Лабораторная работа № 2. Использование циклических алгоритмов для обработки данных и для формирования консольных интерфейсов пользователя.	4				12	12								
	Тема 5. Перечисления и массивы. Перечислимый тип данных. Создание массива. Размещение массивов в памяти. Размерность массива. Действия с массивами. Оператор foreach. Сортировка массивов. Сортировка выбором. Класс Array. Использование методов класса Array. Ступенчатые массивы. Передача массивов как параметров метода.	4		1			1								
	Лабораторная работа № 3. Построение GUI-приложений на C#. Элементы управления форм и функции.	4				12	12								
	Тема 6. Наследование. Синтаксис реализации наследования в описании класса. Конструкторы и наследование. Наследование полей и методов. Совместимость типов при наследовании. Раннее и позднее связывание. Полиморфизм. Применение виртуальных методов. Абстрактные классы. Бесплодные (финальные) классы. Класс object. Наследование и вложение.	4		1			1								
	Лабораторная работа № 4. Работа с базами данных средствами в приложениях на C#.	4		1			1								
	Тема 7. Делегаты. Общие сведения об интерфейсе. Отличия интерфейса от абстрактного класса. Реализация интерфейса. Обращение к реализованному методу через объект типа интерфейса. Операция is. Операция as.	4		1			1								

	Особенности реализации интерфейсов. Стандартные интерфейсы .NET. Сравнение объектов – интерфейс IComparable. Параметризованные интерфейсы. Клонирование объектов. Виды клонирования.														
	Тема 8. Структуры. Синтаксис описания структур. Реализация перечислений. Определение делегатов. Использование делегатов. Реализация обратного вызова (callback). Передача делегата через список параметров. События. Определение события. Механизм событий. Синтаксис описания события. Реализация подписки на событие. Применение делегатов и событий.	4				8	8								
	Лабораторная работа № 5. Использование графических возможностей приложении на C#.	4		1			1								
	Тема 9. Прикладные аспекты разработки приложений на языке C# в среде Visual Studio. Работа с файлами на языке C#. Классы .NET для работы с потоками. Уровни обмена с внешними устройствами. Доступ к файлам. Чтение текстового файла. Чтение чисел из текстового файла. Работа с БД Access. Подключение к БД Access. Считывание данных и внесение изменений. Работа с COM-портом. Работа с изображениями.	4		1			1								
	Тема 10. Графический пользовательский интерфейс. События. Таймеры. Событийно-ориентированное программирование. Компонентно-ориентированное программирование и конструирование графического	4				12	12								

	пользовательского интерфейса. Диаграммы последовательностей действий в описании функционирования интерфейса пользователя. Реализация элементов интерфейса с помощью визуального проектирования и программными средствами. Проблема реализация продолжительных операций в приложениях. Применение таймеров.														
	Лабораторная работа № 6. Создание интерактивных приложений.	4		1			1								
	Тема 11. Практические приемы реализации приложений в WinForms. Построение приложений с использованием WinForms. Использование диалоговых окон выбора файла. Формирование отчетов в HTML. Вывод документа в WebBrowser. Компонент для отображения графиков. Извлечение изображения из Chart для размещения в документах и сохранения.	4		1			1								
	Итого в 4 семестре			***18***		***90***	***108***							+	
	Итого:			***18***		***90***	***108***							**+**	**+**

*ПЛР – написание отчета и подготовка к защите лабораторной работы
СИ** - самостоятельное изучение

Приложение 2

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

Направление подготовки:

09.03.01 “Информатика и вычислительная техника”

Профиль подготовки

«Киберфизические системы»

Форма обучения: очная

Кафедра «СМАРТ-технологии»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

Программирование и алгоритмизация на языках высокого уровня

Состав:

1. Паспорт фонда оценочных средств

2. Описание оценочных средств:

Перечень вопросов для экзамена

Перечень вопросов для зачета

Перечень вопросов для защиты лабораторных работ

Тестовые задания

Москва, 2023 год

1. **Паспорт фонда оценочных средств**

Таблица 1

ПОКАЗАТЕЛЬ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

ПРОГРАММИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ					
ФГОС ВО **09.03.01 «Информатика и вычислительная техника»**					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **компетенции**:					
КОМПЕТЕНЦИИ		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства****	**Степени уровней освоения компетенций**
ИНДЕКС	**ФОРМУЛИРОВКА**				
ОПК-8	способен разрабатывать алгоритмы и программы, пригодные для практического применения	***Знать:*** - основы алгоритмизации (свойства алгоритмов, область применения алгоритмов); - методы построения алгоритмов; - структуры данных; - синтаксис и семантику универсального алгоритмического языка программирования высокого уровня; - основные принципы и методологию разработки прикладного программного обеспечения; - типовые способы организации программных данных; - подходы к построению программных алгоритмов. ***Уметь:***	лекция, лабораторные работы, самостоятельная работа	ЗЛР, Т, З	**Базовый уровень:** воспроизводство полученных знаний в ходе текущего контроля; умение решать типовые задачи, принимать профессиональные решения по известным алгоритмам, правилам и методикам **Повышенный уровень:** практическое применение полученных знаний в процессе изучения дисциплины; готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении

