

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ

"БРАТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"

УТВЕРЖДАЮ

Проректор по учебной работе

_____ Е.И.Луковникова

_____ 14 мая _____ 20 24 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.О.12 Программирование

Закреплена за кафедрой **Информатики, математики и физики**

Учебный план b090302_24_ИСиТ.plx

Направление: 09.03.02 Информационные системы и
технологии

Квалификация **Бакалавр**

Форма обучения **очная**

Общая трудоемкость **12 ЗЕТ**

Виды контроля в семестрах:

Экзамен 2,3, Курсовой проект 3

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	2 (1.2)		3 (2.1)		Итого	
Неделя	18		17			
Вид занятий	УП	РП	УП	РП	УП	РП
Лекции	36	36	34	34	70	70
Лабораторные	72	72	51	51	123	123
В том числе инт.	8	8	8	8	16	16
Итого ауд.	108	108	85	85	193	193
Контактная работа	108	108	85	85	193	193
Сам. работа	72	72	77	77	149	149
Часы на контроль	36	36	54	54	90	90
Итого	216	216	216	216	432	432

Программу составил(и):

ст.пр., Ефремова А.Н. _____

Рабочая программа дисциплины

Программирование

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.02 Информационные системы и технологии (приказ Минобрнауки России от 19.09.2017 г. № 926)

составлена на основании учебного плана:

Направление: 09.03.02 Информационные системы и технологии
утвержденного приказом ректора от 30.01.2024 № 32.

Рабочая программа одобрена на заседании кафедры

Информатики, математики и физики

Протокол от 18.04.2024 г. № 10

Срок действия программы: 2024-2028 уч.г.

Зав. кафедрой Горохов Д. Б.

Председатель МКФ

старший преподаватель Латушкина С.В. 26.04.2024 г. № 8

Ответственный за реализацию ОПОП _____ Горохов Д. Б.

Директор библиотеки _____ Сотник Т.Ф.

№ регистрации _____ 24
(учебный отдел)

Визирование РПД для исполнения в очередном учебном году

Председатель МКФ

_____ 2025 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2025-2026 учебном году на заседании кафедры
Информатики, математики и физики

Внесены изменения/дополнения (Приложение _____)

Протокол от _____ 2025 г. № ____

Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель МКФ

_____ 2026 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2026-2027 учебном году на заседании кафедры
Информатики, математики и физики

Внесены изменения/дополнения (Приложение _____)

Протокол от _____ 2026 г. № ____

Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель МКФ

_____ 2027 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2027-2028 учебном году на заседании кафедры
Информатики, математики и физики

Внесены изменения/дополнения (Приложение _____)

Протокол от _____ 2027 г. № ____

Зав. кафедрой _____

Визирование РПД для исполнения в очередном учебном году

Председатель МКФ

_____ 2028 г.

Рабочая программа пересмотрена, обсуждена и одобрена для
исполнения в 2028-2029 учебном году на заседании кафедры
Информатики, математики и физики

Внесены изменения/дополнения (Приложение _____)

Протокол от _____ 2028 г. № ____

Зав. кафедрой _____

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	Получение базовых знаний в области программирования, непосредственное изучение высокоуровневого языка программирования Python, сред разработки, правил оформления кода, элементов ООП, необходимых выпускнику, освоившему программу бакалавриата, для решения различных задач практической деятельности. Формирование профессиональных умений и навыков, позволяющей решать стандартные задачи профессиональной деятельности на основе информационных технологий.
-----	---

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:		Б1.О.12
2.1	Требования к предварительной подготовке обучающегося:	
2.1.1	Алгоритмы и структуры данных	
2.1.2	Информатика	
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:	
2.2.1	Web-программирование	
2.2.2	Выполнение и защита выпускной квалификационной работы	

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

ОПК-2: Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;	
Индикатор 1	ОПК-2.3. Имеет навыки создания информационных технологий и программных средств для решения задач профессиональной деятельности
ОПК-3: Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;	
Индикатор 1	ОПК-3.2. Умеет решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.
ОПК-6: Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;	
Индикатор 1	ОПК-6.1 - Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий;
Индикатор 2	ОПК-6.2 - Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий;

В результате освоения дисциплины обучающийся должен

3.1	Знать:
3.1.1	Современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности; Принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; Методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий; Методологию разработки программного обеспечения и технологии программирования;
3.2	Уметь:
3.2.1	Выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности; Решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; Составлять алгоритмы, использовать базовые алгоритмические структуры для решения типовых задач; Применять выбранные языки программирования для написания программного кода;
3.3	Владеть:
3.3.1	Навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности; Навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций, и библиографии по научно-исследовательской работе с учетом требований информационной безопасности; Навыками разработки алгоритмов и программ для решения практических задач в области информационных систем и технологий; Навыками выбора, применения методов и алгоритмов и технологии программирования для решения стандартных профессиональных задач.

