

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Наумова Наталия Александровна
Должность: Ректор
Дата подписания: 24.10.2024 14:21:41
Уникальный программный ключ:
6b5279da4e034bffa79172803da5b7b559fc69e7

МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ
Государственное образовательное учреждение высшего образования Московской области
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ОБЛАСТНОЙ УНИВЕРСИТЕТ
(МГОУ)

Физико-математический факультет
Кафедра вычислительной математики и методики преподавания информатики

Согласовано управлением организации
и контроля качества образовательной
деятельности

« 10 » 06 2020 г.
Начальник управления
/М.А. Миненкова/

Одобрено учебно-методическим советом

Протокол « 10 » 06 2020 г. № 7
Председатель
/Т.Е. Суслин/



Рабочая программа дисциплины
Языки и методы программирования

Направление подготовки
44.03.01 Педагогическое образование

Профиль:
Информатика

Квалификация
Бакалавр

Форма обучения
Очная

Согласовано учебно-методической
комиссией физико-математического
факультета:

Протокол « 10 » 06 2020 г. № 10
Председатель УМКом
/ Барabanова Н.Н. /

Рекомендовано кафедрой
вычислительной математики и методики
преподавания информатики

Протокол « 10 » 06 2020 г. № 10
Зав. кафедрой
/ Шевчук М.В. /

Мытищи
2020

Грамаков Дмитрий Анатольевич
доцент, кандидат педагогических наук, доцент кафедры вычислительной математики и
методики преподавания информатики.

Рабочая программа дисциплины «Языки и методы программирования» составлена в соответствии с требованиями Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 44.03.01 Педагогическое образование профиль «Информатика» утвержденная приказом МИНОБРНАУКИ РОССИИ от 22.02.18 № 121

Дисциплина входит в обязательную часть блока Б1 «Дисциплины (модули)» и является обязательной для изучения.

Год начала подготовки 2020

СОДЕРЖАНИЕ

1. Планируемые результаты обучения	4
2. Место дисциплины в структуре образовательной программы	5
3. Объем и содержание дисциплины	6
4. Учебно-методическое обеспечение самостоятельной работы обучающихся	9
5. Фонд оценочных средств для проведения текущей и промежуточной аттестации по дисциплине	11
6. Учебно-методическое и ресурсное обеспечение дисциплины	37
7. Методические указания по освоению дисциплины	39
8. Информационные технологии для осуществления образовательного процесса по дисциплине	41
9. Материально-техническое обеспечение дисциплины	42

1. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

1.1. Цель и задачи дисциплины

Целями освоения дисциплины «Языки и методы программирования» является теоретическая и практическая подготовка студентов в области разработки современных приложений с использованием различных языков программирования

Задачи дисциплины:

1. Формирование устойчивого интереса к изучаемой дисциплине, развитие мировоззрения и творческого потенциала, позволяющего будущему специалисту с минимальными затратами осуществлять доступ к требуемым информационным ресурсам.

2. Организация процесса обучения для развития практических навыков в области информационных технологий и программирования при решении различных инженерных и прикладных задач.

3. Планирование и обеспечение условий обучения, направленных на мобилизацию индивидуальных возможностей и ресурсов обучаемых в рамках самостоятельной работы

1.2. Планируемые результаты обучения

В результате освоения данной дисциплины у обучающихся формируются следующие компетенции:

УК-1 - Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Языки и методы программирования» относится к обязательной части блока 1 и является обязательной дисциплиной для изучения.

Для освоения дисциплины «Программирование на языке Ассемблера» студенты используют знания, умения, навыки, полученные и сформированные в ходе изучения предмета «Информатика и ИКТ» в общеобразовательной школе.

Ее изучение базируется на знаниях студентами таких дисциплин, как: «Практикум решения задач на ЭВМ», «Линейная алгебра», «Программное обеспечение ЭВМ».

Изучение дисциплины «Языки и методы программирования» является базой для дальнейшего освоения студентами дисциплин «Компьютерное моделирование», «Теоретические основы информатики», «Архитектура вычислительных систем», дисциплин по выбору, прохождения учебной практики.

3. ОБЪЕМ И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1. Объем дисциплины

Показатель объема дисциплины	Форма обучения		
	Очная	Заочная	Очно-заочная
Объем дисциплины в зачетных единицах	12		
Объем дисциплины в часах	432		
Контактная работа:	226,6		
Лекции	78		
Лабораторные работы	144		
Контактные часы на промежуточную аттестацию:	4,6		
Экзамен	0,6		
Предэкзаменационная консультация	4		
Самостоятельная работа	186		
Контроль	19,4		

Формой промежуточной аттестации являются экзамены в 6 и 7 семестрах.

3.2. Содержание дисциплины

Наименование разделов (тем) дисциплины с кратким содержанием	Количество часов			
	Лекции	Семинарские занятия	Практические занятия	Лабораторные занятия
Раздел 1. Введение в языки программирования				
Тема 1. Основы языка программирования C#. Введение в C#. Язык C# и платформа .NET. Начало работы с Visual Studio. Первая программа. Структура программы. Компиляция в командной строке. Установка версии языка. Основы программирования на C#. Алгоритмическое программирование. Переменные. Литералы. Типы данных. Консольный ввод-вывод. Арифметические операции. Поразрядные операции. Операции присваивания. Преобразования базовых типов данных. Условные выражения. Условные конструкции. Циклы. Массивы. Программа сортировки массива. Методы. Параметры методов. Передача	30			46

параметров по ссылке и значению. Выходные параметры. Массив параметров и ключевое слово params. Область видимости (контекст) переменных. Рекурсивные функции. Перечисления enum. Кортежи. Классы. Объектно-ориентированное программирование. Классы и объекты. Структуры. Типы значений и ссылочные типы. Модификаторы доступа. Свойства и инкапсуляция. Перегрузка методов. Статические члены и модификатор static. Константы и поля для чтения. Перегрузка операторов. Значение null. Индексаторы. Наследование. Преобразование типов. Перегрузка операций преобразования типов. Виртуальные методы и свойства. Соккрытие методов. Раннее и позднее связывание. Абстрактные классы. Класс System.Object и его методы. Обобщенные типы. Ограничения обобщений. Наследование обобщенных типов. . Обработка исключений. Конструкция try..catch..finally. Блок catch и фильтры исключений. Типы исключений. Класс Exception. Создание классов исключений. Поиск блока catch при обработке исключений. Генерация исключения и оператор throw. Интерфейсы. Введение в интерфейсы. Интерфейсы в обобщениях. Копирование объектов. Интерфейс ICloneable. Сортировка объектов. Интерфейс IComparable. Делегаты, события и лямбды. Делегаты. Применение делегатов. События. Анонимные методы. Лямбды. Коллекции. Введение в коллекции. Необобщенные коллекции. ArrayList. Обобщенные коллекции. Список List<T>. Двухсвязный список LinkedList<T>. Очередь Queue<T>. Стек Stack<T>. Словарь Dictionary<T, V>. Класс ObservableCollection. Интерфейсы IEnumerable и IEnumerator. Итераторы и оператор yield. Работа с потоками и файловой системой. Работа с дисками. Работа с каталогами. Работа с файлами. Классы File и FileInfo. Чтение и запись файла. Класс FileStream. Чтение и запись текстовых файлов. StreamReader и StreamWriter. Работа с бинарными файлами. BinaryWriter и BinaryReader. Работа со строками. Строки и класс System.String. Операции со строками. Форматирование и интерполяция строк. Класс StringBuilder. Регулярные выражения. Основы LINQ. Фильтрация выборки и проекция. Сортировка. Работа с множествами. Агрегатные операции. Методы Skip и Take. Группировка. Соединение коллекций. Метод Join, GroupJoin и Zip. Методы All и Any. Отложенное и немедленное выполнение LINQ.				
Тема 2. Основы языка программирования Python Введение в программирование на языке Python. Что такое Python?. История языка Python. Почему программисты используют Python?. Что можно делать с помощью Python?	20			34

Сильные стороны Python. Философия Python. Версии Python 2.x и 3.x. Первая программа Python. Базовые типы данных в языке Python. Decimal и Fraction. Кортежи. Работа с кортежами. Словари. Операции со словарями. Множества. Операции с множествами. Методы, изменяющие множества. Frozenset. Файлы. Методы работы с файлами. Генераторы. Списки. Методы работы со списками. Множество. Работа с множествами. Оператор If. Цикл for ... in. Функция range(). Цикл while. Строки. Форматирование строк. Строковые методы. Функции. Анонимные функции. Ввод данных пользователя: input(). Интерактивные циклы. Обработка ошибок проверкой ввода. Исключения. Встроенные функции map, filter, reduce. Модули. Как организована программа. Импорт. Пути поиска модулей. Import и from. Пространство имен модуля. Двойное импортирование. Атрибуты модулей. Пакеты. Классы. Создание объектов. Наследование. Метод-конструктор __init__. Инкапсуляция. Вызов методов предков. Вызов конструктора суперкласса. Преимущества ООП. Внутренние методы классов. Регулярные выражения. Базовые шаблоны. Метасимволы повторения. Наборы символов. Работа с группами. Отладка регулярных выражений. Компиляция регулярных выражений. Файловая система. Модули os, os.path, shutil. Модуль urllib. Стандартная библиотека. Визуализация данных с помощью пакета Mathplotlib.				
Раздел 2. Основы разработки приложений с графическим интерфейсом				
Тема 3 Создание графических приложений на C# Введение в Windows Forms. Работа с формами. Контейнеры в Windows Forms. Элементы управления. Меню и панели инструментов.	6			16
Тема 4. Основы графических приложений на Python Разработка GUI-приложений. Создание графического интерфейса. Tkinter. Создание окна приложения. Кнопки. Изменение свойств элементов. Позиционирование элементов. Текстовая метка Label. Поле ввода Entry. Checkbutton. Radiobutton. Listbox. Меню. Графические библиотеки: PyQt, PyGTK, wxPython, Pygames.	4			12
Раздел 3. Основы разработки веб-приложений				
Тема 5. Языки HTML5 и CSS3 Интернет. Базовые службы (услуги) Интернета. Протоколы Интернет. стек TCP/IP. Протоколы Интернета прикладного уровня. HTML5. Элемент. Тег. Атрибуты. Основные элементы	6			

