

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Максимов Алексей Борисович

Должность: директор департамента по образовательной политике

Дата подписания: 30.05.2024 12:01:58

Уникальный идентификатор:

8db180d1a3f02ac9e60521a5672742735c18b1d6

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета

«Информационные технологии»



/ Д.Г.Демидов /

«15» февраля 2024г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Программирование и алгоритмизация на языках высокого
уровня»**

Направление подготовки
09.03.01 «Информатика и вычислительная техника»

Образовательная программа (профиль подготовки)
«Интеллектуальные беспилотные системы»

Год начала обучения:
2024

Квалификация (степень) выпускника
Бакалавр


Форма обучения
Очная

Москва – 2024

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 «**Интеллектуальные беспилотные системы**».

Составитель рабочей программы:

к. ф.-м. н., доцент кафедры

 / Т.Т. Идиатуллов /


Рабочая программа утверждена на заседании кафедры «СМАРТ-технологии»

Заведующий кафедрой,
к.т.н., доцент

 / Е.В. Петрунина /

Согласовано:

Заведующий кафедрой «СМАРТ-технологии»,
к.т.н., доцент

 / Е.В. Петрунина /

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы
2. Место дисциплины в структуре основной профессиональной образовательной программы высшего образования
3. Структура и содержание дисциплины
 - 3.1 Виды учебной работы и трудоемкость (по формам обучения)
 - 3.1.1 Очная форма обучения
 - 3.2 Тематический план изучения дисциплины (по формам обучения)
 - 3.2.1 Очная форма обучения
 - 3.3 Содержание дисциплины
 - 3.4 Тематика семинарских/практических и лабораторных занятий
4. Учебно-методическое и информационное обеспечение
 - 4.1 Основная литература
 - 4.2 Дополнительная литература
 - 4.3 Электронные образовательные ресурсы
 - 4.4 Лицензионное и свободно распространяемое программное обеспечение
 - 4.5 Современные профессиональные базы данных и информационные справочные системы
5. Материально-техническое обеспечение
6. Методические рекомендации
 - 6.1 Методические рекомендации для преподавателя по организации обучения
 - 6.2 Методические указания для обучающихся по освоению дисциплины
 - 6.3 Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
7. Фонд оценочных средств
 - 7.1 Методы контроля и оценивания результатов обучения
 - 7.2 Шкала и критерии оценивания результатов обучения
 - 7.3 Оценочные средства
 - 7.3.1 Текущий контроль на лабораторных занятиях
 - 7.3.2 Промежуточная аттестация (экзамен)
 - 7.3.3 Фонд тестовых заданий
 - 7.3.4 Примеры индивидуальных заданий курсовых проектов

1 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Выпускник, освоивший дисциплину, должен обладать следующими компетенциями:

Код и наименование компетенций	Индикаторы достижения компетенции
<p>ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения</p>	<p>ОПК-8.1 Знает:</p> <ul style="list-style-type: none"> - 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. <p>ОПК-8.2 Умеет:</p> <ul style="list-style-type: none"> - строить блок-схемы алгоритмов; - проводить анализ эффективности алгоритмов; - описывать алгоритмы с использованием графических языков моделирования. <p>ОПК-8.3 Владеет:</p> <ul style="list-style-type: none"> - навыками составления алгоритмов и их представления в виде блок-схем; - навыками формализации прикладных задач.
<p>ПК-3. Способен разрабатывать требования и проектировать программное обеспечение</p>	<p>ПК-3.1 Знает:</p> <ul style="list-style-type: none"> - инструменты разработки программного обеспечения; - виды структур данных; - диаграммы проектирования программного обеспечения; - стадии разработки программного обеспечения. <p>ПК-3.2 Умеет:</p> <ul style="list-style-type: none"> - 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. <p>ПК-3.1 Владеет:</p> <ul style="list-style-type: none"> - навыками использования инструментальных программных средств в процессе разработки программного обеспечения; - навыками построения проектирования программного обеспечения; - навыками разработки программного обеспечения с использованием объектно-ориентированных подходов.

2 Место дисциплины в структуре образовательной программы

Дисциплина «Программирование и алгоритмизация на языках высокого уровня» относится к дисциплинам базовой части (Блока 1) основной образовательной программы

бакалавриата; изучается во 2 семестре. Дисциплина базируется на следующих знаниях и навыках, приобретенных при освоении дисциплин:

- Программное обеспечение рабочего места оператора;
- Программирование и основы алгоритмизации;
- Математический анализ;
- Линейная алгебра.

3 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часа).