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)								
Код занятия	Вид занятия	Наименование разделов и тем	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
	Раздел	Раздел 1. Введение в программирование						
1.1	Лек	Этапы решения задач с помощью ЭВМ: постановка задачи, создание модели, алгоритм, кодирование алгоритма, анализ результатов	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
1.2	Лек	Языки программирование. История развития языков программирования.	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
1.3	Лек	Введение в Python. История создания, области применения и перспективы. Общие сведения о языке программирования Python. Типы и структуры данных. программы.	2	4	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	2	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2 Лекция-визуализация
1.4	Лаб	1. Знакомство с интегрированной средой разработки IDLE Python, Anaconda. Работа с библиотекой Math	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	2	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2 Работа в малых группах
1.5	Ср	Введение в программирование	2	20	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
1.6	Экзамен	Подготовка к экзамену	2	18	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
	Раздел	Раздел 2. Основные алгоритмические структуры и их программная реализация на Python						
2.1	Лек	Линейная алгоритмическая структура. Основные управляющие конструкции линейного алгоритма	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.2	Лаб	2. Программная реализация алгоритмов линейной структуры	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.3	Лек	Разветвленная алгоритмическая структура. Основные управляющие конструкции алгоритма с ветвлением.	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.4	Лаб	3. Программная реализация алгоритмов разветвленной структуры	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.5	Лек	Циклическая алгоритмическая структура. Основные управляющие конструкции циклического алгоритма	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2

2.6	Лаб	4. Программная реализация алгоритмов циклической структуры	2	4	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.7	Лек	Последовательности в Python. Способы создания и считывания списков. Описание методов для работы со списками. Массив. Типовые задачи обработки одномерных массивов	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.8	Лаб	5.Программная реализация обработки одномерных массивов	2	10	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.9	Лек	Двумерные массивы. Особенности программной реализации на Python	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.10	Лаб	6.Программная реализация обработки двумерных массивов	2	10	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.11	Лек	Строки, Кортежи. Основные конструкции, используемые для обработки строковой информации	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.12	Лаб	7. Преобразование символьных величин (списки, кортежи)	2	6	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.13	Лек	Множества	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.14	Лаб	8. Множества	2	8	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.15	Лек	Словари, функции словаря, операции	2	2	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.16	Лаб	9. Словари	2	8	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.17	Лек	Функции. Регулярные выражения	2	4	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.18	Лаб	10. Использование функций	2	8	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.19	Лаб	11. Регулярные выражения	2	4	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.20	Лек	Открытие и закрытие файлов, текстовые файлы, файлы CSV, бинарные файлы. Модуль shelve, Модуль OS и работа с файловой структурой	2	8	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-6.3

2.21	Лаб	12. Файлы	2	8	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
2.22	Ср	Введение в программирование. Основные алгоритмические структуры и их программная реализация на Python. Структуры данных	2	52	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	4	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2 Работа в малых группах
2.23	Экзамен	Подготовка к экзамену	2	18	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
	Раздел	Раздел 3. Основные понятия и определения объектно-ориентированного и событийно-ориентированного программирования						
3.1	Лек	История объектно-ориентированного подхода к программированию. Базовые принципы ООП: абстракция, инкапсуляция, наследование, полиморфизм.	3	4	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	2	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2 Лекция-визуализация
3.2	Лек	Классы и объекты. Атрибуты класса. Поля(свойства) класса. Методы (функции класса). Конструкторы. Уровни доступа атрибутов.	3	6	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
3.3	Лек	Инкапсуляция. Полиморфизм. Перегрузка методов. Переопределение методов.	3	4	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
3.4	Лек	Наследование. Множественное наследование	3	6	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
3.5	Лек	Абстрактные методы и классы	3	4	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
3.6	Лек	Событийно-ориентированное программирование. Библиотека Tkinter. Основные этапы создания оконного интерфейса. Виджеты, обработка событий, виды событий, создание графического изображения и анимации.	3	10	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
3.7	Лаб	13. Лабораторный практикум ООП	3	40	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	2	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2 Работа в малых группах
3.8	Лаб	14.Лабораторный практикум СОП	3	11	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2

3.9	КП	Создание приложения с графическим пользовательским интерфейсом на Python	3	26	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
3.10	Ср	Основные понятия и определения объектно-ориентированного и событийно-ориентированного программирования	3	77	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	4	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2
3.11	Экзамен	Подготовка к экзамену	3	28	ОПК-6 ОПК-2 ОПК-3	Л1.1Л2.1 Л2.2 Л2.3 Л2.4Л3.1 Л3.2	0	ОПК-6.1 ОПК-6.2 ОПК-2.3 ОПК-3.2

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Традиционная (репродуктивная) технология (преподаватель знакомит обучающихся с порядком выполнения задания, наблюдает за выполнением и при необходимости корректирует работу обучающихся)

Технология коллективного взаимодействия (работа в малых группах) (самостоятельное изучение обучающимися нового материала посредством сотрудничества в малых группах, дает возможность всем участникам участвовать в работе, практиковать навыки сотрудничества, межличностного общения)

Образовательные технологии с использованием активных методов обучения (лекция-визуализация)

Технология компьютерного обучения(использование в учебном процессе компьютерных технологий и предоставляемых ими возможностей (онлайн тесты))

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

6.1. Контрольные вопросы и задания

Лекция-визуализация №1 - Введение в Python. История создания, области применения и перспективы. Общие сведения о языке программирования Python. Типы и структуры данных. Средства программирования на языке Python. Структура программы. (2 час.)