версии 4.01. Новые элементы HTML5. Различие в подходах спецификаций в HTML 4.01 и HTML5. Новые семантические элементы. Формы HTML и ввод данных Новые свойства форм. Поддержка видео и аудио. Элементы <video> и <audio>. Элемент <canvas>. Технология SVG. CSS. Определение правил стилей. Селекторы. Типы селекторов. Традиционная блочная модель. Верстка. Боксы. Типы боксов. Строчные боксы. Блочные боксы. Поток и расположение элементов. Позиционирование. Позиционирование контента с помощью свойства float. Flexbox. Основы flexbox-разметки. Свойства flex-контейнера. Свойства flex-элементов. Новые свойства CSS3. Основы Grid-разметки.				20
Тема 6 Язык программирования JavaScript JavaScript. Синтаксис. Типы данных. Операторы. Локальные и глобальные переменные. Функции. Контекст функции. Вызов функции. Объектно-ориентированное программирование. Ссылочные типы. Объекты в JavaScript. Создание объекта. Точечная и скобочная нотация. Прототип. Массивы. Способы создания массивов. Методы для работы с массивами. Введение в библиотеку jQuery. Ненавязчивый JavaScript. Селекторы jQuery. Основы работы с DOM. Объектная модель документа. Методы работы с элементами и событиями. Библиотеки и фреймворки. Преимущества использования JavaScript-фреймворков и библиотек. AngularJS. Общая характеристика. Модель MVC. Bootstrap. Основные инструменты Bootstrap. Адаптивный дизайн. (responsive web-design). Назначение и использование.	12			16
Итого	78			144

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Целью самостоятельной работы является углубление понимания и улучшение усвоения курса лекций и лабораторных работ, подготовка к выполнению контрольных работ, к сдаче экзаменов.

Специфика курса «Языки и методы программирования» ориентирует студентов на активную самостоятельную работу:

- овладение приемами работы с базовым конструкциями языков программирования для решения широкого класса задач программирования;
- приобретение пользовательских навыков в области программирования различных приложений;

- совершенствование умений работы с различными парадигмами программирования;
- изучение современных приемов разработки программных приложений;
- самостоятельный выбор индивидуального задания в соответствии с возможностями и интересом;
- самостоятельная разработка алгоритмов при решении предложенных задач;
- тестирование и отладка приложений;
- слежение за развитием передовых подходов в области информационно-коммуникационных технологий;
- анализ учебных пособий по программированию, алгоритмам и методам разработки различных приложений;
- самостоятельное знакомство (изучение) с постоянно обновляемой литературой в области программирования через глобальную сеть Интернет.

Самостоятельную работу на лабораторных занятиях можно организовать за счет выбора студентом индивидуального задания, самостоятельного решения поставленных задач, выполнения предлагаемых согласно варианту заданий, составления итогового отчета о проделанной работе. На лекциях - дискуссия, обсуждение мнений студентов. На экзамене - проверка ознакомления студентов с литературой.

Формы и методы самостоятельной работы студентов и её оформление:

- конспектирование изучаемой литературы - краткое изложение материала по программированию, алгоритмам и методам из предложенных источников, а также из источников, которые студенты находят самостоятельно согласно предложенной тематике, тематических веб-сайтов, электронных учебников и т.д.; конспект должен быть достаточно кратким и точным, обобщать основные положения авторов, при этом упор делается на программный код и особенности его исполнения;
- подготовка отчета по выполненным заданиям с указанием результатов выполнения разработанных приложений, с различными тестовыми данными и комментариями программного кода, который должен показывать активное понимание алгоритма решения задачи и особенностей применения различных программных конструкций, а также библиотечных классов, используемых при разработке приложений.

С целью оптимизации учебного процесса рекомендуется на первом занятии сообщить студентам общую тематику занятий, цели и задачи курса, темы самостоятельной работы и примерный перечень вопросов к экзамену по дисциплине, а также обозначить особенности проведения экзамена и промежуточного контроля. В процессе изучения курса необходимо постоянное использование возможностей глобальной сети Интернет с целью привлечения материалов профильных сайтов, а также изучения базовых возможностей языков программирования..

№	Темы для самостоятельного изучения	Изучаемые вопросы	Кол-во часов	Формы самостоятельной работы	Методическое обеспечение	Формы отчетности
	Парадигмы программирования	Назначение и базовые функции.	16	Работа с литературой и сетью	Рекомендуемая литература.	Отчет в формате Word.

№	Темы для самостоятельного изучения	Изучаемые вопросы	Кол-во часов	Формы самостоятельной работы	Методическое обеспечение	Формы отчетности
		Общие принципы и приемы работы		Интернет	Ресурсы Интернет	Код приложений
	Алгоритмическое программирование. Базовые конструкции	Назначение и базовые функции. Общие принципы и приемы работы	40	Работа с литературой и сетью Интернет	Рекомендуемая литература. Ресурсы Интернет	Отчет в формате Word. Код приложений
	Объектно-ориентированное программирование	Назначение и базовые функции. Общие принципы и приемы работы	40	Работа с литературой и сетью Интернет	Рекомендуемая литература. Ресурсы Интернет	Отчет в формате Word. Код приложений
	Приложения с графическим интерфейсом	Назначение и базовые функции. Общие принципы и приемы работы	30	Работа с литературой и сетью Интернет	Рекомендуемая литература. Ресурсы Интернет	Отчет в формате Word. Код приложений
	Веб-приложения	Назначение и базовые функции. Общие принципы и приемы работы	60	Работа с литературой и сетью Интернет	Рекомендуемая литература. Ресурсы Интернет	Отчет в формате Word. Код приложений
	Итого		186			

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

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

Изучение дисциплины «Языки и методы программирования» позволяет

сформировать у бакалавров следующие компетенции.

Код и наименование компетенции	Этапы формирования
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	1. Работа на учебных занятиях 2. Самостоятельная работа

5.2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Оцениваемые компетенции	Уровень сформированности	Этап формирования	Описание показателей	Критерии и оценивания	Шкала оценивания
УК-1	Пороговый	1. Работа на учебных занятиях 2. Самостоятельная работа	Знает и понимает: • закономерности поиска, критического анализа и синтеза информации для решения широкого круга задач, связанных с программированием; • особенности системного подхода для решения широкого спектра задач в области программирования; Умеет: • осуществлять поиск, критический анализ и синтез методов для задач обучения в области программирования; • применять системный подход для решения задач в области программирования.	Посещение, тестовые задания, задания по сам работе, домашние задания, экзамен	41-60
	Продвинутый	1. Работа на учебных занятиях Самостоятельная работа	Знает и понимает: • закономерности поиска, критического анализа и синтеза	Посещение, тестовые	61-100

			информации для решения широкого круга задач, связанных с программированием; • особенности системного подхода для решения широкого спектра задач в области программирования; Умеет: • осуществлять поиск, критический анализ и синтез методов для задач обучения в области программирования; • применять системный подход для решения задач в области программирования. <i>Владеет (навыками и/или опытом деятельности):</i> • навыками осуществления поиска, критического анализа и синтеза информации для решения широкого круга задач обучения программированию; • навыками применения системного подхода для решения широкого круга задач в области программирования и обучения программированию.	задания, задания по сам работе, домашние задания, экзамен	
--	--	--	---	--	--

5.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Темы рефератов

Требования к реферату. Объем реферата не более 4 страниц текста содержащего примеры кода по теме реферата, не должна включаться общая информация из кода

программы.