3.1 Виды учебной работы и трудоемкость (по формам обучения)

3.1.1 Очная форма обучения

№ п/п	Вид учебной работы	Количество часов	Семестры
			2
1	Аудиторные занятия	72	72
	В том числе:		
1.1	Лекции	18	18
1.2	Семинарские/практические занятия		
1.3	Лабораторные занятия	54	54
2	Самостоятельная работа	72	72
	В том числе:		
2.1	Подготовка к практическим занятиям	28	28
2.2	Тестирование	8	8
2.3	Выполнение курсового проекта	36	36
3	Промежуточная аттестация		
	Защита курсового проекта		
	Экзамен		
	Итого:	144/4	144/4

3.2 Тематический план изучения дисциплины (по формам обучения)

3.2.1 Очная форма обучения

№ п/п	Разделы/темы дисциплины	Трудоемкость, час					
		Всего	Аудиторная работа				Самостоятельная работа
			Лекции	Семинарские/практические занятия	Лабораторные занятия	Практическая подготовка	
1	Введение. Предпосылки возникновения ООП и КСИ. Среда исполнения .NET.		1		2		2

	Введение в объектно-ориентированное программирование.						
2	Структура языка C#. Управляющие конструкции языка высокого уровня. Блок (составной оператор). Пустой оператор. Операторы ветвления. Операторы цикла.		1		4		6
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.		2		6		8
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.		2		6		8
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.		2		6		8
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.		2		6		8
7	Делегаты. Общие сведения об интерфейсе. Отличия интерфейса от абстрактного класса. Реализация интерфейса. Обращение к реализованному методу через объект		2		6		8

	<p>типа интерфейса. Операция is. Операция as. Особенности реализации интерфейсов. Стандартные интерфейсы .NET. Сравнение объектов – интерфейс IComparable. Параметризованные интерфейсы. Клонирование объектов. Виды клонирования.</p>					
8	<p>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.</p>		2		6	8
9	<p>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 СОМ-портом. Работа с изображениями.</p>		1		4	4
10	<p>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 реализации продолжительных операций в приложениях. Применение таймеров.</p>		2		6	8
11	<p>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</p>		1		2	4

	отображения графиков. Извлечение изображения из Chart для размещения в документах и сохранения.						
Итого			18		54		90

3.3 Содержание дисциплины

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 реализации 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.

3.4 Тематика семинарских/практических и лабораторных занятий

Лабораторно-практические работы

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

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

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

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

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

Лабораторные работы (с индивидуальными заданиями)

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

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

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

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

Лабораторная работа № 5. Использование графических возможностей приложения на С#.

Лабораторная работа № 6. Создание интерактивных приложений.

Для проведения лабораторных работ требуется компьютерный класс, объединенный в локальную сеть с выходом в Интернет. Компьютеры должны быть объединены локальной сетью. Необходим выход в глобальную сеть Интернет. Требуемое программное обеспечение: 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.

3.5 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.

4 Учебно-методическое и информационное обеспечение

4.1 Основная литература

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

2. С# 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).

4.2 Дополнительная литература

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

4.3 Электронные образовательные ресурсы

1. «Программирование и алгоритмизация на языках высокого уровня» на платформе цифрового образования Мосполитеха (ЭОР):
<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/>

4.4 Лицензионное и свободно распространяемое программное обеспечение

1. СУБД MySQL (www.mysql.ru)
2. СУБД MSSQL 2008 (<http://www.microsoft.com/sqlserver/ru/ru/default.aspx>)
3. PyTorch
4. PyQT

4.5 Современные профессиональные базы данных и информационные справочные системы

1. Справочно-правовая системы «КонсультантПлюс: Некоммерческая интернетверсия»
<https://www.consultant.ru/online/>
2. Официальный интернет-портал правовой информации <http://pravo.gov.ru>
3. Российская национальная библиотека <http://www.nlr.ru>
4. ЭБС «Университетская библиотека онлайн» <https://biblioclub.ru/index.php>
5. Единое окно доступа к образовательным ресурсам Федеральный портал <http://window.edu.ru>
6. Научная электронная библиотека <http://www.elibrary.ru>
7. Российская государственная библиотека <http://www.rsl.ru>

5 Материально-техническое обеспечение

1. Компьютерные классы с оснащением: столы, стулья, аудиторная доска, использование переносного мультимедийного комплекса (переносной проектор, персональный ноутбук).
2. Персональные компьютеры, мониторы, мышки, клавиатуры. Рабочее место преподавателя: стол, стул.
3. Аудитория для самостоятельной работы.
4. Библиотека, читальный зал.

6 Методические рекомендации

6.1 Методические рекомендации для преподавателя по организации обучения

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.
2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.
3. При организации и проведения экзаменов в практико-ориентированной форме следует использовать утвержденные кафедрой Методические рекомендации.