		- строить блок-схемы алгоритмов; - проводить анализ эффективности алгоритмов; - описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод). ***<u>Владеть:</u>*** - навыками составления алгоритмов и их представления в виде блок-схем; - навыками формализации прикладных задач.			
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение.	**<u>Знать:</u>** - инструменты разработки программного обеспечения; - виды структур данных; - диаграммы проектирования программного обеспечения; - стадии разработки программного обеспечения среднего и крупного масштаба и сложности. **<u>Уметь:</u>** - уметь согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения;	лекция, лабораторные работы, самостоятельная работа	ЗЛР, Т, Э	**Базовый уровень:** воспроизводство полученных знаний в ходе текущего контроля; умение решать типовые задачи, принимать профессиональные решения по известным алгоритмам, правилам и методикам **Повышенный уровень:** практическое применение полученных знаний в процессе изучения дисциплины; готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные решения в

		- получать программные реализации полученных решений на объектно-ориентированных языках программирования. **<u>Владеть:</u>** - навыками использования инструментальных программных средств в процессе разработки программного обеспечения; - навыками построения проектирования программного обеспечения; - навыками разработки программного обеспечения с использованием объектно-ориентированных подходов..			условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении

2. **Перечень оценочных средств по дисциплине**

Программирование и алгоритмизация на языках высокого уровня

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Тест (Т)	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
2	ЗЛР	Средство проверки умений и навыков применять полученные знания для решения практических задач с помощью инструментальных средств.	Задания для защиты лабораторных работ
3	З	Средство контроля, организованное как специальная беседа педагогического работника с обучающимся на темы, связанные с практическим содержанием изучаемой дисциплиной, и совмещенное с оценкой проделанной в курсе работы	Вопросы по темам/разделам дисциплины/практическим и лабораторным занятия
4	Э	Средство контроля знаний, организованное как специальная беседа педагогического работника с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по всему объему изучаемого курса	Вопросы по темам/разделам дисциплины
	Курсовой проект (самостоятельная работа) (КР)	Средство проверки умений применять полученные знания для решения задач определенного типа по теме или разделу	Комплект индивидуальных заданий для курсового проектирования по вариантам
	Расчетно-графическая работа (РГР)	Средство проверки умений применять полученные знания по заранее определенной методике для решения задач или заданий по модулю или дисциплине в целом.	Комплект заданий для выполнения расчетно-графической работы

2.1. Перечень вопросов для экзамена

1. Контролируемые среды исполнения. Причины возникновения и назначение. Принципы функционирования платформы dotNET.
2. Основные алгоритмические конструкции языка C#. Структура консольного приложения. Ввод-вывод данных в консольном приложении. Организация интерактивных приложений без задержки на ожидание ввода пользователя команд.

3. Массивы элементов примитивных типов. Особенности динамических (доопределяемых в runtime) массивов в представлении C#. Объявление многомерных и ступенчатых массивов. Использование класса ArrayList для формирования “истинных” динамических массивов переменного размера.
4. Моделирование обработки двумерных наборов данных с применением циклических алгоритмов. Реализация вычисления детерминанта матрицы на C# для матрицы с заранее неизвестным размером.
5. Организация работы с видимостью объектов и описаний. Применение директив private, public, static, using. Модульность программного обеспечения.
6. Составные типы данных C# - структуры. Директива struct. Синтаксис объявления состава структуры. Псевдообъектная природа структур и управление памятью при их использовании.
7. Классы и объекты - экземпляры классов. Синтаксис объявления состава класса. Управление доступом к свойствам и методам класса. Создание объектов в памяти и особенности их размещения. Удаление объектов при автоматической “сборке мусора” и “ручная пометка” объекта как неиспользуемого. Использование средой исполнения “таблицы вызовов” для методов класса.
8. Использование функций сторонних библиотек в своих программах на C#. Импорт функций. Применение небезопасного кода. Применение классов обёрток для системных функций.
9. Работа с текстовыми файлами в C#. Формирование HTML отчётов по обработке данных и их отображение на форме приложения (WinForm).
10. Работа с Базами данных с C#. Подключение к СУБД и выполнение SQL-запросов. Организация дампа базы данных с известным набором таблиц в текстовый файл.
11. Использование графических возможностей в программах на C#. Работа с битовыми картами и их отображение в WinForm. Функции рисования и объект Graphics. Отображение графиков в Chart.

2.2. Пример экзаменационного билета

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Факультет информационных технологий, Кафедра СМАРТ-технологии
Дисциплина: Программирование и алгоритмизация на языках высокого уровня
Образовательная программа: Киберфизические системы (Управление в технических системах)

БИЛЕТ № 1

1. Контролируемые среды исполнения. Причины возникновения и назначение. Эмуляция и виртуализация в задачах безопасности и надежности вычислительных систем

2. Принципы работы в C# с объектами в памяти. Структура памяти. Работа с указателями. Сборка мусора. Удаление объектов при автоматической “сборке мусора” и “ручная пометка” объекта как неиспользуемого.