1. Написать программы, демонстрирующие возможности циклов for и while в C#
2. Написать программы, демонстрирующие возможности операторов управления C#.
3. Программный код, демонстрирующий основы ООП в C#.
4. Реализация делегатов при создании приложений с графическим интерфейсом для языка C#.
5. Продемонстрировать использование переменных и типов данных в C#.
6. Программно показать основные операции с числовыми типами в C#.
7. Основные методы класса Math в Base Class Library (примеры программ).
8. Операции со строками в классе String (примеры программ)
9. Основы использования статических методов в C# (примеры программ).
10. Примеры использования структур в C#.
11. Переменные и литералы – примеры программного кода
12. Примеры загрузки консоли ввода-вывода.
13. Арифметические операции – примеры программного кода
14. Примеры использования поразрядных операций.
15. Примеры использования массивов.
16. Методы в C# - примеры описания.
17. Примеры реализации методов с передачей параметров по ссылке и значению.
18. Примеры использования перечисления enum.
19. Использование кортежей в C#.
20. Примеры использования null.
21. Примеры использования индексов.
22. Виртуальные методы и свойства – примеры описания и использования.
23. Примеры использования раннего и позднего связывания.
24. Абстрактные классы - примеры программного кода.
25. Класс System.Object и его методы – примеры использования.
26. Обобщенные типы и их роль – примеры программного кода.
27. Делегаты, события и лямбды – примеры программного кода.
30. Работа с потоками и файловой системой - примеры программного кода.
31. Работа с дисками и с каталогами - примеры программного кода.
32. Основы LINQ – базовые операции - примеры программного кода.
33. Фильтрация выборки и проекция - примеры программного кода.
34. Базовые типы данных в языке Python - примеры программного кода.
35. Работа с кортежами - примеры программного кода.
36. Операции со словарями - примеры программного кода.
37. Операции с множествами - примеры программного кода.
38. Методы, изменяющие множества - примеры программного кода.
39. Методы работы с файлами - примеры программного кода.
40. Методы работы со списками - примеры программного кода.
41. Работа с множествами - примеры программного кода.
42. Функция range() - примеры программного кода.
43. Форматирование строк - примеры программного кода.
44. Строковые методы - примеры программного кода.
45. Анонимные функции - примеры программного кода.
46. Ввод данных пользователя input() - примеры программного кода.

- 47. Исключения - примеры программного кода.
- 48. Встроенные функции `map`, `filter`, `reduce` - примеры программного кода.
- 49. Модули. Пространство имен модуля - примеры программного кода.
- 50. Вызов методов предков - примеры программного кода.
- 51. Вызов конструктора суперкласса - примеры программного кода.
- 52. Внутренние методы классов - примеры программного кода.
- 53. Базовые шаблоны регулярных выражений - примеры программного кода.

Примеры тестовых заданий для текущего контроля:

1. Если `a` равно 7.5 и `b` равно 7, тогда значение `a >= b` равняется `false`

Да

Нет

2. Какого типа будет результат

```
Console.WriteLine(12 % 2);    // _____  
Console.WriteLine(-15 % 2);   // _____  
Console.WriteLine(1555.0f % 2.2f); // _____  
Console.WriteLine(235.0m % 2.2m); // _____  
Console.WriteLine(-65.2 % 2.0); // _____
```

3. Результат выполнения следующего кода

```
int a = 5;  
a /= 6;  
Console.WriteLine(a);
```

4. Результат выполнения следующего кода

```
double x;  
x = 1.5;  
Console.WriteLine(++x);  
Console.WriteLine(x);
```

5. Какой результат вычисления данного выражения?

```
list(range(5, -7, -2))
```

- `[-7, -5, -3, -1, 1, 3]`
- `[-6, -4, -2, 0, 2, 4]`
- `[5, 3, 1, -1, -3, -5, -7]`
- `[5, 3, 1, -1, -3, -5] ***`
- `[3, 1, -1, -3, -5, -7]`

6. Какой результат вычисления данного выражения?

```
list(range(7))[:-2]
```

- [0, 1, 2, 3, 4] ***
- [1, 2, 3, 4, 5]
- [0, 1, 2, 3, 4, 5]
- [6, 5, 4, 3, 2, 1, 0]
- [0, 1, 2, 3, 4, 5, 6]

7. Учитывая следующие две функции, что возвратит функция funA(1, 2) ?

```
def funA(a, b):  
    col = [a, b]  
    for x in range(4):  
        col.append( funB(col) )  
    return col
```

```
def funB(y):  
    return y[-1] + y[-2]
```

- 3
- [1, 2, 0, 1, 2, 3]
- Это вызовет ошибку во время выполнения
- [1, 1, 2, 3, 5, 8]
- [1, 2, 3, 5, 8, 13] ***

8. Какие из этих выражений имеет значение False всякий раз, когда a имеет значение True?

- a or not b
- a and not b
- a == a == b
- not a or a == b
- not not False == a ***

9. Каким будет значение переменной z после выполнения следующего кода на Python?

```
x = [1, 2, 3]
```

```

y = [7, 8, 9]
x += y
x[-1] += 10
x += [20]
z = x[-1] + x[-2]

```

- Z = 39 ***
- Z = 17
- Z = 27
- Z = 29
- Z = 30

10. Если $n = 5$ (т.е. начальная длина $Lx=5$), каким будет окончательная длина Lx ?

```

n = len(Lx)

for y in range(n):
    Lx += Lx[y:-y]

```

- 20
- 25
- 28 ***
- 30
- 35

11. Каким будет значение $valX$ после 100 итераций?

```

valX = 1
for a in range(100):
    if a % 2 == 0:
        valX *= valX/2
    else:
        valX += valX

```

- 100
- 1 ***
- 200
- $100!/2$
- 0.5

12. Какая функция будет возвращать точность (количество десятичных знаков) для введенного пользователем значение типа `float`? (Например, точность 5.147 это 3)

- (A)

```
def prec():  
    z = input('Введите значение типа float: ')  
    return len(z) - z.find('.') + 1
```
- (B)

```
def prec(): ***  
    z = input('Введите значение типа float: ')  
    return len(z) - z.find('.') - 1
```
- (C)

```
def prec():  
    z = eval(input('Введите значение типа float: '))  
    return len(z) - z.find('.') + 1
```
- (D)

```
def prec():  
    z = eval(input('Введите значение типа float: '))  
    return len(z) - z.find('.') - 1
```
- (E) ни одна из вышеперечисленных

13. Каковы значения следующих трех выражений?

`a = 29.8 // 3 % 5`

`b = 49 % 25 // 5`

`c = 29 // 5 / 2`

- `a = 4.9, b = 4.8, c = 2.9`
- `a = 2.8, b = 0.2, c = 2`
- `a = 4.5, b = 1, c = 2.5`
- `a = 5.0, b = 2, c = 4.0`
- `a = 4.0, b = 4, c = 2.5 ***`

14. Что является результатом вычисления этого выражения?

`[1, 2, 3, 4, 5][2:-1]`

- [3, 4] ***
- Ошибка
- [3, 4, 5]
- [2, 3, 4]
- [5, 4, 3]

15. Какое утверждение лучше всего описывает эффект от funC (xy)?

```
def funC(xy):
    z = len(xy) // 2
    xy[:z] = xy[z:]
```

- Она дублирует нечетные индексированные элементы xy
- Она дублирует четные индексированные элементы xy
- Она копирует правую половину XY в левую половину. ***
- Она не изменяет xy.
- Она копирует левую половину XY в правую половину.

16. Какое утверждение наилучшим образом описывает эффект функции myFunc на список ab числовых значений

```
def mystery(ab):
    for z in range(0, len(ab)-1, 2):
        if ab[z] > ab[z+1]:
            ab[z], ab[z+1] = ab[z+1], ab[z]
    return ab
```

- Она сортирует все элементы ab.
- Она меняет все пары элементов ab.
- Она меняет пары элементов ab, за исключением последнего элемента.
- Она зеркалирует ab.
- Она сортирует пары элементов ab. ***
- Она меняет элементы ab так, что больший элемент всегда отображается первым.

17. Какое утверждение не относится к объектам в объектно-ориентированной программе?

- Объекты могут содержать данные.
- Объекты могут иметь операции.

- Объекты могут содержать другие объекты.
- Объекты могут вызывать методы других объектов.
- Объекты не могут видоизменяться. ***

18. Вы создаете новый класс с именем Sphere, производный от класса Shape. Класс Shape имеет следующий код:

```
class Shape
{
    public virtual void Area()
    {
        // дополнительный код...
    }
}
```

Метод Area в классе Sphere должен предоставлять новые функциональные возможности, но также скрыть реализацию метода Area класса Shape. Какой сегмент кода следует использовать для этого?

a) class Sphere : Shape

```
{
    public override void Area()
    {
        // дополнительный код ...
    }
}
```

b) class Sphere : Shape

```
{
    public new void Area()
    {
        // дополнительный код ...
    }
}
```

c) class Sphere : Shape

```
{
    public virtual void Area()
    {
        // дополнительный код ...
    }
}
```

d) class Sphere : Shape

```
{
```

```

public static void Area()
{
    // дополнительный код ...
}

```

19. Вы разрабатываете приложение на C#. Вы должны решить, следует ли объявить член класса как статический. Какое из следующих утверждений верно для статических членов классов?

- a) Вы можете использовать ключевое слово `this` в качестве ссылки на статический метод или свойство.
- b) Только одна копия статического поля совместно используется всеми экземплярами класса.
- c) Статические члены класса можно использовать только после создания экземпляра класса.
- d) Ключевое слово `static` используется для объявления членов, которые не относятся к отдельным объектам, а только к самому классу.

20. Создан класс с именем `GeoShape`. Определен метод `Area` в этом классе, он вычисляет площадь геометрической фигуры. Вы хотите, чтобы производные классы `GeoShape` могли заменить его функциональность для поддержки расчетов дополнительных геометрических фигур. Когда метод `Area` вызывается в объекте `GeoShape`, площадь должна быть рассчитана, основываясь на используемом в текущий момент объекте `GeoShape`. Ключевое слово, которое вы должны использовать с определением площади методом в `GeoShape` класс?