6.2 Методические указания для обучающихся по освоению дисциплины

Получение углубленных знаний по дисциплине достигается за счет активной самостоятельной работы обучающихся. Выделяемые часы целесообразно использовать для знакомства с учебной и научной литературой по проблемам дисциплины, анализа научных концепций.

Промежуточная аттестация обучающихся в форме зачёта проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине проводится преподавателем, ведущим занятия по дисциплине методом экспертной оценки. По итогам промежуточной аттестации по дисциплине выставляется оценка «отлично», «хорошо», «удовлетворительно» или «не удовлетворительно».

К промежуточной аттестации допускаются только обучающиеся, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине «Программирование и алгоритмизация на языках высокого уровня».

6.3 Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья: - создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников, например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и ассимиляционных технологий;
- применение дистанционных образовательных технологий для передачи информации,

организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;

- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи: - продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут; - продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
- продолжительности выступления, обучающегося при защите курсовой работы - не более чем на 15 минут.

Для обеспечения подготовки людей в формате очной аудиторной работы с ограниченными возможностями движения выбираются аудитории с доступностью в рамках требований по организации безбарьерной среды движения.

7 Фонд оценочных средств

7.1 Методы контроля и оценивания результатов обучения

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
ОПК 8.1: Знать: - основы алгоритмизации (свойства алгоритмов, область применения алгоритмов); - методы построения алгоритмов; - структуры данных; - синтаксис и семантику универсального алгоритмического языка	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: основ алгоритмизации ; методов построения алгоритмов, структур данных; синтаксиса и семантики	Обучающийся демонстрирует неполное соответствие следующих знаний: основ алгоритмизации; методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмического языка	Обучающийся демонстрирует частичное соответствие следующих знаний: основ алгоритмизации; методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмического языка	Обучающийся демонстрирует полное соответствие следующих знаний: основ алгоритмизации; методов построения алгоритмов, структур данных; синтаксиса и семантики универсального алгоритмического языка

<p>программирования высокого уровня; - основные принципы и методологию разработки прикладного программного обеспечения; - типовые способы организации программных данных; - подходы к построению программных алгоритмов.</p>	<p>универсального алгоритмического языка программирования высокого уровня; основных принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов.</p>	<p>программирования высокого уровня; основных 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.</p>	<p>программирования высокого уровня; основных 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.</p>	<p>программирования высокого уровня; основных принципов и методологии разработки прикладного программного обеспечения; типовых способов организации программных данных; подходов к построению программных алгоритмов. Свободно оперирует приобретенными знаниями.</p>
<p>ОПК 8.2: Уметь: - строить блок-схемы алгоритмов; - проводить анализ эффективности алгоритмов; - описывать алгоритмы с использованием графических языков</p>	<p>Обучающийся не умеет или в недостаточной степени умеет: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности алгоритмов;</p>	<p>Обучающийся демонстрирует неполное соответствие следующих умений: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности</p>	<p>Обучающийся демонстрирует частичное соответствие следующих умений: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности</p>	<p>Обучающийся демонстрирует полное соответствие следующих умений: осуществлять синтез - строить блок-схемы алгоритмов; проводить анализ эффективности</p>

моделирования.	описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).	алгоритмов; описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).	алгоритмов; описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).	алгоритмов; описать алгоритм, используя ключевые слова языков программирования, но опуская подробности и специфический синтаксис (псевдокод).
ОПК 8.3: 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.				
ПК-3.1 Знать: - инструменты разработки программного обеспечения; - виды структур данных; - диаграммы проектирования программного обеспечения; - стадии разработки программного обеспечения.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие следующих знаний: инструментов разработки программного обеспечения; видов структур данных; диаграмм проектирования программного обеспечения; стадий	Обучающийся демонстрирует неполное соответствие следующих знаний: инструментов разработки программного обеспечения; видов структур данных; диаграмм проектирования программного обеспечения; стадий разработки программного обеспечения.	Обучающийся демонстрирует частичное соответствие следующих знаний: инструментов разработки программного обеспечения; видов структур данных; диаграмм проектирования программного обеспечения; стадий разработки программного обеспечения.	Обучающийся демонстрирует полное 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.	Допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Свободно оперирует приобретенными знаниями.
ПК-3.2 <u>Уметь:</u> - 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.	Обучающийся не умеет или в недостаточной степени умеет согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения; получать программные реализации полученных решений на языке программирования C#.	Обучающийся демонстрирует неполное соответствие следующих умений: согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения; получать программные реализации полученных решений на языке программирования C#. Обучающийся испытывает значительные затруднения при оперировании	Обучающийся демонстрирует частичное соответствие следующих умений: согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения; получать программные реализации полученных решений на языке программирования C#. Умения освоены, но допускаются незначительные ошибки, неточности,	Обучающийся демонстрирует полное соответствие следующих умений: согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения; получать программные реализации полученных решений на языке программирования C#. Свободно оперирует приобретенными умениями, применяет их в ситуациях