Самостоятельная работа - Введение в программирование. Основные алгоритмические структуры и их программная реализация на Python. Структуры данных (самостоятельное изучение обучающимися нового материала посредством сотрудничества в малых группах, дает возможность всем участникам участвовать в работе, практиковать навыки сотрудничества, межличностного общения)(4 час.)

Лабораторные работы

Лабораторная работа №1 Знакомство с интегрированной средой разработки IDLE Python, Anaconda. Работа с библиотекой Math (самостоятельное изучение обучающимися нового материала посредством сотрудничества в малых группах, дает возможность всем участникам участвовать в работе, практиковать навыки сотрудничества, межличностного общения)(2 час.)

Цель:изучить интерфейс интегрированной среды разработки IDLE Python; возможности используемых функции и констант библиотеки Math.

Контрольные вопросы:

- Опишите правила построения имён переменных в языке Python.
- Как записываются комментарии на Python? Подумайте, как комментирование можно использовать при поиске ошибок в алгоритме?
- Расскажите о работе оператора вывода Python.
- Что такое переменная? Как транслятор определяет тип переменной?
- Зачем нужен тип переменной?
- Как изменить значение переменной?
- Какие типы данных вы знаете?
- Какие данные записываются в логические переменные?
- Расскажите об особенностях переменных в языке Python. Почему может получиться, что изменение одной переменной автоматически приводит к изменению другой?
- Что такое приоритет операций? Зачем он нужен?
- В каком порядке выполняются операции, если они имеют одинаковый приоритет?
- Зачем используются скобки?
- Что происходит, если в выражения входят переменные разных числовых типов? Какого типа будет результат?
- Опишите операции // и %.
- Расскажите о проблеме вычисления остатка от деления в различных языках программирования. Обсудите в классе этот вопрос.
- Какие стандартные математические функции вы знаете? В каких единицах задается аргумент тригонометрических

функций?

17. Как выполнить округление вещественного числа к ближайшему целому?
18. Какие числа называют случайными? Зачем они нужны?
19. Как получить «естественное» случайное число? Почему такие числа почти не используются в цифровой технике?
20. Чем отличаются псевдослучайные числа от случайных?
21. Какие функции для получения псевдослучайных чисел вы знаете?

Лабораторная работа №2 Программная реализация алгоритмов линейной структуры

Цель: ознакомиться с понятием линейной алгоритмической структуры; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Линейная алгоритмическая структура?
2. Элементы языка для программной реализации линейных алгоритмов?
3. Форматированный вывод данных?
4. Форматный вывод данных?

Лабораторная работа №3 Программная реализация алгоритмов разветвленной структуры

Цель: ознакомиться с понятием разветвленной алгоритмической структуры; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Чем отличаются разветвляющиеся алгоритмы от линейных?
2. Как вы думаете, почему не все задачи можно решить с помощью линейных алгоритмов?
3. Почему нельзя выполнить обмен значений двух переменных в два шага: $a=b$; $b=a$?
4. Чем отличаются условные операторы в полной и неполной формах? Как вы думаете, можно ли обойтись только неполной формой?
5. Какие отношения вы знаете? Как обозначаются отношения «равно» и «не равно»?
6. Как организовать выбор из нескольких вариантов?
7. Что такое сложное условие?
8. Как определяется порядок вычислений в сложном условии?

Лабораторная работа №4 Программная реализация алгоритмов циклической структуры

Цель: ознакомиться с понятием циклической алгоритмической структуры; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое цикл?
2. Сравните цикл с переменной и цикл с условием. Какие преимущества и недостатки есть у каждого из них?
3. Что означает выражение «цикл с предусловием»?
4. В каком случае цикл с предусловием не выполняется ни разу?
5. В каком случае программа, содержащая цикл с условием, может заиклиться?
6. В каком случае цикл с переменной не выполняется ни разу?
7. Верно ли, что любой цикл с переменной можно заменить циклом с условием? Верно ли обратное утверждение?
8. В каком случае можно заменить цикл с условием на цикл с переменной?

Лабораторная работа №5 Программная реализация обработки одномерных массивов

Цель: ознакомиться с понятием массива; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое массив? Зачем нужны массивы?
2. Как вы думаете, почему в языке Python нет массивов, а вместо них используются списки?
3. Какие способы создания списков вы знаете?
4. Что такое генераторы списков?
5. Зачем нужны генераторы списков с условием?
6. Как построить список, состоящий из 15 единиц, с помощью генератора списка?
7. Как обращаться к отдельному элементу списка?
8. Как ввести список с клавиатуры?
9. Как вывести список на экран? Приведите разные варианты решения этой задачи. Какой из них вам больше нравится?
10. Как заполнить список случайными числами в диапазоне от 100 до 200?
11. С помощью каких функций можно найти сумму и количество элементов списка?

