

Негосударственное образовательное частное учреждение высшего образования
«Московский финансово-промышленный университет «Синергия»

**Дополнительная общеобразовательная программа
«Разработка десктопного приложения на Python»**

Москва, 2022

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН	7
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК	8
РАБОЧАЯ ПРОГРАММА	9
ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ	13
УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ	14
ФОРМЫ АТТЕСТАЦИИ	16
ОЦЕНОЧНЫЕ МАТЕРИАЛЫ, ВКЛЮЧАЯ ПРИМЕРЫ КОНТРОЛЬНЫХ ЗАДАНИЙ	17
ПЕРЕЧЕНЬ ИСТОЧНИКОВ ИНФОРМАЦИОННОГО СОПРОВОЖДЕНИЯ	21
КОМПЕТЕНЦИИ, ФОРМИРУЕМЫЕ В ПРОЦЕССЕ ОБУЧЕНИЯ	22

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Аннотация

Программа «Разработка десктопного приложения на Python» предназначена для учащихся 8-11 классов. Этот курс будет полезен тем, кто интересуется областью разработки и поддержки десктопных приложений. Используя язык Python можно создавать различное программное обеспечение, в том числе разрабатывать клиенты для мессенджеров, организовывать соединения с базами данных (при помощи языка запросов SQL).

Курс не требует специальной дополнительной подготовки, достаточно знаний предметной области информатики на базовом уровне. При необходимости для эффективного прохождения курса можно повторить некоторые темы из школьного курса информатики «Алгоритмы», «Базы данных», «Устройство сети Интернет». В курсе данные темы разбираются более подробно.

Обучение на курсе будет проходить в формате вебинаров, поэтому для успешного прохождения курса потребуется наличие компьютера и высокоскоростного доступа в интернет. После окончания занятия запись вебинара будет доступна на платформе.

В ходе изучения учебных материалов программы слушатели освою навыки разработки программного кода используя синтаксис Python, получат опыт работы с базами данных и их внутренней архитектурой, изучат систему контроля версий Git и создание полноценных проектов.

Формирование знаний и навык на высоком уровне достигается за счет того, что программа носит практико-ориентированный характер. Все знания, полученные во время онлайн-лекций, отрабатываются вовремя онлайн практических занятий. Система заданий во время практики организована таким образом, что сначала обучающимся предлагаются задания на формирование и отработку конкретных умений через решение алгоритмических задач. После чего учащимся предлагается выполнение учебного проекта, например, создание оконного приложения, калькулятора, игры, справочника, клиента для мессенджера, создания схемы базы данных собственного проекта и реализации работы приложения с базой данных.

Актуальность

Язык программирования Python, в настоящее время, является наиболее востребованным во всех сферах ИТ: от разработки до машинного обучения. По данным рейтинга TIOBE в 2022 язык Python является самым популярным языком программирования. Он используется во многих проектах и в различных качествах: как основной язык программирования или для создания расширений и интеграции приложений. На Python реализовано большое количество проектов, также он активно используется для создания прототипов будущих программ. Python активно используется для разработки программного обеспечения таких компаний, как Google, Dropbox, Reddit. Многие компании используют Python для тестирования аппаратного обеспечения, среди этих компаний значатся Intel, Cisco, Hewlett-Packard и IBM. Industrial Light & Magic и Pixar используют его при создании анимационных фильмов.

Обучение на данной программе позволит сформировать компетенций в области решения практических задач разработки приложений с использованием языка Python, применения шаблонов проектирования, работы с базами данных, применения объектно-ориентированного. Программа позволит с помощью формирования практических навыков в разработке приложений повысить интерес школьников к программированию, что повлияет на количество поступающих абитуриентов в образовательные организации среднего профессионального и высшего образования по ИТ-направлениям.

Цель программы

Формирование знаний, умений и навыков достаточных для разработки приложений с использованием языка программирования Python.

Задачи программы

Программа направлена на реализацию следующих задач обучения, развития и воспитания.

Задачи обучения

- Формировать и развивать компетентность в области разработки приложений.
- Формировать представление о процессах разработки.
- Формировать представление об архитектуре компьютеров и сети.
- Формировать представление о правила безопасной работы с информацией.
- Формировать ИКТ-компетенции.
- Формировать представление об уровне технического прогресса.

Задачи развития

- Формировать умение планировать и регулировать свою деятельность.
- Способствовать формированию самонаблюдения и самооценки в процессе познавательной, творческой деятельности.
- Формировать и развивать компетентность в области работы с информацией.
- Способствовать формированию навыков сотрудничества со взрослыми и сверстниками в разных социальных ситуациях, умения избегать конфликтов. Способствовать овладению навыками адаптации в динамично развивающемся мире.