- a) `abstract`
- b) `virtual`
- c) `new`
- d) `overrides`

21. Вы разрабатываете программы на C# и написали следующий код:

```

int x = 16;
int y = --x;
int z = y++;

```

Какое будет значение переменной `z`, после того, как все вышеописанные операторы выполнятся?

- a) 14
- b) 15

- c) 16
- d) 17

22. Вы пишете метод с именем `PrintReport`, который не возвращает значение в вызывающий код. Какое ключевое слово следует использовать в вашем объявлении метода?

- a) `void`
- b) `private`
- c) `int`
- d) `string`

23. Вы разрабатываете программу на C# и написали следующую строку кода:

```
int x = 6 + 4 * 4 / 2 - 1;
```

Каково будет значение переменной `x` после того, как этот оператор выполнится?

- a) 19
- b) 13
- c) 20
- d) 14

24. Вы разрабатываете программы на C# и написали следующий код:

```
int i = 6;
do
{
    if (i == 3)
        break;
    Console.WriteLine("Значение i = {0}", i);
    i++;
}
while (i <= 5);
```

Сколько раз будет выполнен цикл `while`?

- a) 0
- b) 1
- c) 2
- d) 3

25. Вы пишете программы на C#, которые используются для итерации по коллекциям (массивам и спискам). Вы должны убедиться, что вы обрабатываете каждый элемент коллекции один раз. Вы также должны убедиться, что ваш код легко

читать и отлаживать. Какое из следующих выражений на C# обеспечить лучшее решение для этого требования?

- a) while
- b) for
- c) foreach
- d) do-while

26. Вы пишете код для бизнес-приложений с использованием C#. Вы пишете следующий оператор объявления массива:

```
int[] numbers = { 1, 2, 3, 4, 5 };
```

Теперь вам нужно получить доступ ко второму элементу в этом массиве. Какие из следующих выражений следует использовать?

- a) numbers[0]
- b) numbers[1]
- c) numbers[2]
- d) numbers[3]

27. Вы разрабатываете программы на C# и написали рекурсивный метод для вычисления факториала числа. Какой из следующих фрагментов кода следует использовать, чтобы генерировать правильные результаты?

```
a) public static int Factorial(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else
    {
        return n * Factorial(n - 1);
    }
}
```

```
b) public static int Factorial(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else
    {
        return (n - 1) * Factorial(n);
    }
}
```

```
    }  
}
```

```
c) public static int Factorial(int n)  
{  
    if (n == 0)  
    {  
        return n;  
    }  
    else  
    {  
        return Factorial(n - 1);  
    }  
}
```

```
d) public static int Factorial(int n)  
{  
    return n * Factorial(n - 1);  
}
```

28. Вы пишете программы на C# и написали следующий метод

```
public static void TestSwitch(int op1, int op2, char  
opr)  
{  
    int result;  
    switch (opr)  
    {  
        case '+':  
            result = op1 + op2;  
        case '-':  
            result = op1 - op2;  
        case '*':  
            result = op1 * op2;  
        case '/':  
            result = op1 / op2;  
        default:  
            Console.WriteLine("Неизвестный оператор");  
            return;  
    }  
    Console.WriteLine("Результат: {0}", result);  
    return;  
}
```

Однако, когда вы компилируете этот код, вы получаете следующее сообщение об ошибке:

Управление не проваливается из одной метки инструкции case к другой

Как нужно модифицировать код, чтобы убедиться, что он успешно компилируется?

- a) после каждого case добавьте следующую строку кода: `break`;
- b) после каждого case добавьте следующую строку кода: `continue`;
- c) после каждого case добавьте следующую строку кода: `goto default`;
- d) после каждого case добавьте следующую строку кода: `return`;

29. Какие из следующих утверждений являются верными?

- a) Процедурная парадигма программирования отличается от парадигмы структурного программирования.
- b) Парадигма объектно-ориентированного программирования направлена на разделение логики на более мелкие части и написание процедур для каждой части.
- c) Классы и объекты - краеугольные камни парадигмы структурного программирования.
- d) В парадигме объектно-ориентированного программирования одинаково важны данные и процедуры, которые работают с этими данными.
- e) C# на платформе .NET — это язык структурного программирования.

30. Какие из следующих утверждений верны для фрагмента кода на # C ?

```
class Sample
{
    private int i;
    public Single j;
    private void DisplayData()
    {
        Console.WriteLine(i + " " + j);
    }
    public void ShowData()
    {
        Console.WriteLine(i + " " + j);
    }
}
```

- a) j не может быть объявлен как public
- b) DisplayData() не может быть объявлен как private.
- c) DisplayData() не может получить доступ к j.
- d) ShowData() не может получить доступ к i
- e) Ошибки нет в этом классе.

31. Какие из следующих описаний типов относятся к типам значений?

- 1. integer

2. array
3. single
4. string
5. long

- a) 1, 2, 5
- b) 1, 3, 5
- c) 2, 4
- d) 3, 5

- a) System.Int16
- b) System.Int32
- c) System.Int64
- d) System.Int128

**Пример лабораторной работы по дисциплине
«Языки и методы программирования»:
Лабораторная работа № 2 по Python**

Цель выполнения работы. Практическое знакомство с модулем math, функцией range(), использование стандартных функций.

Задание 1. Найти значение выражений.

1. $\log(\cos(7))$
2. $\cos(4) + \log(4)$
3. $25./85$

Для подключения соответствующих функций использовать в начале программы команду

```
from math import log, cos
```

Задание 2. Написать программу, выводящую на экран все целые числа, принадлежащие отрезку [3,92] (включая концы) и делящиеся на 3.

Задание 3. Имеется произвольный список L, состоящий из целых чисел. Вывести на экран все элементы списка L (в том порядке, в котором они следуют в L), делящиеся на 3. Например, для списка L=[27, 10, 28, 2, 19, 4, 15, 45, 17] ответом будет

27

15

45

Задание 4. Найти минимальное из трёх чисел, записанных в переменные x, y, z.

Использовать встроенные функции.

В качестве задания по самостоятельной работе решить данную задачу также на языке C#.

Задание 5. Написать программу выводящую медиану трёх чисел, записанных в переменные *a*, *b*, *c*. Медиана — это число, которое является серединой множества чисел.

Задание 6. Имеется список *L*, содержащий как минимум четыре элемента (но возможно больше). Поменять в нём второй и предпоследний элемент. Например, для *L*=[2, 4, 8, 1, 3] в результате выполнения программы должно выполняться: *L*==[2, 1, 8, 4, 3].

В качестве задания по самостоятельной работе решить данную задачу также на языке C#.

Задание 7. Заданы два числа, записанные в переменные *a* и *b*. В результате выполнения программы в переменной *a* должен быть записан максимум из чисел *a* и *b*, а в переменной *b* их минимум.

В качестве задания по самостоятельной работе решить данную задачу также на языке C#.

Задание 8. В переменной *a* находится некоторое число. В результате выполнения программы в переменную *a* должно быть записано вдвое большее число.

Задание 9. В переменной *b* записан первый член некоторой геометрической прогрессии, а в переменной *q* её знаменатель. Найти сумму первых 18 членов этой прогрессии, не используя общую формулу для суммы членов геометрической прогрессии.

Напоминание: геометрическая прогрессия — это последовательность чисел, в которой каждый следующий член в *q* раз больше предыдущего, где *q* — знаменатель прогрессии.

В качестве задания по самостоятельной работе решить данную задачу также на языке C#.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Задание 1.

Программа

```
from math import log, cos
print (log(cos(7)))
print (cos(4)+log(4))
print (25./85)
```

Результат

-0.28249255552713626

0.7326507402562786

0.29411764705882354

Задание 2.**Программа**

```
for i in range(3,92):
```

```
    if i%3==0:
```

```
        print (i)
```

Результат

3

6

9

12

15

18

21

24

27

30

33

36

39

42

45

48

51

54

57

60

63

66

69

72

75

78

81

84

87

90

Задание 3.**Программа**

```
L=[27, 10, 28, 2, 19, 4, 15, 45, 17]
```

```
for x in L:
```

```
    if x%3==0:
```

```
        print (x)
```

Результат

27

15

45

Задание 4.

Программа

x=248

y=447

z=192

print (min (x,y,z))

Результат

192

Задание 5.

Программа

a=34

b=22

c=49

while (b-a)*(b-c)>0:

 a,b,c=b,c,a

print (b)

Результат

34

34

Задание 6.

Программа

L=[2, 4, 8, 1, 3]

L[1],L[-2]=L[-2],L[1]

print (L)

Результат

[2, 1, 8, 4, 3]

Задание 7.

Программа

a=24

b=38

if a<b:

 a,b=b,a

print (a,b)

Результат

38 24

Задание 8.

Программа

a=78

a=a*2

print (a)

Результат

156

Задание 9.

Программа

```
b=89
q=5
x=b
s=0
for i in range(18):
    s=s+x
    x=x*q
print (s)
```

Результат

84877014160134

Задание 10.