Лабораторная работа №6 Программная реализация обработки двумерных массивов

Цель: ознакомиться с понятием двумерного массива; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое матрицы? Зачем они нужны?
2. Сравните понятия «массив» и «матрица».
3. Как вы думаете, можно ли считать, что первый индекс элемента матрицы – это номер столбца, а второй – номер строки?
4. Что такое главная и побочная диагонали матрицы?
5. Почему суммирование элементов главной диагонали требует одиночного цикла, а суммирование элементов под главной

диагональю – вложенного?

6. Что такое сортировка?

7. На какой идее основан метод пузырька? метод выбора?

8. Объясните, зачем нужен вложенный цикл в описанных методах сортировки.

9. Сравните метод пузырька и метод выбора. Какой из них требует меньше перестановок?

10. Расскажите про основные идеи метода «быстрой сортировки».

11. Как вы думаете, можно ли использовать метод «быстрой сортировки» для нечисловых данных, например, для символьных строк?

12. От чего зависит скорость «быстрой сортировки»? Какой самый лучший и самый худший случай?

13. Как вы думаете, может ли метод «быстрой сортировки» работать дольше, чем метод выбора (или другой «простой» метод)? Если да, то при каких условиях?

14. Как нужно изменить приведенные алгоритмы, чтобы элементы массива были отсортированы по убыванию?

15. Какие встроенные средства сортировки массивов в Python вы знаете?

16. Как задать нестандартный порядок сортировки для функции sorted?

17. Что такое «лямбда-функции»? Когда их удобно использовать?

18. Чем отличаются функция sorted и метод sort для списков?

Лабораторная работа №7 Преобразование символьных величин (списки, кортежи)

Цель: ознакомиться с основными конструкциями, используемыми для обработки строковой информации; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое символьная строка?

2. Как задать значение для символьной строки? Рассмотрите разные способы.

3. Как обращаться к элементу строки с заданным номером?

4. Почему нельзя сразу записать новое значение в заданную позицию строки? Как можно решить эту задачу?

5. Как вычисляется длина строки?

6. Что обозначает оператор '+' применительно к строкам?

7. Перечислите основные операции со строками и приведите примеры их использования.

8. Как определить, что при поиске в строке образец не найден?

9. Как преобразовать число из символьного вида в числовой и обратно?

10. Почему строку не всегда можно преобразовать в число?

11. Объясните выражение «рекурсивный перебор».

12. Сравните рекурсивные и нерекурсивные методы решения переборных задач.

Лабораторная работа №8 Множества

Цель: познакомиться с основными конструкциями, используемыми для работы с множеством; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое множество?

2. Отличие множества от списков?

3. Приемы программной реализации алгоритмов?

Лабораторная работа №9 Словари

Цель: познакомиться с основными конструкциями, используемыми для работы со словарями; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое словарь? Какие операции он допускает?

2. Можно ли обратиться к элементам словаря по индексам? Как вы думаете, почему сделано именно так?

3. Как создать пустой словарь?

4. Как создать словарь из готовых пар «ключ-значение»? Найдите в литературе или в Интернете разные способы решения этой задачи.

5. Как добавить элемент в словарь?

6. Как обращаться к элементу словаря?

7. Расскажите о методе get. Чем его можно заменить?

8. Как получить список всех ключей словаря? всех значений словаря?

9. Как перебрать все пары «ключ-значение»?

10. Зачем при чтении строк из файла используется метод strip?

Лабораторная работа №10 Использование функций

Цель: получить первоначальные навыки создания и использования функций; приемы программной реализации на языке программирования Python. Произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое процедуры? В чем смысл их использования?

2. Как оформляются процедуры в Python? Достаточно ли включить процедуру в текст программы, чтобы она «сработала»?

3. Что такое параметры? Зачем они используются?

4. Какие переменные называются локальными? глобальными?

5. Как в процедуре прочитать и изменить значение глобальной переменной?

6. Как оформляются процедуры, имеющие несколько параметров?

7. Что такое функция? Чем она отличается от процедуры?
8. Как по тексту программы определить, какое значение возвращает функция?
9. Какие функции называются логическими? Зачем они нужны?
10. Напишите функцию, которая вычисляет максимальное из трёх чисел, не используя встроенную функцию `max`.
11. Напишите функцию, которая вычисляет количество цифр числа.
12. Напишите функцию, которая вычисляет наибольший общий делитель двух чисел.

Лабораторная работа №11 Регулярные выражения

Цель: познакомиться с правилами создания регулярных выражений и получить первоначальные навыки их использования; приемы программной реализации на языке программирования Python. Произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Как использовать регулярные выражения?
2. Примеры написания регулярных выражений?