Задачи воспитания

- Способствовать формированию нравственных норм и ценностей в поведении и сознании.
- Способствовать принятию и освоению социальной роли обучающегося, развитию мотивов учебной деятельности и формированию личностного смысла образования.
- Способствовать развитию самостоятельности и личной ответственности за свои поступки, в том числе в информационной деятельности, на основе представлений о нравственных нормах, социальной справедливости и свободе. Развивать рефлексивную деятельность учащихся.
- Способствовать развитию эстетических чувств, доброжелательности и эмоционально-нравственной отзывчивости, понимания чувств других людей и сопереживания им.
- Способствовать формированию установки на безопасный, здоровый образ жизни, наличие мотивации к творческому труду, работе на результат, бережному отношению к материальным и духовным ценностям.

Планируемые результаты обучения

Сформировать у обучающихся школ базовые знания и умения в области решения задач разработки приложений.

По окончании обучения по программе обучающийся будет знать:

- Понятие алгоритма и алгоритмического языка программирования.
- Назначение Python и его сфера применения.
- Основы синтаксиса языка программирования Python.
- Основы программирования на Python:
- Базовые операторы, переменные и их типы, константы, массивы, управляющие конструкции, условные операторы, циклы, функции, принципы создание функций,

- аргументы функции, синхронные и асинхронные функции, область видимости переменных, стандартные библиотеки языка программирования Python.
- Структуру и элементы приложений. Компоненты, из которых состоит приложение. Основные принципы UX/UI-дизайна.
 - Реляционные базы данных. СУБД. Модели данных, основные операции и ограничения. Технология установки и настройки сервера баз данных. Язык запросов SQL. Настройка окружения и PostgreSQL. Транзакции, понятие данных и метаданных. Особенности системы управления базами данных DBeaver.
 - Систему контроля версий.
 - Возможности используемой системы контроля версий и вспомогательных инструментальных программных средств для обработки исходного текста программного кода.
 - Основные принципы отладки и тестирования программных продуктов. Типы тестирования.
 - Методы и приемы отладки программного кода. Типы и форматы сообщений об ошибках, предупреждениях.
 - Принципы работы современных компиляторов, отладчиков и оптимизаторов программного кода.

По окончании обучения по программе обучающийся будет уметь:

- Устанавливать и настраивать среду разработки Python.
- Применять язык программирования Python для написания программного кода для решения учебных и практических задач.
- Создавать структуру кода, размещающую элементы приложения.
- Использовать возможности языка программирования Python для разработки десктопного приложения.
- Проектировать и создавать базы данных.
- Создавать сценарии исполнения запросов на создание, изменение, выборку и удаление данных на языке SQL.
- Работать с системой контроля версий.
- Использовать инструментальные средства для отладки и тестирования.
- Выполнять отладку и тестирование программы на уровне модуля.
- Выявлять ошибки в программном коде.
- Интерпретировать сообщения об ошибках, предупреждениях.

Категория обучающихся (возраст) по программе

Учащиеся, осваивающие образовательные программы основного общего и среднего общего образования (с 8 по 11 классы)

Форма обучения:

Очная форма обучения без применения дистанционных образовательных технологий.

Срок реализации программы:

2 года

Трудоемкость программы:

150 академических часов в рамках двухлетнего курса обучения, с включением 4 модулей трудоемкостью не менее 36 академических часов.

Ссылка на официальную страницу курса:

http://cospaces.synergyitschool.com/python_offline

УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН

№ п/п	Наименование модулей/тем программы	Всего, час	Виды учебных занятий			Формы контроля
			Л е к ц и и	П р а к т и ч е с к и е з а н я т и я	С а м о с т о я т е л ь н а я р а б о т а	
1	Тема 1.1. Архитектура компьютера и сетей	8	2	2	4	
2	Тема 1.2. Введение в разработку. Знакомство с Python	28	4	10	14	
3	Тема 2.1 Введение в работу с функциями на языке Python	16	3	7	6	
4	Тема 2.2 Работа с файлами	6	2	2	2	
5	Тема 2.3 Основы работы с базами данных	14	3	5	6	
6	Тема 3.1 Работа с Git. Версионность	9	2	3	4	
7	Тема 3.2 Введение в объектно-ориентированное программирование	22	6	8	8	
8	Тема 3.3 Интерфейс приложения	5	1	2	2	
9	Тема 4.1 Архитектура приложений и шаблоны проектирования	6	1	2	3	
10	Тема 4.2 Разработка полноценного приложения	25	4	9	12	
11	Тема 4.3 Тестирование приложений	5	1	2	2	
13	Промежуточная аттестация	4				Тестирование
14	Итоговая аттестация	2				Защита проекта
15	Итого	150	29	52	63	

РАБОЧАЯ ПРОГРАММА

Модуль 1. Введение в разработку. Знакомство с Python

Модуль знакомит с основами и особенностями языка программирования Python, его назначения и его сфера применения, архитектурой компьютера и сетей. Во время обучения, обучающиеся подготовят свое рабочее место и приступят к изучению синтаксис и базовых конструкций языка программирования Python. Научатся использовать язык программирования Python для решения учебных и практических задач.

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы
Тема 1.1. Архитектура компьютера и сетей	Архитектура компьютера Принцип работы процессора, устройства памяти. Операционные системы. Различия и сходства в языках программирования разных уровней. