МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Институт искусственного интеллекта, робототехники и системной инженерии

УТВЕРЖДАЮ
Проректор по образовательной деятельности КФУ
_____ Турилова Е.А.
"___" _____ 2023 г.

Программа дисциплины
Основы Python

Направление подготовки: 15.03.06 - Мехатроника и робототехника
Профиль подготовки: Робототехника и искусственный интеллект
Квалификация выпускника: бакалавр
Форма обучения: очное
Язык обучения: русский
Год начала обучения по образовательной программе: 2024

Содержание

Программу дисциплины разработал: ведущий инженер-программист Фахрутдинов А.Ф. (Научно-исследовательский центр: Центр превосходства Специальная робототехника и искусственный интеллект, Институт вычислительной математики и информационных технологий), timvaz@yandex.ru

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ПК-1	Способен разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

- синтаксис управляющие конструкции языка, структуры данных языка Python;

- основные стандартные модули и библиотеки Python;

- фундаментальные принципы программирования разработки программного обеспечения и алгоритмику на языке Python;

- принципы разработки собственных модулей и библиотек.

Должен уметь:

- работать с основным инструментарием для проектирования, разработки и тестирования программного обеспечения на языке программирования Python;

- использовать как стандартные, так и дополнительные модули, расширения и пакеты при написании программ на Python;

- создавать и импортировать собственные модули;

- следовать основным шаблонам проектирования приложений.

Должен владеть:

- навыками разработки приложений на языке Python, в том числе с использованием библиотек из внешних источников;

- навыками анализа и обработки больших объемов данных, ошибок и отладки;

- инструментами, предоставляемыми библиотеками Python для хранения, обработки, чтения и анализа данных, а также их визуализации;

- навыками объектно-ориентированного программирования.

Должен демонстрировать способность и готовность:

Применять полученные знания и навыки в практической деятельности

Дисциплина связана со следующими дисциплинами: “цифровая обработка сигналов”, “основы программирования C/C++”, “курсовая работа по профилю подготовки”, “компьютерные игры и стрессоустойчивое проектирование”, “основы BigData и DataMining”, “основы машинного обучения”, “нейросети, генеративные платформы и глубокое обучение”, “основы машинного зрения и обработки сенсорных данных”, “технологическая (проектно-технологическая) практика”, “выполнение, подготовка к процедуре защиты и защита выпускной квалификационной работы”.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в раздел "Б1.О.20 Дисциплины (модули)" блока обязательной части 15.03.06 "Мехатроника и робототехника" и относится к части ОПОП ВО, формируемой участниками образовательных отношений.

Осваивается на 2 курсе в 4 семестре.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 4 зачетные единицы на 144 часа.

Контактная работа - 90 часов, в том числе лекции - 36 часов, практические занятия - 54 часа, лабораторные

работы - 0 часов, контроль самостоятельной работы - 0 часов.

Самостоятельная работа - 54 часа.

Форма контроля дисциплины: зачет в 4 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)						Самостоятельная работа
			Лекции, всего	в т.ч. лекции в эл.форме	Практические занятия, всего	в т.ч. практические в эл.форме	Лабораторные работы, всего	в т.ч. лабораторные в эл.форме	
1.	Тема 1. Введение в язык Python.	4	3	0	4	0	0	0	4
2.	Тема 2. Операторы и выражения.	4	3	0	4	0	0	0	4
3.	Тема 3. Условные операторы и циклы.	4	3	0	4	0	0	0	4
4.	Тема 4. Списки, кортежи, словари и множества.	4	3	0	4	0	0	0	4
5.	Тема 5. Функции, модули. Сторонние модули.	4	3	0	4	0	0	0	4
6.	Тема 6. Обработка исключений.	4	3	0	4	0	0	0	4
7.	Тема 7. Строки и регулярные выражения.	4	3	0	5	0	0	0	5
8.	Тема 8. Работа с файлами и ввод-вывод.	4	3	0	5	0	0	0	5
9.	Тема 9. Введение в объектно-ориентированное программирование Python.	4	3	0	5	0	0	0	5
10.	Тема 10. Многопоточность в Python. Callback функции.	4	3	0	5	0	0	0	5
11.	Тема 11. Синхронизация потоков. Семафоры, мьютексы	4	3	0	5	0	0	0	5
12.	Тема 12. Сокеты в Python	4	3	0	5	0	0	0	5
	Итого		36	0	54	0	0	0	54

4.2 Содержание дисциплины (модуля)

Тема 1. Введение в язык Python.

История и происхождение Python. Основные концепции языка Python. Версии Python их особенности. Установка и настройка среды Python.

Тема 2. Операторы и выражения.

Арифметические операторы. Операторы сравнения. Логические операторы. Битовые операторы. Унарные операторы. Приоритеты операций: приоритет арифметических операторов, приоритет битовых операторов, приоритеты логических операторов. Математические функции в выражениях.

Тема 3. Условные операторы и циклы.

Условные операторы if-else и if-elif-else в Python. Вложенные условия. Конструкция match. Циклы for и range() в Python. Циклы while и условия продолжения выполнения. Вложенные циклы. Прерывание циклов.

Тема 4. Списки, кортежи, словари и множества.

Списки в Python: создание, доступ, изменение элементов, срезы, методы работы со списками. Кортежи в Python: особенности, отличия от списков, использование кортежей для передачи аргументов функций. Словари в Python: создание, добавление и получение элементов, итераторы словарей, методы работы со словарями. Множества в Python: основные операции над множествами, сравнение множеств, методы работы с множествами. Генерация уникальных случайных чисел с использованием множеств и генераторов списков.

Тема 5. Функции, модули. Сторонние модули.

Введение в функции и модули в Python: обзор основных понятий и синтаксиса. Создание и использование пользовательских функций: передача аргументов, возврат значений, определение и вызов функций. Модульное программирование и организация кода в Python: преимущества использования модулей, импорт и экспорт функций и переменных. Сторонние (встроенные и пользовательские) модули в Python: установка, использование и создание собственных модулей с расширением функционала языка. Использование сторонних библиотек и пакетов в Python: популярные модули и их применение на примере numpy, pandas, matplotlib, установка и настройка окружения для работы с ними.

Тема 6. Обработка исключений.

Введение в обработку исключений в Python: основные понятия и определения. Виды исключений в Python: SystemError, TypeError, ValueError, IndexError, KeyboardInterrupt и другие. Основы обработки исключений с использованием блоков try-except-else-finally. Углубленная обработка исключений: создание пользовательских исключений, множественная обработка исключений, обработка исключений на верхнем уровне. Исключения и инструменты форматирования и обработки ошибок: встроенные инструменты print(), strip(), assert(), raise() и пользовательские исключения. Примеры применения обработчиков исключений для различных задач.

Тема 7. Строки и регулярные выражения.

Введение в строки и регулярные выражения в Python: основные концепции и понятия. Создание и изменение строк в Python: строковые литералы, строковые переменные, методы для работы со строками (например, split(), join(), replace()).Регулярные выражения в Python: синтаксис, основные метасимволы и квантификаторы, использование регулярных выражений для поиска и замены в строках. Базовые операции с регулярными выражениями: сопоставление строк с шаблоном, нахождение всех вхождений шаблона в строке, замена текста по шаблону. Продвинутые техники работы с регулярными выражениями в Python: обратные ссылки, группы, жадные и ленивые квантификаторы, изучение примеров сложных паттернов и их применение в реальных задачах. F строки.

Тема 8. Работа с файлами и ввод-вывод.

Ввод и вывод в Python: операторы print() и input(), файловые объекты и их методы. Чтение и запись данных в файлы: функции open(), close(), read(), write(), seek(), flush(). Работа с текстовыми файлами. Обработка ошибок при работе с файлами: исключения IOError, OSError, EnvironmentError и их обработка.

Тема 9. Введение в объектно-ориентированное программирование Python.

Основы объектно-ориентированного программирования в Python: определение классов и объектов, инкапсуляция, наследование и полиморфизм. Проектирование классов и интерфейсов: принципы SOLID, паттерны проектирования, использование mixins и ABC. Создание пользовательских классов исключений.

Тема 10. Многопоточность в Python. Callback функции.