Лабораторная работа №12 Файлы

Цель: познакомиться с основными конструкциями, используемыми для работы с файлами; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Чем отличаются текстовые и двоичные файлы по внутреннему содержанию? Можно ли сказать, что текстовый файл – это частный случай двоичного файла?
2. Объясните «принцип сэндвича» при работе с файлами.
3. Как вы думаете, почему открытый программой файл, как правило, блокируется и другие программы не могут получить к нему доступ?
4. Почему рекомендуется вручную закрывать файлы, хотя при закрытии программы они закрываются автоматически? В каких ситуациях это может быть важно?
5. Что такое файловая переменная? Почему для работы с файлом используют не имя файла, а файловую переменную?
6. В каком случае одна и та же файловая переменная может быть использована для работы с несколькими файлами, а в каком – нет?
7. Что такое «последовательный доступ к данным»?
8. Как можно начать чтение данных из файла с самого начала?
9. Как определить, что данные в файле закончились?
10. В каких случаях нужно открывать одновременно несколько файлов?

Лабораторная работа №13 (Лабораторный практикум ООП)

Цель: познакомиться с основными парадигмами ООП, конструкциями, используемыми для работы с объектами, классами; приемами программной реализации на языке программирования Python; произвести отладку и тестирование полученных программ.

Контрольные вопросы:

1. Что такое ООП? Опишите базовые концепции, которые лежат в основе данной методологии программирования.
2. Для чего было создано ООП? На каких принципах базируется ООП? Опишите их основные идеи?
3. Приведите преимущества и недостатки объектно-ориентированного подхода.
4. Что такое объект и чем характеризуются объекты в ООП?
5. Что такое класс и зачем он нужен? Приведите общее определение класса в ООП.
6. Какая разница между классом и объектом в ООП? Какие разновидности классов существуют в ООП?
7. Как описывается класс в Python?
8. Каким способом можно создать объект (экземпляр класса) в Python?
9. Как можно обратиться к полям и методам объекта? В чём состоит особенность первого аргумента в методах класса?
10. Какой стандартный метод при описании класса необходимо переопределить, чтобы гарантировать, что в тех ситуациях, где требуется строковое представление объекта данного класса, система автоматических вызывала данный метод у ссылочной переменной?
11. Что такое UML? Как с помощью UML можно описать классы и их взаимосвязь друг с другом (т.е. описать UML-диаграмму классов)?
12. Что такое инстанцирование?
13. Кто и как в языке Python создаёт экземпляр (объект) класса?
14. Как в терминах ООП называется метод, который вызывается сразу после создания объекта для инициализации его состояния?
15. Как его описать в классе в языке Python?
16. Когда именно вызывается конструктор?
17. Можно ли перегружать конструкторы? Если да, то зачем это нужно?
18. Кто занимается удалением объектов из памяти, если они уже не нужны системе?
19. Как в терминах ООП называется метод, который вызывается перед удалением объекта из памяти?
20. Что такое абстракция (абстрагирование) в ООП?
21. Чем отличаются атрибуты класса от атрибутов экземпляров класса?
22. Каким образом можно получить доступ к атрибутам класса? А как можно получить доступ к состоянию и поведению экземпляра класса?
23. Что такое инкапсуляция и зачем она нужна?
24. Каков главный козырь инкапсуляции?

25. Где Вы в реальной жизни встречаете инкапсуляцию и на что она влияет?
26. Является ли инкапсуляция только прерогативой ООП или где то она уже Вам встречалась при получении опыта программирования?
27. Какими средствами обычно обеспечивается инкапсуляция в ООП?
28. Как в языке Python реализуется инкапсуляция на уровне синтаксиса языка и на уровне методологии программирования?
29. Как получить доступ к инкапсулированным данным в языке Python?
30. Что такое свойства в языке Python и как их реализовать?
31. Как реализовать свойства «только для чтения» или свойство «только для записи»?
32. Зачем нужны динамические методы и как их реализовать в языке Python?
33. Зачем нужны методы класса и как их реализовать в языке Python?
34. Зачем нужны статические методы и как их реализовать в языке Python?
35. Каким образом можно вызвать в языке Python динамические, статические методы и методы класса?
36. В чём концептуальная разница между разными типами методов и функций, которые можно описать внутри класса? Приведите примеры.
37. Преимущества и недостатки инкапсуляции?
38. Что такое повторное использование кода?
39. Какие подходы и средства предлагает методология ООП для повторного использования кода?
40. На что указывает принцип DRY (Don't Repeat Yourself)?
41. Что такое зависимость? Как зависимость реализуется в языке Python?
42. Что такое ассоциация? Как реализована ассоциация в языке Python?
43. Что такое наследование? Как реализовано наследование в языке Python?
44. Какие преимущества и недостатки есть у множественного наследования классов?
45. Разрешено ли в языке Python множественное наследование? Если разрешено множественное наследование в языке Python, то как наследуются дочерним классом одинаковые характеристики из разных базовых классов?
46. Как при наследовании вызываются конструкторы?
47. Кто находится во главе всей иерархии в языке Python?
48. Какой именно минимальный функционал получают все экземпляры классов в языке Python? С помощью какой встроенной функции можно посмотреть этот функционал?
49. Что показывает функция `isinstance()` и как ей пользоваться?
50. Что показывает функция `issubclass()` и как ей пользоваться?
51. Что такое агрегация? Как реализована агрегация в языке Python?
52. Что такое композиция? Как реализована композиция в языке Python?
53. Что такое делегирование? Как реализовано делегирование в языке Python?
54. Основное отличие наследования от агрегации/композиции?
55. Преимущества и недостатки композиции и агрегации? Преимущества и недостатки наследования?
56. Как с помощью языка UML отобразить все зависимости между классами на UML-диаграмме классов?
57. Что такое полиморфизм?
58. Какие разновидности полиморфизма бывают в языках программирования?
59. Как реализуется статический полиморфизм в языке Python?
60. Что такое перегрузка методов (функций или операторов) в языке Python?
61. Как динамическая типизация влияет на реализацию динамического полиморфизма в языке Python?
62. Чем полиморфизм языка Python отличается от многих других языков программирования?
63. Что такое виртуальный метод? Что такое переопределение методов в языке Python?
64. Что такое абстрактный класс? А что такое абстрактный метод?
65. Можно ли реализовать абстрактный класс (метод) в языке Python? Если да, то как и какой модуль при этом нужно подключать?
66. Что должен всегда делать абстрактный метод в языке Python при его объявлении в базовом классе?