		умениями при их переносе на новые ситуации.	затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	повышенной сложности.
ПК-3.3 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.	Обучающийся не владеет или в недостаточной степени владеет навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками разработки программного обеспечения.	Обучающийся владеет навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками разработки программного обеспечения. Обучающийся испытывает значительные затруднения при применении навыков в новых ситуациях.	Обучающийся частично владеет навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками разработки программного обеспечения, но допускаются незначительные ошибки, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.	Обучающийся в полном объеме владеет навыками использования инструментальных программных средств в процессе разработки программного обеспечения; навыками построения проектирования программного обеспечения; навыками разработки программного обеспечения, свободно применяет полученные навыки в ситуациях повышенной сложности.

7.2 Шкала и критерии оценивания результатов обучения

Критерии оценки ответа на экзамене

Шкала оценивания	Описание
Отлично	Выполнены все виды учебной работы, предусмотренные

	<i>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.</i>
<i>Хорошо</i>	<i>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 незначительные ошибки.</i>
<i>Удовлетворительно</i>	<i>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.</i>
<i>Неудовлетворительно</i>	<i>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.</i>

Критерии оценки работы обучающегося при защите курсового проекта

Шкала оценивания	Описание
<i>Отлично</i>	<i>Выполнены все задачи, предусмотренные индивидуальным заданием на курсовое проектирование. В тексте работы и во время защиты 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.</i>
<i>Хорошо</i>	<i>Выполнены все задачи, предусмотренные индивидуальным заданием на курсовое проектирование. В тексте работы и во время защиты студент</i> 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 незначительные ошибки.
<i>Удовлетворительно</i>	<i>Выполнены все задачи, предусмотренные индивидуальным заданием на курсовое проектирование. В тексте работы и во время защиты студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка</i>

	<i>или неточность.</i>
<i>Неудовлетворительно</i>	<i>Не выполнен один или более видов учебной работы, предусмотренных задачами курсового проектирования. В тексте работы и во время защиты студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателям, 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.</i>

Критерии оценки работы обучающегося на лабораторных занятиях

Шкала оценивания	Описание
<i>Отлично</i>	<i>Выполнены все практические задания, предусмотренные практическими занятиями, обучающийся четко и без ошибок ответил на все контрольные вопросы, активно работал на практических занятиях.</i>
<i>Хорошо</i>	<i>Выполнены все практические задания, предусмотренные практическими занятиями, обучающийся с корректирующими замечаниями преподавателя ответил на все контрольные вопросы, достаточно активно работал на практических занятиях.</i>
<i>Удовлетворительно</i>	<i>Выполнены все практические задания, предусмотренные практическими занятиями с замечаниями преподавателя; обучающийся ответил на все контрольные вопросы с замечаниями.</i>
<i>Неудовлетворительно</i>	<i>Обучающийся не выполнил или выполнил неправильно практические задания, предусмотренные практическими занятиями; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.</i>

Критерии оценки тестирования

Тестирование оценивается в соответствии с процентом правильных ответов, данных студентом на вопросы теста. Стандартная шкала соответствия результатов тестирования выставяемой балльной оценке:

Шкала оценивания	Описание
<i>Отлично</i>	<i>Свыше 85% правильных ответов (включительно);</i>
<i>Хорошо</i>	<i>От 70 % до 84,9 % правильных ответов;</i>
<i>Удовлетворительно</i>	<i>От 55 % до 69,9 % правильных ответов;</i>
<i>Неудовлетворительно</i>	<i>Менее 54,9 % правильных ответов.</i>

7.3 Оценочные средства

7.3.1 Текущий контроль на лабораторных занятиях

Пример задания текущего контроля:

1. Составить программу вычисления Y :

$Y = 3 * x^2 - 1/x$ при $x > 1$,

$Y = x + 5$ при $x \leq -5$

Напечатать:

При $x = \dots$ функция вычисляется по формуле \dots Результат = \dots

7.3.2 Промежуточная аттестация (экзамен)

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 SMART-технологии
Дисциплина: Программирование и алгоритмизация на языках высокого уровня
Образовательная программа: Интеллектуальные беспилотные системы

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 _____ / _____

Пример заданий рубежного контроля

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`.

7.3.3 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 имеющих 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 C «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 C знаков операций отношений

- a) `~ >, <, =, ?`
- b) `=, <>, ><, >`

c) =, >=, <=, !=

d) ~ =>, =<, =, <

Правильный ответ: d

7.3.4 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