Введение в многопоточность в Python. Создание и управление потоками в Python. Создание и использование callback функций в Python для асинхронного выполнения задач. Основной поток приложения и графический интерфейс. Основные проблемы, возникающие при многопоточности.

Тема 11. Синхронизация потоков. Семафоры, мьютексы.

Введение в синхронизацию потоков в Python: основные концепции и необходимость синхронизации. Семафоры в Python: их назначение, виды и реализация. Мьютексы в Python: понятие, использование и примеры решения типичных задач с их применением. Сравнение семафоров и мьютексов: когда и какой инструмент синхронизации потоков использовать в Python. Обработка исключений и взаимной блокировки (deadlock) при использовании механизмов синхронизации потоков в Python, а также способы их предотвращения. Разделяемая память.

Тема 12. Сокеты в Python

Основы сокетов в Python: определение, назначение и виды сокетов. Создание и настройка сокетов. Клиент-серверная архитектура с использованием сокетов в Python: разработка и примеры. Многопоточность и сокеты: оптимизация производительности и масштабируемость. Распространенные ошибки и обработка исключений при работе с сокетами, а также советы по отладке и мониторингу сетевых приложений на Python.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства науки и высшего образования Российской Федерации от 6 апреля 2021 года №245)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);

- критерии оценивания сформированности компетенций;

- механизм формирования оценки по дисциплине (модулю);

- описание порядка применения и процедуры оценивания для каждого оценочного средства;

- критерии оценивания для каждого оценочного средства;

- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение основной и дополнительной учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями;

- в печатном виде - в Научной библиотеке им. Н.И. Лобачевского. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования Научной библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,5 экземпляра (для обучающихся по ФГОС 3++ - не менее 0,25 экземпляра) каждого из изданий основной литературы и не менее 0,25 экземпляра дополнительной литературы на каждого обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов Научной библиотеки КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

База данных научной электронной библиотеки - https://elibrary.ru/

Электронно-библиотечная система Znanium - https://znanium.com/

Документация Python - https://docs.python.org/

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лекции	Лекционные занятия проходят в интерактивной форме, предполагающей вовлечение обучающих в обсуждение всех предложенных тем. Применяются такие формы лекционных занятий как

Вид работ	Методические рекомендации
	лекция-презентация, лекция-дискуссия, проблемная лекция, видео-лекция. Студенты активно участвуют в конструировании знаний во время круглых столов, дискуссионных площадок.
практические занятия	Практические занятия, семинары являются одной из основных форм образовательного процесса, ориентированной на усвоение студентами теоретического материала и выработку практических компетенций. Основной целью практических занятий является комплексный контроль усвоения пройденного материала, хода выполнения студентами самостоятельной работы и рассмотрение наиболее сложных и спорных вопросов в рамках темы занятия. Подготовка к семинарам предполагает самостоятельную работу студентов по изучению материала по конкретной теме.
самостоя-тельная работа	Самостоятельная работа преследует цель закрепить, углубить и расширить знания, полученные студентами в ходе аудиторных занятий, а также сформировать навыки работы с научной, учебной и учебно-методической литературой, развивать творческое, продуктивное мышление обучаемых, их креативные качества, формирование общекультурных и профессиональных компетенций.
зачет	Зачет проводится в письменной форме. В билет включаются тестовые вопросы и задачи из перечня вопросов для подготовки к зачету. Студенту дается 90 минут для выполнения своего варианта зачетного задания.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:

Помещения для самостоятельной работы обучающихся, укомплектованные специализированной мебелью (столы и стулья) и оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду КФУ.

Учебные аудитории для контактной работы с преподавателем, укомплектованные специализированной мебелью (столы и стулья).

Компьютер и принтер для распечатки раздаточных материалов.

Мультимедийная аудитория.

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

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;

- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;

- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;

- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;

- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных

работ, проведения тренингов, организации коллективной работы;

- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;

- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:

- продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;

- продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;

- продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 15.03.06 "Мехатроника и робототехника" и профилю подготовки "Робототехника и искусственный интеллект".

Приложение №1
к рабочей программе дисциплины (модуля)
«Основы Python»

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования
«Казанский (Приволжский) федеральный университет»
Институт искусственного интеллекта, робототехники и системной инженерии

Фонд оценочных средств по дисциплине

Б1.О.20 Основы Pyhton

Направление подготовки: 15.03.06 «Мехатроника и робототехника»
Профиль: Робототехника и искусственный интеллект
Квалификация выпускника: бакалавр
Форма обучения: очная
Язык обучения: русский
Год начала обучения по образовательной программе: 2024

СОДЕРЖАНИЕ

Python. Callback функции", "Синхронизация потоков. Семафоры, мьютексы", "Сокеты в Python."

4.2.2.1. Порядок проведения и процедура оценивания

4.2.2.2. Критерии оценивания

4.2.2.3. Содержание оценочного средства

1. Соответствие компетенций планируемым результатам обучения по дисциплине (модулю)

Код и наименование	**Индикаторы достижения компетенций**	**Оценочные средства текущего контроля и промежуточной аттестации**
ПК-1 Способен разрабатывать программное обеспечение, необходимое для обработки информации и управления в мехатронных и робототехнических системах, а также для их проектирования	ПК-1. И-1: знает основные принципы и методы разработки программного обеспечения для управляющих и информационных систем в мехатронике и робототехнике ПК-1. И-2: умеет разрабатывать программное обеспечение для управления мехатронными и робототехническими системами ПК-1. И-3: владеет навыками проектирования и моделирования мехатронных систем	**Текущий контроль:** Устный опрос по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python”; Задачи по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python”; **Промежуточная аттестация:** Тестирование по темам: “Введение в язык Python”, “Операторы и выражения”,

		“Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python”; Задачи по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python”;

2. Критерии оценивания сформированности компетенций

Компетенция	Зачтено			Не зачтено (0-55 баллов)
	Высокий уровень (86-100 баллов)	**Средний уровень (71-85 баллов)**	**Низкий уровень (56-70 баллов)**	
ПК-1 И-1	Знает принципы организации и управления проектами в области программирования, фундаментальные принципы программирования и разработки программного обеспечения, включая алгоритмику, структуры данных, управление памятью и основы языков программирования.	Знает фундаментальные принципы программирования и разработки программного обеспечения, включая алгоритмику, структуры данных, управление памятью и основы языков программирования.	Знает фундаментальные принципы программирования и разработки программного обеспечения, структуры данных и основы языков программирования.	Знает на крайне низком уровне принципы организации и управления проектами в области программирования, фундаментальные принципы программирования и разработки программного обеспечения, включая алгоритмику, структуры данных, управление памятью и основы языков программирования

ПК-1 И-2	Умеет работать с инструментарием для проектирования, разработки и тестирования программного обеспечения, интерпретировать результаты и принимать решения на основе полученных данных.	Умеет работать с инструментарием для проектирования, разработки и тестирования программного обеспечения.	Умеет работать с инструментарием для разработки программного обеспечения.	Умеет на крайне низком уровне работать с инструментарием для проектирования, разработки и тестирования программного обеспечения, интерпретировать результаты и принимать решения на основе полученных данных.

ПК-1 И-3	Владеет навыками анализа и обработки больших объемов данных, интерпретации результатов и принятия решений на основе полученных данных, языками программирования и методами разработки программного обеспечения, управления проектами.	Владеет навыками анализа и обработки больших объемов данных, интерпретации результатов и принятия решений на основе полученных данных, языками программирования и методами разработки программного обеспечения	На базовом уровне владеет навыками анализа и обработки больших объемов данных, интерпретации результатов и принятия решений на основе полученных данных	Не обладает или владеет на крайне низком уровне навыками анализа и обработки больших объемов данных, интерпретации результатов и принятия решений на основе полученных данных, языками программирования и методами разработки программного обеспечения, управления проектами.

3. Распределение оценок за формы текущего контроля и промежуточную аттестацию

4 семестр:

Текущий контроль:

1. Устный опрос по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python” - 20 баллов
2. Задачи по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python” - 30 баллов

Итого 50 баллов

Промежуточная аттестация – зачет

Зачет проходит в письменной форме. Студенту предоставляется 90 минут на письменный ответ по билету. Каждый билет содержит задание, охватывающее все темы дисциплины, предусмотренные Учебной программой.