Программа

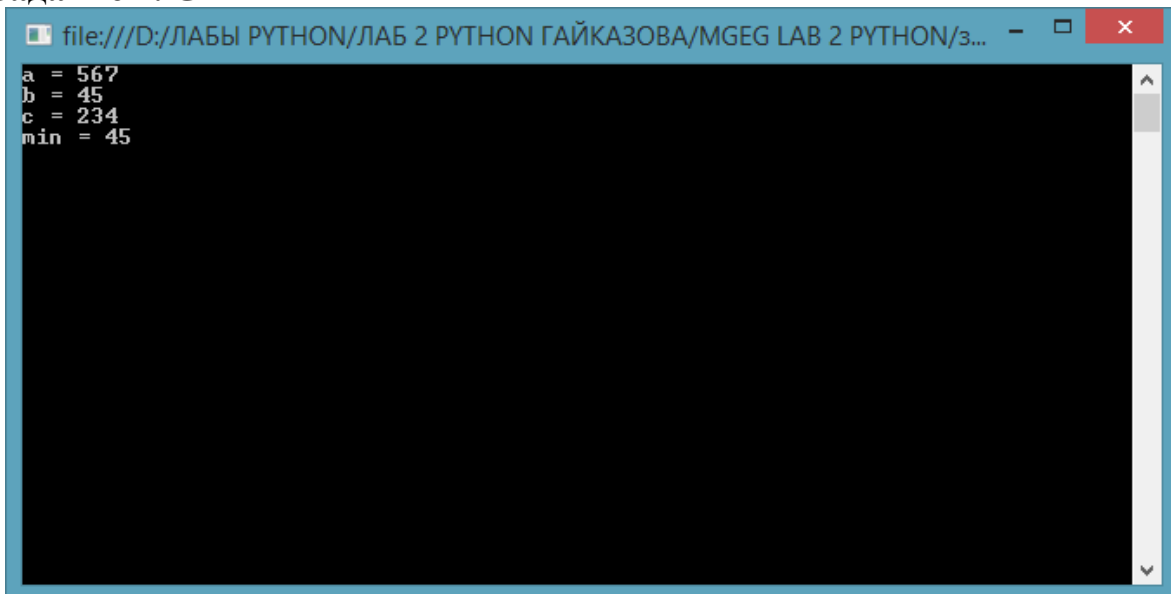
```
a=18
b=1
bnew=2
while abs(b-bnew)>0.0001:
    b=bnew
    bnew=0.5*(b+float(a)/b)
print (bnew)
from math import sqrt
print (sqrt(a))
```

Результат

4.242640687119335

4.24264068711928

Задание 4. C#



```
file:///D:/ЛАБЫ PYTHON/ЛАБ 2 PYTHON ГАЙКАЗОВА/MGEG LAB 2 PYTHON/з...
a = 567
b = 45
c = 234
min = 45
```

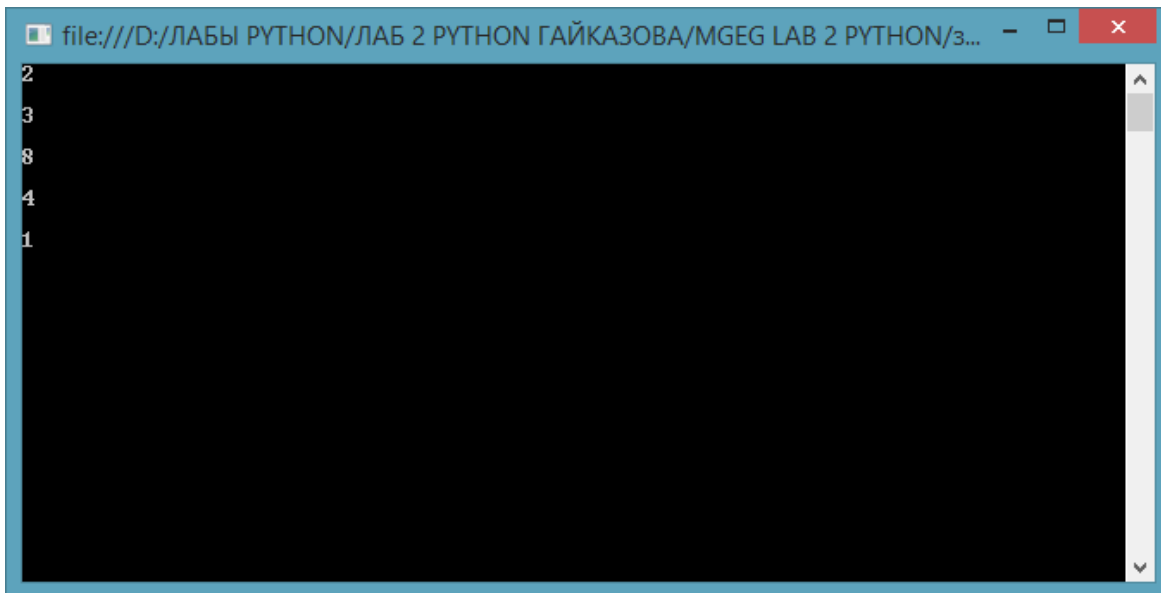
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
```

```

namespace задание_4
{
    class Program
    {
        static double min(double a, double b, double c)
        {
            double min = 0;
            if (a <= b && a <= c) { min = a; }
            else
            {
                if (b <= a && b <= c) { min = b; }
                else
                {
                    if (c <= b && c <= a) { min = c; }
                }
            }
            return min;
        }
        static void Main(string[] args)
        {
            double a, b, c;
            Console.Write("a = ");
            a = Convert.ToDouble(Console.ReadLine());
            Console.Write("b = ");
            b = Convert.ToDouble(Console.ReadLine());
            Console.Write("c = ");
            c = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("min    =    "    +    min(a,    b,
c).ToString());
            Console.ReadLine();
        }
    }
}

```

Задание 6. C#



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace задание_6
{
    class Program
    {
        static void Main(string[] args)

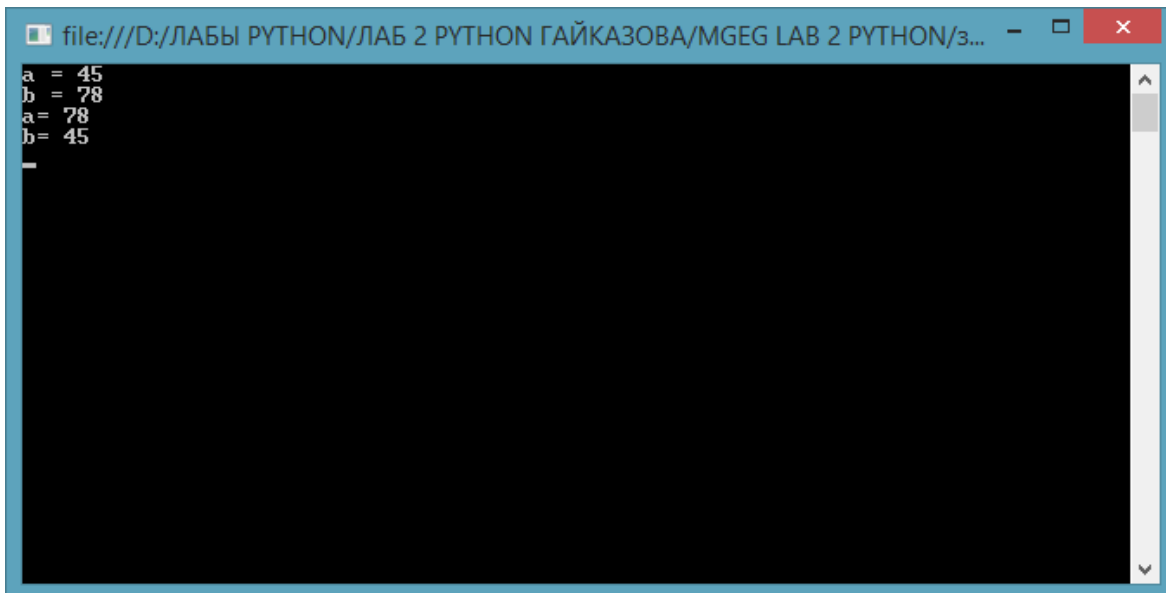
        {

            List<int> numbers = new List<int>() { 2, 4, 8, 3, 1

};

            int temp = numbers[1];
            numbers[1] = numbers[3];
            numbers[3] = temp;
            foreach (int i in numbers)
            {
                Console.WriteLine(i);
                Console.ReadLine();
            }
        }
    }
}
```

Задание 7. C#



```
a = 45
b = 78
a= 78
b= 45
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace задание_7
{
    class Program
    {
        public static void MinMax(ref double a, ref double b)
        {
            if (a == b)
                throw new ArgumentException("Error!");
            if (a > b)
            {
                double temp = a;
                a = b;
                b = temp;
            }
        }
        public static void Main(string[] args)
        {
            double a, b;
            Console.Write("a = ");
            a = Convert.ToDouble(Console.ReadLine());
            Console.Write("b = ");
            b = Convert.ToDouble(Console.ReadLine());
            MinMax(ref a, ref b);
            Console.WriteLine("a= {0}", b);
            Console.WriteLine("b= {0}", a);
            Console.ReadLine();
        }
    }
}
```



```

    }
}