Лабораторная работа № 14 (Лабораторный практикум СОП)

Цель: познакомиться с основными этапами создания оконного интерфейса (библиотека Tkinter)

Контрольные вопросы:

1. Создание и отображение окна.
2. Подключение стандартного модуля `tkinter`.
3. Указание координат и размеров.
4. Управление прозрачностью окон.
5. Изменение фона.
6. Особенности создания прозрачных окон в `tkinter`.
7. Размещение компонент на форме с помощью метода `pack`
8. Назначение обработчиков событий.
9. Блокировка и удаление обработчика событий.
10. Передача данных в обработчик.
11. Использование свойства `command` для подключения обработчиков.
12. Подключение обработчика с помощью универсального метода `bind`.
13. Применение виджета `Frame`.
14. Размещения компонент с помощью метода `pack`.
15. Размещение нескольких компонентов в окне.
16. Объединение компонент в группу.
17. Применение полос прокрутки.

18. Особенности методов pack, grid и place.
19. Создание диалоговых окон.

6.2. Темы письменных работ

Курсовой проект

Цель: закрепление знаний, получение теоретических и практических навыков программной реализации приложений с использованием компонентов ООП и СОП в Python

Создание приложения с графическим пользовательским интерфейсом на Python

6.3. Фонд оценочных средств

Экзаменационные вопросы

- 1.1. Этапы решения задач с помощью ЭВМ: постановка задачи, создание модели, алгоритм, кодирование алгоритм, анализ результатов.
- 1.2. Языки программирования. История и тенденции развития языков программирования.
- 1.3. Трансляторы: интерпретаторы и компиляторы.
- 1.4. Области применения языка программирования Python
- 1.5. Переменные в Python. Наименование. Модель памяти Python при работе с переменными
- 1.6. Создание программ на языке Python в отдельном файле. Отличие от интерактивного режима
- 1.7. Операторы отношений в Python. Логические операции над объектами
- 2.1. Условная инструкция if
- 2.2. Строковые методы в Python. Отличие функций от методов
- 2.3. Списки в Python. Создание списка
- 2.4. Операции над списками в Python
- 2.5. Методы списка в Python
- 2.6. Преобразование типов в Python (списки, строки)
- 2.7. Вложенные списки в Python
- 2.8. Циклы в Python
- 2.9. Цикл for для списков и строк в Python
- 2.10. Функции в Python. Создание функций
- 2.11. Функция range() и цикл for в Python
- 2.12. Способы генерации списка в Python
- 2.13. Цикл while в Python
- 2.14. Вложенные циклы в Python (на примере вложенных списков)
- 2.15. Множества и операции над ними в Python
- 2.16. Кортежи и операции над ними в Python
- 2.17. Словари и операции над ними в Python
- 2.18. Работа с файлами в Python
- 2.19. Поддержка в Python функционального программирования
- 3.1. Что такое класс в Python? Каковы его основные характеристики?
- 3.2. Опишите базовые принципы ООП.
- 3.3. Что такое экземпляр класса? Каким образом можно осуществить его создание?
- 3.4. Дайте определение атрибута класса и опишите его основные особенности.
- 3.5. Что такое методы класса? Каковы особенности создания и вызова метода?
- 3.6. Каковы отличия закрытых методов от обычных?
- 3.7. В чем заключается преимущество использования конструктора __init__() при создании класса?
- 3.8. Как осуществляется перегрузка специальных методов класса в Python?
- 3.9. Как реализуется принцип наследования в Python? Приведите примеры.
- 3.10. В чем смысл использования абстрактного метода в Python?
- 3.11. Объекты. Динамическая типизация.
- 3.12. Инкапсуляция.
- 3.13. Генерация объекта class.
- 3.14. Объект экземпляр класса.
- 3.15. Атрибуты класса. Атрибуты данных.
- 3.16. Атрибуты-методы. Параметр self. Добавление атрибутов к классу во время исполнения программы.
- 3.17. Специальные методы и атрибуты классов. Методы __init__() и __del__() в Python.
- 3.18. Декораторы функций и декораторы классов.
- 3.19. Метаклассы.
- 3.20. Абстрактные методы в Python.
- 3.21. Наследование. Базовый и производный класс. Построение производного класса.
- 3.22. Множественное наследование.
- 3.23. Агрегация. Контейнеры. Иерархия наследования.
- 3.24. Полиморфизм. Доступ к методам базового класса.
- 3.25. Создание и отображение окна.
- 3.26. Подключение стандартного модуля tkinter.
- 3.27. Указание координат и размеров.
- 3.28. Управление прозрачностью окон.
- 3.29. Изменение фона.