Билет состоит из двух частей: теоретической (тестовой) и практической (задачи).

В билет входят:

- Тестирование;
- Задачи;

Первая часть включает в себя 20 тестовых вопросов разных типов. Каждый тестовый вопрос оценивается в 1 балл.

Далее идут 2 задачи, практического характера, выявляющих умение обучающегося анализировать информацию, работать с ней, проводить на ее основе разработку программы. При оценке каждой задачи учитывается полнота ответа, его логичность, правильность решения. Решение каждой задачи оценивается в 15 баллов.

Итоговая оценка за зачет определяется путем суммирования баллов за все правильно выполненные задания билета.

Распределение баллов на зачете с оценкой:

1. Тестирование по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python”- 20 баллов
2. Задачи по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python” - 30 баллов

Итого 50 баллов

Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: 50+50=100 баллов.

Соответствие баллов и оценок:
56-100 – зачтено
0-55 – не зачтено

4. Оценочные средства, порядок их применения и критерии оценивания

4.1. Оценочные средства текущего контроля

4.1.1. Устный опрос по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python.”

4.1.1.1. Порядок проведения и процедура оценивания

В рамках данного курса студенты, помимо изучения теоретического материала и разбора практических примеров должны показать степень усвоения рассмотренного вопроса занятий путем устного опроса. Теоретические материалы и практические примеры студенты совместно с преподавателем изучают на лекционных и практических занятиях соответственно.

4.1.1.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся: - Правильно ответил на все вопросы и обосновал свой ответ.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- Правильно ответил на все вопросы, но при этом не обосновал свой ответ;
- Обосновал свой ответ, но не раскрыл его полностью.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- Ответил не на все вопросы;
- Ответил на все вопросы, но меньшая часть ответов являются ошибочными.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

- Не ответил на большую часть вопросов;

4.1.1.3. Содержание оценочного средства

1. В чем разница между списком и кортежем?
2. Что такое интерполяция строк и как она выполняется?
3. В чем разница между “is” и “==”?
4. Что такое декоратор?
5. Объясните функцию range.
6. Соблюдаются ли все принципы ООП (объектно-ориентированного программирования) в Python?
7. В чем разница между методами экземпляра класса и статическими методами?
8. В чем разница между func и func()?
9. Объясните, как работает функция map.
10. Как работает функция reduce?
11. Объясните, как работает функция filter.
12. Переменные в Python передаются по ссылке или по значению?
13. Как развернуть список?
14. Как работает умножение строк?
15. Как работает умножение списка?
16. Что означает “self” в классе?
17. Как объединить списки в Python?
18. В чем разница между глубокой и мелкой копиями?
19. Как правильно передать callback функцию?
20. В чем разница между списками и массивами?
21. Как объединить два массива?
22. Как округлять вещественное числа до n-го количества знаков?
23. Как разбить список?
24. Где быстрее поиск: в словарях или списках?
25. В чем разница между модулем и пакетом?
26. Как можно увеличить или уменьшить целое число?

27. Как верить двоичный код числа?
28. Как проверить существование значения в списке?
29. В чем разница между append и extend?
30. Как объединить два списка в список кортежей?
31. Как отсортировать словарь по ключам в алфавитном порядке?
32. Как реализуется наследование в Python?
33. Как удалить все пробелы из строки?
34. В чем разница между pass, continue и break?
35. Что такое тернарный оператор?
36. В чем разница между remove, del и pop?
37. Какая конструкция используется при обработке исключений?
38. Что такое итератор и генератор?
39. Чем отличаются iter и next?
40. Что такое контекстный менеджер?
41. Как устроен принцип работы сборщика мусора в Python?
42. Как использовать глобальные переменные? Является ли это хорошей практикой?
43. Для чего в классе используется атрибут slots?
44. Какие пространства имен существуют в Python?
45. Как реализуется управление памятью в Python?
46. Что такое метаклассы и когда их следует использовать?
47. Как создать класс без ключевого слова “class”?
48. Какова роль ветки else в конструкции try-except-else?
49. Какой порядок разрешения методов в Python при наследовании?
50. Что такое дескрипторы?
51. Что такое виртуальное окружение?
52. Что такое многопоточность в контексте программирования на Python?
53. Какие существуют подходы к многопоточности в Python?
54. Как создавать потоки в Python?
55. Что такое приоритет потоков и как им управлять?
56. Каковы основные преимущества использования многопоточности в Python?
57. В чем заключаются основные проблемы и ограничения многопоточности в Python?
58. Как синхронизировать доступ к общим ресурсам между потоками?
59. Как можно реализовать кооперацию между потоками с помощью механизмов блокировки и условных переменных?
60. Как осуществить чтение и запись данных в файлы с использованием встроенных функции в Python?
61. Какие режимы открытия файлов доступны в Python и как они используются?
62. В чем разница между текстовыми и двоичными файлами в Python?
63. Что такое сокеты в Python?
64. Какие типа сокетов доступны в Python?
65. Каковы методы для взаимодействия с сокетами в Python?
66. Какими значениями описывается сокет узла?

67. В чем разница между блокирующим и неблокирующим режимами сокетов Python?

68. Что нужно учесть при масштабировании приложений с использованием сокетов Python?

69. Как можно описать локальный сокет?

70. Какая библиотека подходит для запаковывания (распаковывания) данных разных типов в один пакет, предназначенный для передачи через сокеты?

4.1.2. Задачи по темам: "Введение в язык Python", "Операторы и выражения", "Условные операторы и циклы", "Списки, кортежи, словари и множества", "Функции, модули. Сторонние модули", "Обработка исключений", "Строки и регулярные выражения", "Работа с файлами и ввод-вывод", "Введение в объектно-ориентированное программирование Python", "Многопоточность в Python. Callback функции", "Синхронизация потоков. Семафоры, мьютексы", "Сокеты в Python".

4.1.2.1. Порядок проведения и процедура оценивания

Задачи являются одной из форм текущего контроля. Задачи включают в себя задания, которые охватывают все темы курса, поэтому соответствуют ПК-1.
Каждый из вариантов включает в себя 2 задачи, каждый из которых оценивается в 15 баллов. В случае неверно выполненного задания результат ответа признается равным 0.
Общий итог тестирования рассчитывается путем суммирования баллов за правильные ответы. Задачи даются в конце семестра после того, как обучающиеся освоили все темы курса.

4.1.2.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- студент полностью решил обе задачи;
- студент полностью решил одну задачу и с небольшими ошибками вторую.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- студент с небольшими ошибками решил обе задачи;
- студент полностью решил одну задачу и со значимыми ошибками вторую.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- студент полностью решил только одну задачу;
- студент со значимыми ошибками решил одну задачу и с небольшими ошибками решил вторую.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся: - студент не решил ни одной задачи;

- студент со значимыми ошибками решил обе задачи;
- студент не решил одну задачу и со значимыми ошибками решил вторую.

4.1.2.3. Содержание оценочного средства

Пример вариантов заданий:

ВАРИАНТ 1.

1. Создайте систему управления библиотекой. Разработайте классы "Book", "Author", "Library". Разработайте взаимодействие между объектами этих классов, включая добавление/удаление книг, поиск по автору или названию.

2. Реализуйте простой TCP сервер, который будет принимать сообщения от клиентов и отправлять их обратно.

ВАРИАНТ 2.

1. Создайте систему управления зоопарком. Разработайте классы "Animal", "Zoo", "Keeper". Разработайте взаимодействие между объектами этих классов, включая добавление/удаление животных, учет сотрудников, расписание кормления.

2. Используйте TCP-сокеты для создания системы навигации. Сервис должен поддерживать многопользовательские сессии, а сервер должен контролировать хранить координаты всех клиентов. Клиенты должны передавать свои координаты раз в 50 миллисекунд.

База заданий

1. Создайте систему управления библиотекой. Разработайте классы "Book", "Author", "Library". Разработайте взаимодействие между объектами этих классов, включая добавление/удаление книг, поиск по автору или названию.