```

Задание 9. C#

```

file:///D:/ЛАБЫ PYTHON/ЛАБ 2 PYTHON ГАЙКАЗОВА/MGEG LAB 2 PYTHON/з...
Введите первый член прогрессии
20
Введите шаг
5
Введите число членов прогрессии
18
Сумма первых 18 членов геометрической прогрессии 19073486328120

```

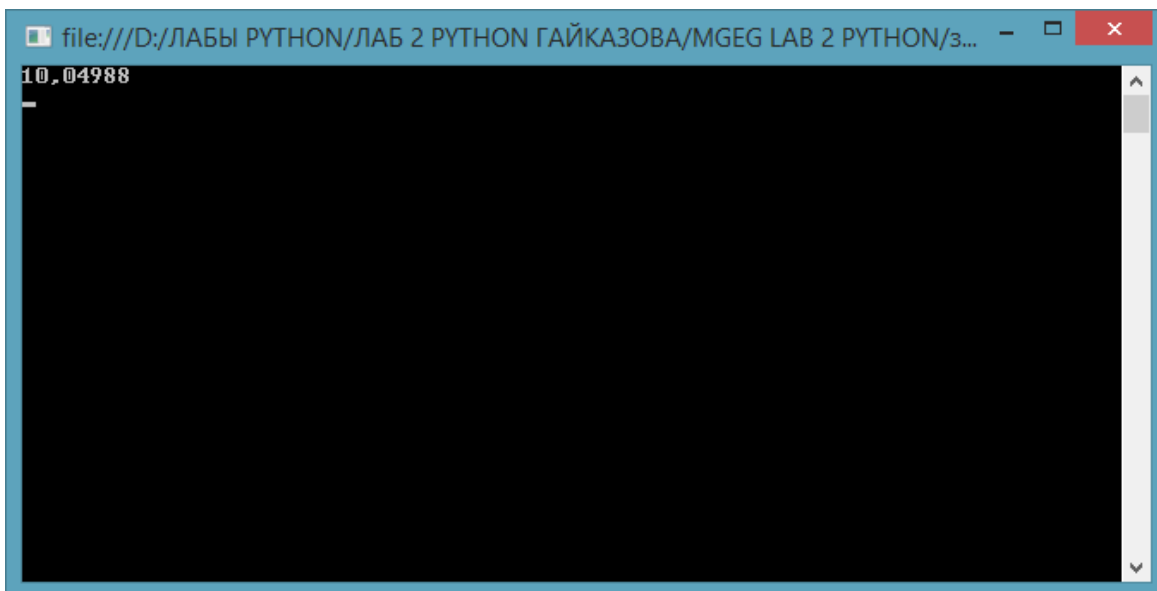
```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace задание_9
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Введите первый член прогрессии");
            double b1 = double.Parse(Console.ReadLine());
            Console.WriteLine("Введите шаг");
            double q = double.Parse(Console.ReadLine());
            Console.WriteLine("Введите число членов прогрессии");
            int n = int.Parse(Console.ReadLine());
            double Sum = 0;
            for (int i = 1; i <= n; i++) Sum = (b1 * (1 -
Math.Pow(q, n))) / (1 - q);
            Console.WriteLine("Сумма        первых        18        членов
геометрической прогрессии " + Sum);
            Console.ReadLine();
        }
    }
}