Заведующий кафедрой _____ Т.Т. Идиатуллов

2.3. Фонд тестовых заданий

1. Что такое функция?

a) Некоторая часть программы, содержащая описание переменных и констант основной программы
b) Некоторая часть программы, имеющая собственное имя и которая может вызываться из основной программы
c) Некоторая часть программы, содержащая вредоносный код, и блокирует определенные действия системы
d) Некоторая часть программы, в которой происходит начальная инициализация всех полей структур, массивов, переменных.

Правильный ответ: b

2. Что такое массив?

a) Именованный набор переменных имеющих различные типы данных, и располагающихся в одной памяти
b) Именованный набор переменных и функций, которые располагаются в одной области памяти
c) Именованный набор переменных имеющий один тип данных, и располагающихся в одной области памяти
d) Именованный набор переменных имеющих символьный тип данных, и располагающихся в одной области памяти

Правильный ответ: d

3. Как написать следующее выражение на языке С «Переменной a присвоено значение b»?

a) a==b
b) a=b
c) b=a
d) a:=b

Правильный ответ: b

4. Как написать следующее выражение «Второму элементу массива Myarray присвоено значение пяти »?

a) int [1] Myarray=«пять»
b) int Myarray [1] = 5
c) int Myarray [2] = «пять»
d) int Myarray [2] = 5

Правильный ответ: d

5. Как написать следующее выражение «Если переменная index больше size то мы инкрементируем переменную count »?

a) if (index>size) { count++; }
b) if (index<size) { count--; }
c) if (index>=size) { ++count; }
d) if (index<size) { --count; }

Правильный ответ: a

6. Какой диапазон значений имеет тип int для 32-разрядных вычислительных систем:

a) от 0 до 255
b) от -32768 до 32767
c) от 0 до 65535
d) от 0 до 4 294 967 295

Правильный ответ: b

7. Какой размер в байтах имеет переменная вещественного типа float

a) 2
b) 4
c) 8
d) 10

Правильный ответ: a

8. Дан массив int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } };. Чему будет равно значение элемента этого массива L[1][2]

a) 2
b) 3
c) 4
d) 8

Правильный ответ: d

9. Что называется прототипом функции?

a) описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров
b) описание функции, включая ее имя, тип возвращаемого значения, типы параметров
c) имя функции и тип возвращаемого значения
d) описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров, тело функции

Правильный ответ: c

10. В каких случаях необходимо использовать оператор return в теле функции?

a) Всегда
b) если необходимо, чтобы функция вернула значение
c) если необходимо обеспечить выход из функции в произвольном месте
d) если указан тип возвращаемого значения, в том числе и void

Правильный ответ: a

11. При открытии файла выполняется следующее действие:

a) физический файл связывается с логическим (файловой переменной)
b) устанавливается тип файла (текстовый или бинарный)
c) устанавливается вид (режим) использования файла
d) функцией открытия файла возвращается результат (ошибка)

Правильный ответ: a

12. Укажите группу, содержащую последовательность правильно записанных на языке С знаков операций отношений

a) ~ >, <, =, ?
b) =, <>, ><, >
c) =, >=, <=, !=
d) ~ =>,=<, =, <

Правильный ответ: d

2.4. Примеры индивидуальных заданий курсовых проектов

1. ***Интерактивная симуляция столкновений***
Разработать приложение для визуализации движения круглого объекта (мяча) в плоскости экрана с расчетом упругой модели столкновений с круглыми препятствиями и границами стенками зоны движения.

2. ***Интерактивная симуляция движения в коридоре***
Разработать приложение для визуализации движения круглого объекта (мяча) в плоскости экрана с расчетом упругой модели столкновений с препятствиями квадратной формы и границами стенками зоны движения

2.5. Примеры тем защиты лабораторных работ

Лабораторно-практические работы
Лабораторно-практическая работа № 1. Создание приложений с графическим пользовательским интерфейсом на языке C# с использованием WinForms
Лабораторно-практическая работа № 2. Табличное представление данных на языке C#
Лабораторно-практическая работа № 3. Работа с базами данных в приложениях на языке C#
Лабораторно-практическая работа № 4. Создание приложений для работы с файлами на языке C#

Лабораторно-практическая работа № 5. Использование графических возможностей языка C#

Лабораторные работы (с индивидуальными заданиями)

Лабораторная работа № 1. Разработка консольных приложений на C#.

Лабораторная работа № 2. Использование циклических алгоритмов для обработки данных и для формирования консольных интерфейсов пользователя.

Лабораторная работа № 3. Построение GUI-приложений на C#. Элементы управления форм и функции.

Лабораторная работа № 4. Работа с базами данных средствами в приложениях на C#.

Лабораторная работа № 5. Использование графических возможностей приложении на C#.

Лабораторная работа № 6. Создание интерактивных приложений.