2. Создайте систему управления продажами в магазине. Разработайте классы "Product", "Customer", "Seller" и "Sale". Разработайте взаимодействие между объектами этих классов, включая добавление/удаление продуктов, продажу товаров, учет клиентов.

3. Создайте систему управления банковским счетом. Разработайте классы "Account", "Owner", "Transaction". Разработайте взаимодействие между объектами этих классов, включая создание/закрытие счета, перевод средств, просмотр истории транзакций.

4. Создайте систему управления школой. Разработайте классы "Student", "Teacher", "Subject", "Classroom". Разработайте взаимодействие между объектами этих классов, включая регистрацию студентов, назначение учителей, создание расписания.

5. Создайте систему управления автосервисом. Разработайте классы "Car", "Mechanic", "Repair", "Parts". Разработайте взаимодействие между объектами этих классов, включая регистрацию автомобилей, назначение механиков, выполнение ремонта.

6. Создайте систему управления гостиницей. Разработайте классы "Room", "Guest", "Reservation". Разработайте взаимодействие между объектами этих классов, включая бронирование номеров, регистрацию гостей, управление статусом номеров.

7. Создайте систему управления складом. Разработайте классы "Item", "Warehouse", "Supplier", "Order". Разработайте взаимодействие между объектами этих классов, включая добавление/удаление товаров, учет поставщиков, создание заказов.

8. Создайте систему управления зоопарком. Разработайте классы "Animal", "Zoo", "Keeper". Разработайте взаимодействие между объектами этих классов, включая добавление/удаление животных, учет сотрудников, расписание кормления.

9. Создайте систему управления больницей. Разработайте классы "Patient", "Doctor", "Appointment". Разработайте взаимодействие между объектами этих классов, включая регистрацию пациентов, назначение врачей, создание расписания приема.

10. Создайте систему управления аэропортом. Разработайте классы "Flight", "Passenger", "Crew", "Airport". Разработайте взаимодействие между объектами этих классов, включая добавление/удаление рейсов, регистрацию пассажиров, учет экипажа.

11. Реализуйте простой TCP сервер, который будет принимать сообщения от клиентов и отправлять их обратно.

12. Напишите UDP-сервер, который будет принимать сообщения от клиентов и отправлять их обратно.

13. Используйте TCP-сокеты для создания простого файл-сервера, который позволит загружать и скачивать бинарные файлы.

14. Разработайте чат-сервер и клиент на основе UDP. Пользователи должны иметь возможность отправлять и получать сообщения в реальном времени.

15. Используйте TCP-сокеты для создания системы навигации. Сервис должен поддерживать многопользовательские сессии, а сервер должен контролировать хранить координаты всех клиентов. Клиенты должны передавать свои координаты раз в 50 миллисекунд.

16. Разработайте систему обмена сообщениями на основе TCP, которая обеспечивает доставку сообщений. Система должна гарантировать, что сообщение достигнет получателя, даже если это потребует нескольких попыток.

17. Разработайте TCP-сервер, который будет обслуживать несколько клиентов одновременно. Каждый клиент должен быть обслужен в отдельном потоке.

4.2. Оценочные средства промежуточной аттестации

Зачет проходит в письменной форме. Студенту предоставляется 90 минут на письменный ответ по билету. Каждый билет содержит задание, охватывающее все темы дисциплины, предусмотренные Учебной программой.

Билет состоит из двух частей: теоретической (тестовой) и практической (задачи). В билет входят:

- Тестирование;
- Задачи;

Первая часть включает в себя 20 тестовых вопросов. Каждый тестовый вопрос оценивается в 1 балл.

Далее идут две задачи, в каждой из которых показывает умение обучающегося анализировать информацию, работать с ней, разрабатывать на основе нее программное обеспечение. При оценке каждой задачи учитывается полнота ответа, его логичность, правильность решения. Решение каждой задачи оценивается в 15 баллов.

Итоговая оценка за зачет определяется путем суммирования баллов за все правильно выполненные задания билета.

Результат зачета оценивается следующим образом:

56 – 100 баллов - «зачтено»

55 баллов и менее - «не зачтено».

4.2.1. Тестирование по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python”.

4.2.1.1. Порядок проведения и процедура оценивания

Тестирование является одной из форм промежуточной аттестации. Тестирование включает Тестовые вопросы, которые охватывают все темы курса, поэтому соответствуют ПК-1. Тесты могут включать в себя вопросы с одним или множественным выбором.

Каждый из тестовых вариантов включает в себя 20 вопросов, каждый из которых оценивается в 1 балл. В случае частичного или неверно выполненного задания результат ответа признается равным 0.

Общий итог тестирования рассчитывается путем суммирования баллов за правильные ответы. Тестирование проводится в конце семестра после того, как обучающиеся освоили все темы курса в рамках зачетного билета.

4.2.1.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- студент дал правильные ответы на 17-20 вопросов теста;

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- студент дал правильные ответы на 14-17 вопросов теста;

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- студент дал правильные ответы на 11-14 вопросов теста;

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся: - студент дал правильные ответы на 10 или менее вопросов теста.

4.2.1.3. Содержание оценочного средства

Пример вариантов тестирования:

ВАРИАНТ 1.

1. Какой оператор используется для объединения двух или более строк в Python?
   А) +
   Б) *
   В) &
   Г) /
2. Что делает следующий фрагмент кода? i = 5 if i == 4: print(‘i равно 4’) else: print(‘i не равно 4’)
   А) Ничего не делает
   Б) Всегда выводит ‘i не равно 4’
   В) Всегда выводит ‘i равно 4’
   Г) Выводит ‘i равно 4’, если i равно 5
   Д) Выводит ‘i не равно 4’, если i не равно 5
3. Какой оператор используется для проверки нескольких условий и выполнения соответствующего блока кода?
   А) if
   Б) else
   В) elif
   Г) while
4. Какой оператор используется для проверки неравенства в Python?
   А) ==

Б) =
В) !=
Г) <>

5. Какой оператор используется для проверки принадлежности значения к списку или кортежу?
   А) in
   Б) is
   В) not
   Г) for
6. Как объявить пустой список в Python?
   А) []
   Б) {}
   В) ()
   Г) set()
7. Как удалить элемент из списка в Python?
   А) delete()
   Б) remove()
   В) pop()
   Г) discard()
8. Как объявить функцию в Python?
   А) def
   Б) func
   В) define
   Г) function
9. Что такое исключение в Python?
   А) Ошибка в синтаксисе программы
   Б) Непредвиденное событие или ошибка, возникающая во время выполнения программы
   В) Предупреждение о потенциальной ошибке
   Г) Результат работы функции
10. Как получить длину строки в Python?
    А) length()
    Б) count()
    В) size()
    Г) len()
11. Какой специальный символ используется для указания шаблона в регулярном выражении в Python?
    А) \
    Б) $
    В) #
    Г) @
12. Какой метод используется для добавления данных в конец файла в Python?
    А) read()
    Б) write()
    В) append()
    Г) close()

13. Какой метод используется для удаления файла в Python?
    А) delete()
    Б) remove()
    В) erase()
    Г) destroy()
14. Что такое полиморфизм в объектно-ориентированном программировании?
    А) Возможность объекта иметь разные типы данных
    Б) Способность объекта изменять свое состояние
    В) Возможность объекта принимать разные формы
    Г) Способность объекта выполнять разные действия
15. Как создать экземпляр класса в Python?
    А) MyClass()
    Б) create MyClass
    В) new MyClass
    Г) instance = MyClass
16. Как создать новый поток в Python?
    А) thread.start()
    Б) threading.Thread()
    В) process.start()
    Г) multiprocessing.Process()
17. Что такое callback функция в Python?
    А) Функция, переданная в другую функцию в качестве аргумента, которая затем вызывается по завершению какого-либо действия.
    Б) Функция, вызываемая при возникновении исключения
    В) Функция, вызываемая при старте программы
    Г) Функция, вызываемая при импорте модуля
18. Что такое синхронизация потоков в Python?
    А) Механизм, позволяющий потокам выполняться параллельно
    Б) Механизм, позволяющий потокам обмениваться данными
    В) Механизм, позволяющий контролировать доступ потоков к общим ресурсам
    Г) Механизм, позволяющий управлять жизненным циклом потоков
19. Какой метод в Python используется для отправки данных через сокет?
    А) send()
    Б) receive()
    В) connect()
    Г) accept()
20. Что такое порт в контексте сетевого программирования?
    А) Число, которое идентифицирует конкретное сетевое приложение на устройстве
    Б) Число, которое идентифицирует конкретный сокет
    В) Число, которое идентифицирует конкретный процесс
    Г) Число, которое идентифицирует конкретный пакет данных

ВАРИАНТ 2.