```

Задание 10. C#



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace задание10
{
    class Program
    {
        static void Main(string[] args)
        {
            float a = 0.0F;
            a = (float)Math.Sqrt(101);
            Console.WriteLine(a);
            Console.ReadLine();
        }
    }
}
```

Пример домашнего задания по дисциплине «Языки и методы программирования»

Задача 1: Элемент арифметической прогрессии

Найдите 100-й член арифметической прогрессии, если её нулевой член равен 20, а шаг — 207.

Задача 2: Сумма кубов

Напишите программу вычисляющую сумму кубов натуральных чисел от 1 до 100:

Задача 3: Факториал

Напишите программу вычисляющую факториал числа. Использовать цикл для вычисления.

Задача 4: Трехзначные числа

Квадрат трехзначного числа оканчивается тремя цифрами, равными этому числу. Найдите и выведите все такие числа в порядке возрастания.

В поле для записи ответа нужно ввести все эти числа через пробел.

Задача 6. Сложные проценты

Клиент решил положить деньги в банк, ставка по вкладу, в котором составляет $p\%$ годовых. Он решил положить m рублей на k лет. Какую сумму он получит после окончания срока вклада?

Примерные вопросы к экзамену (проводится в устной форме) в 7 семестре

1. HTML5. Общая характеристика.
2. Подход HTML 4 к веб-разметки.
3. Подход HTML5 к веб-разметке. Общая характеристика. Семантическая разметка.
4. Элементы header и footer. Назначение и использование. Пример.
5. Элемент <section>. Назначение и использование. Пример.
6. Элемент <article>. Назначение и использование. Пример.
7. Элемент <aside>. Назначение и использование. Пример.
8. Универсальные селекторы. Селекторы атрибутов элементов. Селекторы нижележащих элементов.
9. Псевдо-классы в CSS.
10. Псевдо-элементы в CSS.
11. Боксы. Типы боксов.
12. Поток и расположение элементов
13. Выравнивание блока
14. Поток и расположение элементов
15. Позиционирование. Типы позиционирований. z-index
16. Границы и отступы. Выступ за границы родительского элемента. Плавающие блоки
17. Очистка float. Методы очистки float.
18. Flexbox. Основные преимущества flexbox
19. Типы данных JavaScript.
20. Объявление и использование переменных.
21. Арифметические операторы и операторы сравнения
22. Условные операторы: if, if-else. Оператор switch-case
23. Функции в JavaScript.
24. Анонимные функции
25. Способы определения функции в JS
26. Область действия переменных и функций
27. Четыре способа вызова функций в JavaScript

28. Объекты в JavaScript. Создание объекта. Способы.
29. Прототип. Цепь прототипов
30. Классическое наследование и безклассовое наследование (Prototypal)
31. Массивы в JavaScript. Общая характеристика.
32. Методы для работы с массивами
33. Введение в библиотеку jQuery. Общая характеристика.
34. Ненавязчивый JavaScript.
35. AngularJS. Общая характеристика
36. Модель MVC.
37. Bootstrap. Основные инструменты Bootstrap.
38. Адаптивный дизайн. Назначение и использование
39. Элементы header, nav, section, aside, footer, article, hgroup, figure и figcaption
40. Элементы mark, small, cite, address, time,
41. Традиционная блочная модель.
42. Выравнивание тела документа по центру
43. Навигационная полоса
44. Раздел и боковая врезка
45. Создание двух столбцов с помощью свойства float
46. Нижний колонтитул.
47. Свойства border-radius, box-shadow, text-shadow, @font-face
48. Линейный и радиальный градиенты. RGBA(). HSLA()
49. Свойства outline, border-image, transform и transition
50. Функции transform: scale, transform: rotate, transform: skew, transform: translate
51. Одновременное использование всех видов трансформации
52. Формы HTML и ввод данных. Общая характеристика. Базовые элементы.
53. Фиксированный, резиновый дизайн и адаптивный дизайны.
54. Макеты на основе обтекаемых элементов..
55. Логические операции.
56. Циклы while, do while, for, for ... in. Операторы break и continue
57. Основы работы с DOM. DOM.
58. Работа с событиями в DOM.
59. Знакомство и работа с элементом canvas. Общая характеристика.
60. Знакомство с SVG. Общая характеристика.

5.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Оценивание степени освоения обучающимися дисциплины осуществляется на основе «Положение о балльно-рейтинговой системе оценки успеваемости студентов МГОУ».

Шкала соответствия рейтинговых оценок пятибалльным оценкам:

Оценка по 5-балльной системе		Оценка по 100-балльной системе
5	отлично	81 – 100
4	хорошо	61 - 80
3	удовлетворительно	41 - 60
2	неудовлетворительно	21 - 40
1	необходимо повторное изучение	0 - 20

В зачетно-экзаменационную ведомость и зачетную книжку выставляются оценки по пятибалльной шкале и рейтинговые оценки в баллах.

При получении студентом на зачёте неудовлетворительной оценки в ведомость выставляется рейтинговая оценка в баллах (<40 баллов), соответствующая фактическим знаниям (ответу) студента.

Общее количество баллов по дисциплине – 100 баллов.

Максимальное количество баллов, которое можно набрать в течение семестра за посещаемость, выполнение лабораторных и самостоятельных работ, тестирование – 40 баллов.

За посещение лекционных занятий и написание конспектов обучающийся может набрать максимально 10 баллов.

За выполнение заданий по самостоятельной работе обучающийся может набрать максимально 10 баллов (4 задания по 2,5 балла).

За выполнение лабораторных работ обучающийся может набрать максимально 10 баллов (25 работ по 0,4 балла).

За тестирование обучающийся может набрать максимально 10 баллов (10 тестовых вопросов по 1 баллу за каждый).

Обучающийся, набравший 15 балл и более, допускается к экзамену. Максимальная сумма баллов, которые обучающийся может набрать при сдаче экзамена, составляет 60 баллов.

Для сдачи экзамена по дисциплине необходимо выполнить все требуемые лабораторные работы (получить допуск к экзамену у преподавателя, проводившего лабораторные работы). Существенным моментом является посещаемость занятий (в случае пропусков занятий предполагается более подробный опрос по темам пропущенных занятий). На экзамен выносятся материал, излагаемый в лекционном курсе и рассматриваемый на лабораторных занятиях. Для сдачи экзамена надо правильно ответить на два вопроса билета и решить предложенную задачу, а также ответить на несколько поставленных дополнительных вопросов. В затруднительных ситуациях (в отдельных случаях) допускается на экзамене воспользоваться записью материалов лекций и семинаров в присутствии преподавателя. При этом преподаватель может убедиться, в какой степени студент ориентируется в «своих» материалах, и по ряду дополнительных вопросов (по тетради) решить вопрос о уровне выставаемых баллов для экзамена.

При пересдаче экзамена используется следующее правило для формирования рейтинговой оценки:

- 1-я пересдача – фактическая рейтинговая оценка, полученная студентом за ответ, минус 10 баллов;
- 2-я пересдача – фактическая рейтинговая оценка, полученная студентом за ответ, минус 10 баллов.

Учет посещаемости лекционных и лабораторных занятий осуществляется по ведомости, представленной ниже в форме таблицы.

Московский государственный областной университет
Ведомость учета посещения
Физико-математический факультет

Направление подготовки: 44.03.05 - Педагогическое образование

Профиль подготовки: Математика и информатика

Дисциплина: Языки и методы программирования

Группа: 41

Преподаватель: Грамаков Д.А.

№ п/п	Фамилия И.О.	Посещение занятий								Итого
		1	2	3	4				16	
1.	Иванов И.И.	+	-	+	-				+	10
2.	Петров П.П.	-	+	+	+				+	5

Московский государственный областной университет
Ведомость учета текущей успеваемости
Физико-математический факультет

Направление подготовки: 44.03.05 - Педагогическое образование

Профиль подготовки: Математика и информатика

Дисциплина: Языки и методы программирования

Группа: 41

Преподаватель: Грамаков Д.А.

№ п/п	Ф. И.О.	Сумма баллов, набранных в семестре					Общая сумма баллов (макс. 100)	Итоговая оценка		Подпись препода- вателя
		Посещ. до 10 баллов	Лаб. работы до 10 баллов	Самостоятел ьная работа до 10 баллов	Тести- рование до 10 баллов	Экзаменд о 60 баллов		Цифра	Про- пись	
1	2	3	4	5	7	8	9	10	11	12
1.	Иванов И.И.	6	8	6		19		4	хор.	
2.	Петров	7	7	6		10		4	удовл.	

	П.П.									
3.										

Структура оценивания заданий по самостоятельной работе

Критерии оценивания	Баллы
Аккуратность и полнота выполнения всех пунктов задания	0-1
Понимание логики выполнения задания и значения полученных результатов	0-1

Структура оценивания лабораторных работ

Критерии оценивания	Баллы
Аккуратность и полнота выполнения всех пунктов задания	0-1
Понимание логики выполнения задания и значения полученных результатов	0-1

Структура оценивания тестирования

Критерии оценивания	Баллы
Задание выполнено верно	1
Задание выполнено наполовину или вообще неверно	0

Структура оценивания на экзамене

Уровни оценивания	Критерии оценивания	Баллы
<i>оценка «отлично»</i>	Ставится, если студент обнаруживает всестороннее, систематическое и глубокое знание базовых понятий языков программирования, изучаемых по дисциплине. Выполнил все предложенные лабораторные и самостоятельные работы, вовремя сдал отчеты по ним. При ответах по теоретическим вопросам на экзамене показал умение четко излагать изученный материал, демонстрируя взаимосвязь основных понятий дисциплины и умение оперировать ими; проявляя творческие способности в понимании, изложении и использовании учебного материала. Правильно решил задачу, предложенную к теоретическим вопросам, показав полное	41-60

Уровни оценивания	Критерии оценивания	Баллы
	понимание каждой конструкции написанной веб-программы и понимание реализованного алгоритма решения. Показал знакомство с дополнительной литературой, рекомендованной программой. В качестве дополнительного критерия выставления оценки студент должен иметь не менее 35-40 баллов за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу.	
<i>оценка «хорошо»</i>	Ставится, если студент, обнаруживает полное знание базовых понятий языков программирования, изучаемых по дисциплине. Выполнил не менее 80 % предложенных лабораторных и самостоятельных работ, имеет незначительные нарушения по срокам сдачи отчетов. При ответах по теоретическим вопросам на экзамене по одному из вопросов не показал умения четко излагать изученный материал, демонстрируя взаимосвязь основных понятий дисциплины и умение оперировать ими. Правильно решил задачу, предложенную к теоретическим вопросам, но не показал полного понимания каждой конструкции написанной программы и не объяснил суть реализованного алгоритма решения. Не ознакомился с дополнительной литературой, рекомендованной программой. В качестве дополнительного критерия выставления оценки студент должен иметь не менее 25-30 баллов за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу.	31-50
<i>оценка «удовлетворительно»</i>	Ставится, если студент обнаруживает знание базовых понятий языков программирования, изучаемых по дисциплине. Выполнил не менее 60% предложенных лабораторных и самостоятельных работ, имеет	21-40

Уровни оценивания	Критерии оценивания	Баллы
	незначительные нарушения по срокам сдачи отчетов. При ответе на вопросы билета не предоставил правильного ответа на один из вопросов, но при этом по другому вопросу показал умения излагать изученный материал, демонстрируя его понимание. Правильно решил задачу, предложенную к теоретическим вопросам, но не показал полного понимания каждой конструкции написанной веб-программы и не объяснил суть реализованного алгоритма решения. В случае, когда предложенная задача не решена, но на теоретические вопросы студент ответил. Не ознакомился с дополнительной литературой, рекомендованной программой. В качестве дополнительного критерия выставления оценки студент должен иметь не менее 15-20 баллов за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу.	
<i>оценка</i> <i>«неудовлетворительно»</i>	Ставится в том случае, если студент на экзамене воспользовался запрещенными средствами списывая (смартфон, компьютер, бумажные носители) или не выполнил два из предложенных трех заданий билета. В качестве дополнительного критерия выставления оценки студент имеет за посещение лекционных, лабораторных занятий, а также тестирование и самостоятельную работу менее 15 баллов.	1-20

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная литература

1. Бугримов, А.Л. Python. Быстрое погружение в программирование: учеб.пособие / А. Л. Бугримов, В. В. Лаврентьев. - М. : МГОУ, 2018. - 48с. – Текст: непосредственный.