- 3.30. Особенности создания прозрачных окон в tkinter.
- 3.31. Размещение компонент на форме с помощью метода pack
- 3.32. Назначение обработчиков событий.
- 3.33. Блокировка и удаление обработчика событий.
- 3.34. Передача данных в обработчик.
- 3.35. Использование свойства command для подключения обработчиков.
- 3.37. Применение виджета Frame.
- 3.38. Размещения компонент с помощью метода pack.
- 3.39. Размещение нескольких компонентов в окне.
- 3.40. Объединение компонент в группу.
- 3.41. Применение полос прокрутки.
- 3.42. Особенности методов pack, grid и place.
- 3.43. Создание диалоговых окон.

6.4. Перечень видов оценочных средств

Перечень тем лекций для интерактивных занятий
 Лабораторные работы, в том числе в интерактивной форме
 Экзаменационные вопросы
 Курсовой проект

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература

7.1.1. Основная литература

	Авторы,	Заглавие	Издательство,	Кол-во	Эл. адрес
Л1. 1	Шелудько В. М.	Основы программирования на языке высокого уровня Python: учебное пособие	Ростов-на-Дону Таганрог: Южный федеральный университет, 2017	1	http://biblioclub.ru/index.php?page=book&id=500056

7.1.2. Дополнительная литература

	Авторы,	Заглавие	Издательство,	Кол-во	Эл. адрес
Л2. 1	Шелудько В. М.	Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули: учебное пособие	Ростов-на-Дону Таганрог: Южный федеральный университет, 2017	1	http://biblioclub.ru/index.php?page=book&id=500060
Л2. 2	Балджи А. С., Хрипунова М. Б., Александров а И. А.	Математика на Python: учебно-методическое пособие	Москва: Прометей, 2018	1	http://biblioclub.ru/index.php?page=book&id=494849
Л2. 3	Хахаев И. А.	Практикум по алгоритмизации и программированию на Python: курс	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	1	http://biblioclub.ru/index.php?page=book&id=429256
Л2. 4	Златопольский Д. М.	Программирование: типовые задачи, алгоритмы, методы: учебное пособие	Москва: Лаборатория знаний, 2020	1	http://biblioclub.ru/index.php?page=book&id=222873

7.1.3. Методические разработки

	Авторы,	Заглавие	Издательство,	Кол-во	Эл. адрес
Л3. 1	Ефремова А.Н.	Программирование (1 часть): методические указания к выполнению лабораторных работ	Братск: БрГУ, 2020	1	https://ecat.brstu.ru/catalog/Учебные%20и%20учебно-методические%20пособия/Информатика%20-%20Вычислительная%20техника%20-%20Программирование/Ефремова%20А.Н.Программирование.Ч.1.МУк ЛР.2020.pdf

	Авторы,	Заглавие	Издательство,	Кол-во	Эл. адрес
ЛЗ. 2	Ефремова А.Н.	Программирование (II часть): методические указания по выполнению лабораторных работ	Братск: БрГУ, 2021	1	https://ecat.brstu.ru/catalog/Учебные%20и%20учебно-методические%20пособия/Информатика%20-%20Вычислительная%20техника%20-%20Программирование/Ефремова%20А.Н.Программирование.Ч.2.МУкЛР.2021.pdf

7.3.1 Перечень программного обеспечения

7.3.1.1	Python IDLE
7.3.1.2	Adobe Acrobat Reader DC
7.3.1.3	LibreOffice
7.3.1.4	Microsoft Windows Professional 7 Russian Upgrade Academic OPEN No Level
7.3.1.5	Microsoft Office 2007 Russian Academic OPEN No Level
7.3.1.6	Visual Studio Code (VS Code)

7.3.2 Перечень информационных справочных систем

7.3.2.1	Издательство "Лань" электронно-библиотечная система
7.3.2.2	«Университетская библиотека online»
7.3.2.3	Электронный каталог библиотеки БрГУ
7.3.2.4	Электронная библиотека БрГУ