1. Какой тип данных используется для представления целых чисел в Python?
    А) int

Б) float
В) str
Г) bool

2. Какой символ используется в Python для обозначения комментариев?
   А) //
   Б) --
   В) #
   Г) /* */
3. Какой оператор используется для проверки истинности условия и выполнения блока кода, если условие не выполняется в Python?
   А) if
   Б) else
   В) elif
   Г) while
4. Какой оператор используется для проверки логического И в Python?
   А) &&
   Б) ||
   В) and
   Г) or
5. Какие скобки используются для объявления словаря в Python?
   А) []
   Б) {}
   В) ()
   Г) <>
6. Как получить значение из словаря по ключу в Python?
   А) get()
   Б) value()
   В) fetch()
   Г) retrieve()
7. Как проверить наличие элемента в множестве в Python?
   А) check()
   Б) contains()
   В) contains_element()
   Г) in
8. Какой оператор используется для возврата значения из функции в Python?
   А) return
   Б) yield
   В) exit
   Г) break
9. Какой блок кода выполняется всегда, независимо от того, возникло исключение или нет?
   А) try
   Б) except
   В) raise
   Г) finally
10. Какой метод используется для разделения строки на подстроки в Python?

А) split()
Б) join()
В) replace()
Г) strip()

11. Какой метод используется для поиска совпадений с регулярным выражением в строке в Python?
А) search()
Б) match()
В) findall()
Г) find()
12. Какой метод используется для записи данных в файл в Python?
А) read()
Б) write()
В) append()
Г) close()
13. Какой оператор используется для открытия файла в Python?
А) read()
Б) open()
В) close()
Г) write()
14. Что такое объект в объектно-ориентированном программировании?
А) Переменная
Б) Функция
В) Экземпляр класса
Г) Условный оператор
15. Какой метод в классе Python вызывается при создании нового экземпляра класса?
А) init()
Б) create()
В) new()
Г) instance()
16. Какой модуль в Python используется для работы с многопоточностью?
А) thread
Б) threading
В) multiprocessing
Г) concurrent
17. Как передать callback функцию в другую функцию в Python?
А) В качестве аргумента функции
Б) Возвращая ее из другой функции
В) Через глобальную переменную
Г) Через декоратор
18. Какая основная разница между семафорами и мьютексами в Python?
А) Семафоры позволяют нескольким потокам одновременно получить доступ к ресурсу, а мьютексы - только одному потоку
Б) Семафоры используют блокировку, а мьютексы - счетчик

В) Семафоры поддерживают только два состояния (заблокирован и разблокирован), а мьютексы - множество состояний

Г) Семафоры используются только для синхронизации потоков, а мьютексы - для синхронизации процессов

19. Что такое сокеты в контексте сетевого программирования?

А) Механизм, который позволяет программам обмениваться данными через сеть

Б) Механизм, который позволяет программам работать с базами данных

В) Механизм, который позволяет программам работать с файлами

Г) Механизм, который позволяет программам выводить информацию на экран

20. Какой метод в Python используется для закрытия сокета?

А) close()

Б) shutdown()

В) disconnect()

Г) terminate()

База тестовых вопросов для тестирования

1. Какой тип данных является встроенным типом данных в Python?

А) int

Б) float

В) str

Г) bool

Д) None

2. Какая структура данных является последовательной в Python?

А) список

Б) кортеж

В) словарь

Г) множество

Д) файл

3. Что из нижеперечисленного является объектом в Python?

А) переменная

Б) функция

В) класс

Г) исключение

Д) все из вышеперечисленного

4. Что делает следующая строка кода? a = ‘Привет, мир!’ print(a)

А) Выводит ‘Привет’ на экран

Б) Выводит ‘, мир!’ на экран

В) Выводит оба на экран

Г) Ничего не выводит

Д) Генерирует исключение

5. Что делает следующий фрагмент кода? i = 5 if i == 4: print(‘i равно 4’) else: print(‘i не равно 4’)

А) Ничего не делает

Б) Всегда выводит ‘i не равно 4’

В) Всегда выводит ‘i равно 4’

Г) Выводит ‘i равно 4’, если i равно 5

Д) Выводит ‘i не равно 4’, если i не равно 5

6. Какой оператор используется для объединения двух или более строк в Python?
   А) +
   Б) *
   В) &
   Г) /
7. Какой тип данных используется для представления целых чисел в Python?
   А) int
   Б) float
   В) str
   Г) bool
8. Какой символ используется в Python для обозначения комментариев?
   А) //
   Б) --
   В) #
   Г) /* */
9. Что делает оператор + когда он используется с различными типами данных в Python?
   А) Сложение чисел
   Б) Конкатенация строк
   В) Объединение множеств
   Г) Ничего из перечисленного
   Д) Все из перечисленных
10. Какой оператор используется для выполнения целочисленного деления в Python?
    А) /
    Б) //
    В) %
    Г) *
11. Какой оператор используется для возведения числа в степень в Python?
    А) ^
    Б) **
    В) //
    Г) *
12. Какой оператор используется для проверки неравенства в Python?
    А) ==
    Б) =
    В) !=
    Г) <>
13. Какой оператор используется для объединения двух строк в Python?
    А) +
    Б) *
    В) &
    Г) /
14. Какой оператор используется для проверки логического И в Python?
    А) &&

Б) ||
В) and
Г) or

15. Какой оператор используется для проверки истинности выражения в Python?
А) if
Б) for
В) while
Г) assert
16. Какой оператор используется для вычисления остатка от деления в Python?
А) /
Б) //
В) %
Г) *
17. Какой оператор используется для проверки принадлежности значения к списку или кортежу в Python?
А) in
Б) is
В) not
Г) for
18. Какой оператор используется для проверки истинности условия и выполнения блока кода, если условие не выполняется в Python?
А) if
Б) else
В) elif
Г) while
19. Какой оператор используется для увеличения значения переменной на 1 в Python?
А) +=
Б) -=
В) *=
Г) ++
20. Какой оператор используется для выполнения блока кода только при выполнении определенного условия?
А) if
Б) for
В) while
Г) switch
21. Какой оператор используется для выполнения блока кода, если условие не выполняется?
А) if
Б) else
В) elif
Г) while
22. Какой оператор используется для проверки нескольких условий и выполнения соответствующего блока кода?

А) if
Б) else
В) elif
Г) while

23. Какой оператор используется для выполнения блока кода несколько раз до тех пор, пока условие истинно?
А) if
Б) else
В) elif
Г) while
24. Какой оператор используется для выполнения блока кода для каждого элемента в итерируемом объекте?
А) if
Б) else
В) elif
Г) for
25. Какой оператор используется для выхода из цикла и перехода к следующей итерации, если условие истинно?
А) break
Б) continue
В) return
Г) exit
26. Какой оператор используется для выполнения блока кода только в том случае, если все условия истинны?
А) and
Б) or
В) not
Г) all
27. Какой оператор используется для выполнения блока кода, если хотя бы одно условие истинно?
А) and
Б) or
В) not
Г) any
28. Какой оператор используется для проверки принадлежности значения к списку или кортежу?
А) in
Б) is
В) not
Г) for
29. Какой оператор используется для выполнения блока кода определенное количество раз?
А) if
Б) else
В) elif