2. Тюкачев, Н.А. С#. Программирование 2D и 3D векторной графики : учеб.пособие / Н. А. Тюкачев, В. Г. Хлебостроев. - 3-е изд.,стереотип. - СПб. : Лань, 2018. - 320с + CD. – Текст: непосредственный.
3. Подбельский В.В., Язык С#. Базовый курс [Электронный ресурс] : учеб. пособие / В.В. Подбельский. - 2-е изд., перераб. и доп. - М. : Финансы и статистика, 2015. - 408 с. - ISBN 978-5-279-03534-2 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785279035342.html> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
4. Тюкачев, Н.А. С#. Основы программирования : учебное пособие / Н.А. Тюкачев, В.Г. Хлебостроев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2018. — 272 с. — ISBN 978-5-8114-2567-9. — URL: <https://e.lanbook.com/book/104962> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Лань» . — Текст : электронный
5. Златопольский Д.М., Основы программирования на языке Python [Электронный ресурс]: учебник / Златопольский Д. М. - М. : ДМК Пресс, 2017. - 284 с. - ISBN 978-5-97060-552-3 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785970605523.html> дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
6. Шелудько В.М., Основы программирования на языке высокого уровня Python [Электронный ресурс]: учебное пособие / Шелудько В. М. - Ростов н/Д : Изд-во ЮФУ, 2017. - 146 с. - ISBN 978-5-9275-2649-9 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785927526499.html> дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
7. Федоров, Д. Ю. Программирование на языке высокого уровня python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 161 с. — (Бакалавр. Прикладной курс). — ISBN 978-5-534-10971-9. — URL: <https://biblio-online.ru/bcode/437489> (дата обращения: 26.07.2019).— Режим доступа: для авториз. пользователей Электронно-библиотечная система «Юрайт» . — Текст : электронный

6.2. Дополнительная литература

Рекомендуемые интернет-источники и литература:

1. Диков, А.В. Веб-технологии HTML и CSS : учебное пособие / А.В. Диков. - 2-е изд. - Москва : Директ-Медиа, 2012. - 78 с. : ил.,табл., схем. - URL: <http://biblioclub.ru/index.php?page=book&id=96968> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Университетская библиотека онлайн» . — Текст : электронный
2. Залогова, Л.А. Основы объектно-ориентированного программирования на базе языка С# : учебное пособие / Л.А. Залогова. — Санкт-Петербург : Лань, 2018. — 192 с. — ISBN 978-5-8114-3093-2. — URL: <https://e.lanbook.com/book/106731>

- (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Лань» . — Текст : электронный
3. Конова, Е.А. Алгоритмы и программы. Язык С++ : учебное пособие / Е.А. Конова, Г.А. Поллак. — 4-е изд., стер. — Санкт-Петербург : Лань, 2019. — 384 с. — ISBN 978-5-8114-4039-9. — URL: <https://e.lanbook.com/book/114696> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Лань». — Текст : электронный
 4. Солдатенко, И.С. Практическое введение в язык программирования Си : учебное пособие / И.С. Солдатенко, И.В. Попов. — Санкт-Петербург : Лань, 2018. — 132 с. — ISBN 978-5-8114-3150-2. — URL: <https://e.lanbook.com/book/109619> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Лань» . — Текст : электронный.
 5. Хоган, Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Б. Хоган; [пер. с англ. Е. Матвеева]. - 2-е изд. - СПб.: Питер, 2014. - 320 с.: ил. - (Библиотека программиста).
 6. Буйначев С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. — Екатеринбург : Изд-во Урал. ун-та, 2014. — 91 с.
 7. МакГрат Майк. Программирование для начинающих: [перевод с англ. М.А. Райтмана] /Майк МакГрат. — Москва : Эксмо, 2015. — 192 с. — (Программирование для начинающих).
 8. Чан Уэсли. Python: создание приложений. Библиотека профессионала, 3-е изд. Пер. с англ. - М. : ООО "И.Д. Вильямс", 2015. - 816 с. : ил. - Парал. тит. англ .
 9. Васильев А. Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.
 10. Чан Уэсли. Python: создание приложений. Библиотека профессионала, 3-е изд. Пер. с англ. - М. : ООО "И.Д. Вильямс", 2015. - 816 с. : ил. - Парал. тит. англ .
 11. Мэттиз Эрик. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»).
 12. Васильев А. Н. Python на примерах. Практический курс по программированию. - СПб.: Наука и Техника, 2016. - 432 с.: ил.
 13. Бриггс Джейсон. Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017. — 320 с.
 14. Буйначев С. К. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. — Екатеринбург : Изд-во Урал. ун-та, 2014. — 91, [1] с.
 15. МакГрат Майк. Программирование для начинающих: [перевод с англ. М.А. Райтмана] /Майк МакГрат. — Москва : Эксмо, 2015. — 192 с. — (Программирование для начинающих).
 16. Саммерфилд М., Python на практике / Марк Саммерфилд - М. : ДМК Пресс, 2014. - 338 с. - ISBN 978-5-97060-095-5 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785970600955.html> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный Маккинли У., Python и анализ данных / Уэс Маккинли - М. : ДМК

- Пресс, 2015. - 482 с. - ISBN 978-5-97060-315-4 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785970603154.html> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
17. Рамальо, Л. Python. К вершинам мастерства / Лучано Рамальо ; пер. с англ. А.А. Слинкина. - Москва : ДМК Пресс, 2016. - 768 с. - ISBN 978-5-97060-384-0. - Режим доступа: <http://znanium.com/catalog/product/1028052> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «znanium.com» . — Текст : электронный
18. Мишенин А.И., Сборник задач по программированию : учеб. пособие / А.И. Мишенин. - М. : Финансы и статистика, 2014. - 224 с. - ISBN 978-5-279-03300-3 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785279033003.html>. (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
19. Ошероув Р., Искусство автономного тестирования с примерами на C#. / Ошероув Р. - 2-е издание. - М. : ДМК Пресс, 2014. - 360 с. - ISBN 978-5-94074-945-5 - URL: <http://www.studentlibrary.ru/book/ISBN9785940749455.html> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
20. Робисон У., C# без лишних слов / Робисон У. ; Пер. с англ. - М. : ДМК Пресс, 2008. - 352 с. (Серия "Для программистов") - ISBN 5-94074-177-0 - URL: <http://www.studentlibrary.ru/book/ISBN5940741770.html> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
21. Тейт Брюс, Семь языков за семь недель. Практическое руководство по изучению языков программирования / Тейт Брюс - М. : ДМК Пресс, 2014. - 384 с. - ISBN 978-5-94074-539-6 - URL: <http://www.studentlibrary.ru/book/ISBN9785940745396.html> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный
22. Кингсли-Хью Э., JavaScript в примерах / Кингсли-Хью Э., Кингсли-Хью К. ; Пер. с англ. - М. : ДМК Пресс, 2009. - 272 с. - ("Для программистов"). - ISBN 978-5-94074-668-3 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785940746683.html> (дата обращения: 26.07.2019). — Режим доступа: для авториз. пользователей Электронно-библиотечная система «Консультант студента» . — Текст : электронный

6.3. Ресурсы информационно-телекоммуникационной сети «Интернет»

1. Документация по Python [Электронный ресурс] – URL: <https://docs.python.org /3/>
2. Электронная библиотечная система ZNANIUM.COM <http://znanium.com/>

3. Электронная библиотечная система «Университетская библиотека онлайн» <http://biblioclub.ru/index.php>
4. Интерактивные онлайн-курсы по HTML и CSS – <https://htmlacademy.ru/>
5. Справочник по HTML – <http://htmlbook.ru/>
6. Онлайн-учебник по Javascript <http://javascript.ru/>
7. Интернет-портал образовательных ресурсов по ИТ - <http://www.intuit.ru>
8. Интернет-портал с ресурсами по программным продуктам компании Microsoft <http://www.msdn.ru>
9. Интернет-портал со статьями по алгоритмике и программированию - <http://algolist.manual.ru/>
10. Электронная библиотека по техническим наукам - <http://techlibrary.ru>

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины «Программирование на языке Ассемблера» обучающиеся могут найти в следующих пособиях:

1. Грань Т.Н., Холина С.А. Методические рекомендации по проведению лекционных занятий.
2. Грань Т.Н., Холина С.А. Методические рекомендации об организации выполнения и защиты курсовой работы.
3. Грань Т.Н., Холина С.А. Методические рекомендации по проведению лабораторных и практических занятий.

Использование в процессе обучения компетентностного подхода предусматривает применение в образовательном процессе активных и интерактивных форм проведения занятий (компьютерных симуляций, разбор конкретных ситуаций, круглых столов) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся. Учебный процесс строится на концептуальной основе, предполагающей выделение единой основы, сквозных и межпредметных идей курса. Важным аспектом при обучении информационным технологиям в курсе «Языки и методы программирования» является проблема разработки и внедрения подходов и приемов обучения, которые обеспечивали бы возможность непрерывного обновления знаний в области информационных технологий у студентов. Реализация этого подхода требует использование новых средств обучения - электронных учебников и пособий, справочников, Интернет-ресурсов, а также определение наиболее эффективных условий и форм организации деятельности обучаемого. Основная задача видится в грамотном использовании дидактических возможностей применения информационных технологий в ходе учебного процесса. При использовании ПК и проекционного оборудования в ходе лекции делает возможным наглядно демонстрировать функциональные особенности изучаемых языков программирования. Специально для таких лекций разрабатываются комплексы слайд-презентаций, что позволяет существенно сократить время,

необходимое на изложение нового учебного материала и демонстрировать важные понятия языков программирования на многочисленных примерах, приведенных на слайдах.

Важной особенностью изучения дисциплины «Языки и методы программирования» состоит в выполнении комплекса лабораторных работ, главной задачей которых является получение навыков программирования и использования современных технологий для решения различных профессиональных задач в области образования. При этом выполняемые лабораторные работы, кроме непосредственных заданий содержат дополнительный теоретический материал, способствующий дальнейшему освоению изучаемой дисциплины. Особое место в овладении частью тем данной дисциплины отводится самостоятельной работе, при этом во время аудиторных занятий могут быть рассмотрены и проработаны наиболее важные и трудные вопросы по той или иной теме дисциплины, а второстепенные и более легкие вопросы, а также вопросы, специфичные для направления подготовки, могут быть изучены студентами самостоятельно.

Использование дидактических возможностей применения информационных технологий в ходе учебного процесса значительно совершенствует его организацию, реализовывает индивидуальный подход к каждому студенту, значительно экономит время при обучении, помогает в формировании исследовательских навыков и умений принимать оптимальные решения. Такой подход позволяет в должной мере обеспечить уровень подготовки будущих специалистов к реализации всех компонентов их профессиональной деятельности.

8. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ

Лицензионное программное обеспечение:

Microsoft Windows

Microsoft Office

Kaspersky Endpoint Security

Информационные справочные системы:

Система ГАРАНТ

Система «КонсультантПлюс»

Профессиональные базы данных

fgosvo.ru

pravo.gov.ru

www.edu.ru

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение дисциплины включает в себя:

- учебные аудитории для проведения занятий лекционного и семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, укомплектованные учебной мебелью, доской, демонстрационным оборудованием.
- помещения для самостоятельной работы, укомплектованные учебной мебелью, персональными компьютерами с подключением к сети Интернет и обеспечением доступа к электронным библиотекам и в электронную информационно-образовательную среду МГОУ;
- помещения для хранения и профилактического обслуживания учебного оборудования, укомплектованные мебелью (шкафы/стеллажи), наборами демонстрационного оборудования и учебно-наглядными пособиями;
- лаборатория, оснащенная лабораторным оборудованием: комплект учебной мебели, проектор, проекционная доска, персональные компьютеры с подключением к сети Интернет и обеспечением доступа к электронным библиотекам и в электронную информационно-образовательную среду МГОУ.