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Аудитория	Назначение	Оснащение аудитории	Вид занятия
2201	читальный зал №1	Комплект мебели (посадочных мест) Стеллажи Комплект мебели (посадочных мест) для библиотекаря Выставочные шкафы ПК i5-2500/H67/4Gb (монитор TFT19 Samsung) (10шт.); принтер HP Laser Jet P2055D (1шт.)	Ср
1343	Учебная аудитория (дисплейный класс)	Основное оборудование: -комплект серверного оборудования для построения технической архитектуры комплекса терминальных решений в составе терминального сервера, терминальных рабочих мест и периферии в составе: - терминальный сервер Dell PowerEdge RX740XD; - тонких клиентов SmartClient Mini PC (Intel CPU J1900 1.99GHzx4, 4GB) - 15шт. - монитор Forgame Liquid Crystal Dispay MK27FC 27" 1800R 1920x1080 144 Hz -15 шт. - вебкамера Logitech C920 PRO, принтер HP LaserJet 1150; - доска интерактивная сенсорная Smart Board SB480 - 1 шт. Дополнительно: - маркерная доска - 1 шт.; Учебная мебель: - комплект мебели (посадочных мест/APM) -20/15 шт.	Лек
1343	Учебная аудитория (дисплейный класс)	Основное оборудование: -комплект серверного оборудования для построения технической архитектуры комплекса терминальных решений в составе терминального сервера, терминальных рабочих мест и периферии в составе: - терминальный сервер Dell PowerEdge RX740XD; - тонких клиентов SmartClient Mini PC (Intel CPU J1900 1.99GHzx4, 4GB) - 15шт. - монитор Forgame Liquid Crystal Dispay MK27FC 27" 1800R 1920x1080 144 Hz -15 шт. - вебкамера Logitech C920 PRO, принтер HP LaserJet 1150; - доска интерактивная сенсорная Smart Board SB480 - 1 шт. Дополнительно: - маркерная доска - 1 шт.; Учебная мебель: - комплект мебели (посадочных мест/APM) -20/15 шт.	Лаб
1343	Учебная аудитория (дисплейный класс)	Основное оборудование: -комплект серверного оборудования для построения технической архитектуры комплекса терминальных решений в составе терминального сервера, терминальных рабочих мест и периферии в составе: - терминальный сервер Dell PowerEdge RX740XD;	Экзамен

		- тонких клиентов SmartClient Mini PC (Intel CPU J1900 1.99GHzx4, 4GB) - 15шт. - монитор Forgame Liquid Crystal Display MK27FC 27" 1800R 1920x1080 144 Hz -15 шт. - вебкамера Logitech C920 PRO, принтер HP LaserJet 1150; - доска интерактивная сенсорная Smart Board SB480 - 1 шт. Дополнительно: - маркерная доска - 1 шт.; Учебная мебель: - комплект мебели (посадочных мест/APM) -20/15 шт.	
1343	Учебная аудитория (дисплейный класс)	Основное оборудование: -комплект серверного оборудования для построения технической архитектуры комплекса терминальных решений в составе терминального сервера, терминальных рабочих мест и периферии в составе: - терминальный сервер Dell PowerEdge RX740XD; - тонких клиентов SmartClient Mini PC (Intel CPU J1900 1.99GHzx4, 4GB) - 15шт. - монитор Forgame Liquid Crystal Display MK27FC 27" 1800R 1920x1080 144 Hz -15 шт. - вебкамера Logitech C920 PRO, принтер HP LaserJet 1150; - доска интерактивная сенсорная Smart Board SB480 - 1 шт. Дополнительно: - маркерная доска - 1 шт.; Учебная мебель: - комплект мебели (посадочных мест/APM) -20/15 шт.	КП

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Учебным планом предусмотрены лекции, лабораторные работы, курсовой проект, самостоятельная работа студента, подготовка и сдача экзамена.

Лекции

- 1) Написание конспекта лекций: кратко, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометить важные мысли, выделять ключевые слова, термины.
- 2) Проверка терминов с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, лабораторной работе.

Лабораторные работы

- 1) Работа с конспектом лекций, обобщение, систематизация, углубление и конкретизация полученных теоретических знаний, выработка способности и готовности их использования на практике.
- 2) Подготовка ответов к контрольным вопросам, работа с основной и дополнительной литературой, необходимой для освоения дисциплины.
- 3) Выполнение заданий, решение задач, активное участие в интерактивной, активной, инновационной формах обучения, составление отчетов.

Курсовой проект

- 1) Подбор, анализ и систематизация необходимого теоретического материала по программированию в соответствии с выданным заданием на курсовой проект.
- 2) Написание и оформление отчета курсового проекта в соответствии с установленными требованиями и методическими указаниями.

Самостоятельная работа обучающихся

- 1) Подготовка к лабораторным работам.
 - а) Проработка основной и дополнительной литературы, терминов, сведений, формул требующихся для запоминания и являющихся основополагающими в разделе.
 - б) Конспектирование прочитанных литературных источников. Проработка материалов по изучаемому вопросу, с использованием на рекомендуемых ресурсах информационно-телекоммуникационной сети «Интернет».
 - в) Выполнение заданий преподавателя, необходимых для подготовки к участию в интерактивной, активной, инновационных формах обучения по изучаемой теме.
- 2) Подготовка к экзамену
 - а) Систематическая работа с конспектом лекций: чтение записей; проверка терминов с помощью энциклопедий, словарей и справочников;
 - б) Обозначение вопросов, материал, которых вызывает трудности; попытка найти ответ в рекомендуемых источниках; подготовка вопросов преподавателю для консультации, если не удастся самостоятельно разобраться в материале.