Г) for

30. Как объявить пустой список в Python?
    А) []
    Б) {}
    В) ()
    Г) set()
31. Как получить длину списка или кортежа в Python?
    А) length()
    Б) count()
    В) size()
    Г) len()
32. Какие скобки используются для объявления списка в Python?
    А) []
    Б) {}
    В) ()
    Г) <>
33. Какие скобки используются для объявления кортежа в Python?
    А) []
    Б) {}
    В) ()
    Г) <>
34. Какие скобки используются для объявления словаря в Python?
    А) []
    Б) {}
    В) ()
    Г) <>
35. Как добавить элемент в список в Python?
    А) add()
    Б) insert()
    В) append()
    Г) extend()
36. Как получить значение из словаря по ключу в Python?
    А) get()
    Б) value()
    В) fetch()
    Г) retrieve()
37. Как объединить два множества в Python?
    А) union()
    Б) merge()
    В) combine()
    Г) join()
38. Как удалить элемент из списка в Python?
    А) delete()
    Б) remove()
    В) pop()

Г) discard()

39. Как проверить наличие элемента в множестве в Python?
    А) check()
    Б) contains()
    В) contains_element()
    Г) in
40. Что такое функция в Python?
    А) Библиотека стандартных функций
    Б) Специальная переменная
    В) Блок кода, который выполняет определенную задачу
    Г) Часть оператора if
41. Как объявить функцию в Python?
    А) def
    Б) func
    В) define
    Г) function
42. Какой оператор используется для возврата значения из функции в Python?
    А) return
    Б) yield
    В) exit
    Г) break
43. Какой модуль используется для работы с датами и временем в Python?
    А) math
    Б) time
    В) datetime
    Г) random
44. Как подключить сторонний модуль в Python?
    А) import
    Б) include
    В) require
    Г) load
45. Какой модуль используется для работы с файлами в Python?
    А) os
    Б) sys
    В) file
    Г) io
46. Какой модуль используется для работы с математическими функциями в Python?
    А) math
    Б) random
    В) calc
    Г) number
47. Какой оператор используется для вызова функции в Python?
    А) call
    Б) invoke

В) run
Г) ()

48. Какой модуль используется для работы с регулярными выражениями в Python?
А) re
Б) regex
В) match
Г) pattern

49. Какой оператор используется для проверки принадлежности имени к модулю в Python?
А) import
Б) from
В) in
Г) is

50. Что такое исключение в Python?
А) Ошибка в синтаксисе программы
Б) Непредвиденное событие или ошибка, возникающая во время выполнения программы
В) Предупреждение о потенциальной ошибке
Г) Результат работы функции

51. Какой оператор используется для обработки исключений в Python?
А) try-except
Б) if-else
В) for-in
Г) while

52. Какой блок кода выполняется, если исключение не возникает в блоке try?
А) except
Б) finally
В) else
Г) raise

53. Какой оператор используется для генерации исключения в Python?
А) try
Б) except
В) raise
Г) finally

54. Какой блок кода выполняется всегда, независимо от того, возникло исключение или нет?
А) try
Б) except
В) raise
Г) finally

55. Какой оператор используется для указания типа исключения, которое нужно обработать?
А) try
Б) except

В) raise
Г) finally

56. Какой оператор используется для обработки нескольких типов исключений одновременно?
А) try
Б) except
В) raise
Г) finally

57. Какой блок кода используется для выполнения определенных действий перед выходом из блока try-except?
А) try
Б) except
В) raise
Г) finally

58. Какой оператор используется для перехвата всех типов исключений в Python?
А) try
Б) except
В) raise
Г) finally

59. Какой оператор используется для передачи исключения другому участку кода?
А) try
Б) except
В) raise
Г) finally

60. Как объявить строку в Python?
А) “Hello World”
Б) ‘Hello World’
В) Hello World
Г) [Hello World]

61. Как получить длину строки в Python?
А) length()
Б) count()
В) size()
Г) len()

62. Какой оператор используется для конкатенации строк в Python?
А) +
Б) -
В) /
Г) *

63. Какой метод используется для преобразования строки в верхний регистр в Python?
А) upper()
Б) lower()
В) capitalize()
Г) swapcase()

64. Какой метод используется для разделения строки на подстроки в Python?
    А) split()
    Б) join()
    В) replace()
    Г) strip()
65. Какой метод используется для проверки начала строки определенной подстрокой в Python?
    А) startswith()
    Б) endswith()
    В) contains()
    Г) find()
66. Какой метод используется для замены подстроки в строке в Python?
    А) replace()
    Б) insert()
    В) append()
    Г) update()
67. Какой модуль используется для работы с регулярными выражениями в Python?
    А) re
    Б) regex
    В) match
    Г) pattern
68. Какой метод используется для поиска совпадений с регулярным выражением в строке в Python?
    А) search()
    Б) match()
    В) findall()
    Г) find()
69. Какой специальный символ используется для указания шаблона в регулярном выражении в Python?
    А) \
    Б) $
    В) #
    Г) @
70. Какой оператор используется для открытия файла в Python?
    А) read()
    Б) open()
    В) close()
    Г) write()
71. Какой режим открытия файла используется для чтения содержимого файла?
    А) r
    Б) w
    В) a
    Г) x
72. Какой метод используется для чтения содержимого файла в Python?

А) read()
Б) write()
В) append()
Г) close()

73. Какой метод используется для записи данных в файл в Python?
А) read()
Б) write()
В) append()
Г) close()

74. Какой метод используется для добавления данных в конец файла в Python?
А) read()
Б) write()
В) append()
Г) close()

75. Какой метод используется для перемещения указателя файла в определенную позицию?
А) seek()
Б) move()
В) position()
Г) index()

76. Какой метод используется для проверки наличия файла в Python?
А) exists()
Б) open()
В) check()
Г) create()

77. Какой метод используется для удаления файла в Python?
А) delete()
Б) remove()
В) erase()
Г) destroy()

78. Какой функцией можно считать введенную пользователем строку в Python?
А) input()
Б) print()
В) read()
Г) write()

79. Какой функцией можно вывести результат на экран в Python?
А) input()
Б) print()
В) read()
Г) write()

80. Что такое объект в объектно-ориентированном программировании?
А) Переменная
Б) Функция
В) Экземпляр класса
Г) Условный оператор

81. Что такое класс в объектно-ориентированном программировании?
    А) Ошибка в программе
    Б) Группа переменных
    В) Контейнер для функций
    Г) Шаблон для создания объектов
82. Что такое наследование в объектно-ориентированном программировании?
    А) Процесс создания нового класса на основе существующего класса
    Б) Процесс изменения существующего класса
    В) Процесс объединения нескольких классов в один
    Г) Процесс создания объекта на основе класса
83. Что такое полиморфизм в объектно-ориентированном программировании?
    А) Возможность объекта иметь разные типы данных
    Б) Способность объекта изменять свое состояние
    В) Возможность объекта принимать разные формы
    Г) Способность объекта выполнять разные действия
84. Что такое инкапсуляция в объектно-ориентированном программировании?
    А) Способность объекта быть доступным из любой части программы
    Б) Способность объекта скрывать свою внутреннюю реализацию
    В) Способность объекта изменять свое поведение во время выполнения
    Г) Способность объекта сохранять свое состояние после завершения программы
85. Как объявить класс в Python?
    А) class MyClass:
    Б) def MyClass:
    В) class = MyClass
    Г) def = MyClass
86. Как создать экземпляр класса в Python?
    А) MyClass()
    Б) create MyClass
    В) new MyClass
    Г) instance = MyClass
87. Как объявить метод в классе в Python?
    А) def my*method():*
    *Б) method my*method():
    В) def my*method(self):*
    *Г) def = my*method():
88. Как получить доступ к атрибуту объекта в Python?
    А) object.attribute
    Б) object.method()
    В) object.attribute()
    Г) object.method
89. Какой метод в классе Python вызывается при создании нового экземпляра класса?
    А) init()
    Б) create()

В) new()
Г) instance()

90. Что такое многопоточность в Python?
А) Возможность запускать несколько программ одновременно
Б) Возможность запускать несколько функций одновременно
В) Возможность запускать несколько потоков одновременно
Г) Возможность запускать несколько процессов одновременно
91. Какой модуль в Python используется для работы с многопоточностью?
А) thread
Б) threading
В) multiprocessing
Г) concurrent
92. Что такое поток в многопоточности?
А) Часть программы, выполняющаяся параллельно
Б) Функция, вызываемая асинхронно
С) Компонент, объединяющий несколько процессов
Д) Переменная, обновляемая из разных потоков
93. Как создать новый поток в Python?
А) thread.start()
Б) threading.Thread()
В) process.start()
Г) multiprocessing.Process()
94. Как остановить выполнение потока в Python?
А) thread.stop()
Б) threading.Thread.stop()
В) thread.terminate()
Г) threading.Thread.join()
95. Что такое callback функция в Python?
А) Функция, переданная в другую функцию в качестве аргумента, которая затем вызывается по завершению какого-либо действия.
Б) Функция, вызываемая при возникновении исключения
В) Функция, вызываемая при старте программы
Г) Функция, вызываемая при импорте модуля
96. Как передать callback функцию в другую функцию в Python?
А) В качестве аргумента функции
Б) Возвращая ее из другой функции
В) Через глобальную переменную
Г) Через декоратор
97. Какой тип данных используется для представления callback функции в Python?
А) int
Б) list
В) str
Д) function
98. Какой метод в Python используется для запуска потока?

А) start()
Б) run()
В) execute()
Г) launch()

99. Какой метод в Python используется для ожидания завершения потока?
А) join()
Б) wait()
В) sleep()
Г) stop()

100. Что такое синхронизация потоков в Python?
А) Механизм, позволяющий потокам выполняться параллельно
Б) Механизм, позволяющий потокам обмениваться данными
В) Механизм, позволяющий контролировать доступ потоков к общим ресурсам
Г) Механизм, позволяющий управлять жизненным циклом потоков

101. Что такое семафоры в синхронизации потоков?
А) Механизм, позволяющий контролировать доступ потоков к общим ресурсам с помощью счетчика
Б) Механизм, позволяющий потокам параллельно выполняться
В) Механизм, позволяющий потокам обмениваться данными
Г) Механизм, позволяющий управлять жизненным циклом потоков

102. Что такое мьютексы (мьютекс) в синхронизации потоков?
А) Механизм, позволяющий контролировать доступ потоков к общим ресурсам с помощью блокировки
Б) Механизм, позволяющий потокам параллельно выполняться
В) Механизм, позволяющий потокам обмениваться данными
Г) Механизм, позволяющий управлять жизненным циклом потоков

103. Как создать семафор в Python?
А) semaphore = Semaphore()
Б) semaphore = threading.Semaphore()
В) semaphore = threading.Lock()
Г) semaphore = Lock()

104. Как получить доступ к семафору в Python?
А) semaphore.acquire()
Б) semaphore.release()
В) semaphore.wait()
Г) semaphore.notify()

105. Что произойдет, если поток вызывает метод acquire() на семафоре, который уже заблокирован другим потоком?
А) Поток будет заблокирован до тех пор, пока семафор не будет разблокирован
Б) Поток будет продолжать выполнение без ожидания разблокировки семафора
В) Будет выброшено исключение
Г) Поток будет остановлен навсегда

106. Как создать мьютекс в Python?
    А) mutex = Mutex()
    Б) mutex = threading.Semaphore()
    В) mutex = threading.Lock()
    Г) mutex = Lock()
107. Как получить доступ к мьютексу в Python?
    А) mutex.acquire()
    Б) mutex.release()
    В) mutex.wait()
    Г) mutex.notify()
108. Какая основная разница между семафорами и мьютексами в Python?
    А) Семафоры позволяют нескольким потокам одновременно получить доступ к ресурсу, а мьютексы - только одному потоку
    Б) Семафоры используют блокировку, а мьютексы - счетчик
    В) Семафоры поддерживают только два состояния (заблокирован и разблокирован), а мьютексы - множество состояний
    Г) Семафоры используются только для синхронизации потоков, а мьютексы - для синхронизации процессов
109. Какой метод в Python используется для проверки, заблокирован ли семафор или мьютекс?
    А) locked()
    Б) is_locked()
    В) is_blocked()
    Г) is_acquired()
110. Что произойдет, если поток вызывает метод release() на семафоре, который уже разблокирован?
    А) Семафор не изменится
    Б) Семафор снова заблокируется
    В) Будет выброшено исключение
    Г) Поток будет остановлен навсегда
111. Что такое сокеты в контексте сетевого программирования?
    А) Механизм, который позволяет программам обмениваться данными через сеть
    Б) Механизм, который позволяет программам работать с базами данных
    В) Механизм, который позволяет программам работать с файлами
    Г) Механизм, который позволяет программам выводить информацию на экран
112. Как создать сокет в Python?
    А) socket.create_socket()
    Б) socket.bind()
    В) socket.socket()
    Г) socket.connect()
113. Какой протокол используется по умолчанию при создании сокета в Python?
    А) TCP
    Б) UDP
    В) HTTP
    Г) FTP

114. Какой метод в Python используется для привязки сокета к конкретному адресу и порту?
    А) bind()
    Б) connect()
    В) listen()
    Г) send()
115. Что такое IP-адрес в контексте сетевого программирования?
    А) Уникальный идентификатор, присвоенный каждому устройству в сети
    Б) Уникальный идентификатор, присвоенный каждому сокету
    В) Уникальный идентификатор, присвоенный каждому процессу
    Г) Уникальный идентификатор, присвоенный каждому пакету данных
116. Какой метод в Python используется для прослушивания входящих соединений на сокете?
    А) listen()
    Б) bind()
    В) accept()
    Г) connect()
117. Какой метод в Python используется для отправки данных через сокет?
    А) send()
    Б) receive()
    В) connect()
    Г) accept()
118. Какой метод в Python используется для приема данных из сокета?
    А) recv()
    Б) send()
    В) accept()
    Г) connect()
119. Какой метод в Python используется для закрытия сокета?
    А) close()
    Б) shutdown()
    В) disconnect()
    Г) terminate()
120. Что такое порт в контексте сетевого программирования?
    А) Число, которое идентифицирует конкретное сетевое приложение на устройстве
    Б) Число, которое идентифицирует конкретный сокет
    В) Число, которое идентифицирует конкретный процесс
    Г) Число, которое идентифицирует конкретный пакет данных

4.2.2. Задачи по темам: “Введение в язык Python”, “Операторы и выражения”, “Условные операторы и циклы”, “Списки, кортежи, словари и множества”, “Функции, модули. Сторонние модули”, “Обработка исключений”, “Строки и регулярные выражения”, “Работа с файлами и ввод-вывод”, “Введение в объектно-ориентированное программирование Python”, “Многопоточность в Python. Callback функции”, “Синхронизация потоков. Семафоры, мьютексы”, “Сокеты в Python”.

4.2.2.1. Порядок проведения и процедура оценивания

Задачи являются одной из форм промежуточной аттестации. Задачи включают в себя задания, которые охватывают все темы курса, поэтому соответствуют ПК-1.
Каждый из вариантов включает в себя 2 задачи, каждый из которых оценивается в 15 баллов. В случае неверно выполненного задания результат ответа признается равным 0.
Общий итог рассчитывается путем суммирования баллов за правильные ответы. Задачи даются в конце семестра после того, как обучающиеся освоили все темы курса в рамках зачетного билета.

4.2.2.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

- студент полностью решил обе задачи;
- студент полностью решил одну задачу и с небольшими ошибками вторую.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

- студент с небольшими ошибками решил обе задачи;
- студент полностью решил одну задачу и со значимыми ошибками вторую.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

- студент полностью решил только одну задачу;
- студент со значимыми ошибками решил одну задачу и с небольшими ошибками решил вторую.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся: - студент не решил ни одной задачи;
- студент со значимыми ошибками решил обе задачи;
- студент не решил одну задачу и со значимыми ошибками решил вторую.

4.2.2.3. Содержание оценочного средства

Пример вариантов задач:

ВАРИАНТ 1.

1. Создайте массив из 50 элементов, каждым из элементов которого является случайное число. Создайте 2 потока, каждый из которых будет принимать в качестве входных данных половину массива (у каждого потока своя половина). Определите максимальное значение в исходном массиве проведя поиск максимума в каждой из половин в своем потоке. Конечный результат записать в текстовый файл.
2. Имеется следующая система: сервер и 2 клиента. Разработать проект - мини-мессенджер, реализующий отправку сообщений между двумя клиентами через сервер используя сокет программирование.

ВАРИАНТ 2.

1. Создайте сокет приема дейтаграммных сообщений. Создайте массив из 50 элементов, каждым из элементов которого является случайное число. Создайте 2 потока, каждый из которых будет принимать в качестве входных данных половину массива (у каждого потока своя половина). Определите минимальное

значение в исходном массиве проведя поиск минимума в каждой из половин в своем потоке. Конечный результат отправить с помощью дейтаграммного сокета на созданный сокет дейтаграммных сообщений.

2. Имеется следующая система: сервер и 2 клиента. Разработать систему логирования действий, реализующий прием сообщений от двух клиентов на сервере используя сокет программирование. В качестве логов принимать сообщение, имеющего следующую структуру: <IP адрес><номер порта><дата и время><номер клиента>.

База задач

1. Создайте массив из 50 элементов, каждым из элементов которого является случайное число. Создайте 2 потока, каждый из которых будет принимать в качестве входных данных половину массива (у каждого потока своя половина). Определите максимальное значение в исходном массиве проведя поиск максимума в каждой из половин в своем потоке. Конечный результат записать в текстовый файл.
2. Имеется следующая система: сервер и 2 клиента. Разработать проект - мини-мессенджер, реализующий отправку сообщений между двумя клиентами через сервер используя сокет программирование.
3. Создайте сокет приема дейтаграммных сообщений. Создайте массив из 50 элементов, каждым из элементов которого является случайное число. Создайте 2 потока, каждый из которых будет принимать в качестве входных данных половину массива (у каждого потока своя половина). Определите минимальное значение в исходном массиве проведя поиск минимума в каждой из половин в своем потоке. Конечный результат отправить с помощью дейтаграммного сокета на созданный сокет дейтаграммных сообщений и вывести на экран.
4. Имеется следующая система: сервер и 2 клиента. Разработать систему логирования действий, реализующий прием сообщений от двух клиентов на сервере используя сокет программирование. В качестве логов принимать сообщение, имеющего следующую структуру: <IP адрес><номер порта><дата и время><номер клиента>.
5. Ученикам, желающим запомнить правила написания слов в английском языке, часто напоминают следующее рифмованное одностишие: «I before E except after C» (I перед E, если не после C). Это правило позволяет запомнить, в какой последовательности писать буквы I и E, идущие в слове одна за другой, а именно: буква I должна предшествовать букве E, если непосредственно перед ними не стоит буква C. Если стоит – порядок гласных будет обратным. Примеры слов, на которые действует это правило: believe, chief, fierce, friend, ceiling и receipt. Но есть и исключения из этого правила, и одним из них является слово weird (странный). Напишите программу, которая будет построчно обрабатывать текстовый файл. В каждой строке может присутствовать много слов, а может и не быть ни одного. Слова, в которых буквы E и I не соседствуют друг с другом, обработке подвергать не следует. Если же такое соседство присутствует, необходимо проверить, соответствует ли написание анализируемого слова указанному выше правилу. Создайте и выведите на экран два списка. В первом должны располагаться слова, следующие правилу, а во втором – нарушающие

его. При этом списки не должны содержать повторяющиеся слова. Также отобразите на экране длину каждого списка, чтобы пользователю было понятно, сколько слов в файле не отвечает правилу.

6. Написать программу, которая будет удалять все комментарии из исходного файла с кодом на языке Python. Обнаружив комментарий, программа должна удалить все содержимое комментария. Сохраните новое содержимое в созданном файле. Имена файла источника и файла назначения должны быть запрошены у пользователя. Удостоверьтесь в том, что программа корректно обрабатывает возможные ошибки при работе с обоими файлами.
7. Разработайте сервер, который будет использовать многопоточность для мониторинга нескольких клиентов в сети одновременно. Сервер должен получать статусные сообщения от устройств и обрабатывать их. Используйте ООП для реализации функциональности сервера и TCP сокеты для обмена сообщениями с клиентами.
8. Разработайте сервер, который будет использовать многопоточность для обработки команд от нескольких клиентов одновременно. Клиенты могут отправлять команды на сервер, сервер выполняет команды и возвращает результаты. Команды клиента представляют собой числа. В качестве результата обработки сервера возвращать факториал полученного числа. Клиенты при получении ответа должны вывести ответ сервера на экран. Используйте ООП для реализации функциональности сервера и TCP сокеты для обмена сообщениями с клиентами.
9. Разработайте сервер, который будет использовать многопоточность для обработки команд от нескольких клиентов одновременно. Клиенты могут отправлять команды на сервер, сервер выполняет команды и возвращает результаты. Команды клиента представляют собой строку вида: “Число для обработки <Число> .”. В качестве результата обработки сервера возвращать число, которое было получено. Клиенты при получении ответа должны вывести ответ сервера на экран. Используйте ООП для реализации функциональности сервера и TCP сокеты для обмена сообщениями с клиентами.
10. Написать программу, которая заменять комментарии на вызов функции print с содержимым комментария из исходного файла с кодом на языке Python. Обнаружив комментарий, программа должна удалить все содержимое комментария. Сохраните новое содержимое в созданном файле. Имена файла источника и файла назначения должны быть запрошены у пользователя. Удостоверьтесь в том, что программа корректно обрабатывает возможные ошибки при работе с обоими файлами.

Приложение 2
к рабочей программе дисциплины (модуля)
Б1.О.20 Основы Python

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 15.03.06 – Мехатроника и робототехника

Профиль подготовки: Робототехника и искусственный интеллект

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2024

Основная литература:

1. Лучано, Р. Python. К вершинам мастерства : практическое пособие / Р. Лучано ; пер. с англ. А. А. Слинкин. — Москва : ДМК Пресс, 2016. — 768 с. — ISBN 978-5-97060-384-0. — Текст : электронный. — URL: https://znanium.com/catalog/document?id=341183 (дата обращения: 11.09.2023). — Режим доступа: по подписке.

2. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2023. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: https://znanium.com/catalog/product/1915716 (дата обращения: 11.09.2023). – Режим доступа: по подписке.

3. Златопольский, Д. М. Основы программирования на языке Python : учебник / Д. М. Златопольский. - 2-е изд. - Москва : ДМК Пресс, 2018. - 396 с. - ISBN 978-5-97060-641-4. - Текст : электронный. - URL: https://znanium.com/catalog/product/2012512 (дата обращения: 11.09.2023). – Режим доступа: по подписке.

Дополнительная литература:

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: https://znanium.com/catalog/product/1913856 (дата обращения: 11.09.2023). – Режим доступа: по подписке.

2. Титов, А. Н. Визуализация данных в Python. Работа с библиотекой Matplotlib : учебно-методическое пособие / А. Н. Титов, Р. Ф. Тазиева ; Минобрна-уки России, Казан. нац. исслед. технол. ун-т. - Казань : Изд-во КНИТУ, 2022. - 92 с. - ISBN 978-5-7882-3176-1. - Текст : электронный. - URL: https://znanium.com/catalog/product/2069267 (дата обращения: 11.09.2023). – Режим доступа: по подписке.

3. Саммерфильд, М. Python на практике. Создание качественных программ с использованием параллелизма, библиотек и паттернов : практическое пособие / М. Саммерфильд ; пер. с англ. А. А. Слинкина. - 2-е изд. - Москва : ДМК Пресс, 2023. - 340 с. - ISBN 978-5-89818-322-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/2102609 (дата обращения: 11.09.2023). – Режим доступа: по подписке.

Приложение 3
к рабочей программе дисциплины (модуля)
Б1.О.20 Основы Python

Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: 15.03.06 – Мехатроника и робототехника

Профиль подготовки: Робототехника и искусственный интеллект

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

Год начала обучения по образовательной программе: 2024

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows 7 Профессиональная или Microsoft Windows 10 Профессиональная

Пакет офисного программного обеспечения Microsoft Office 365 или Microsoft Office Professional plus 2010

Браузер Mozilla Firefox

Браузер Google Chrome

Adobe Reader XI или Adobe Acrobat Reader DC

PyCharm Community Edition

Kaspersky Endpoint Security для Windows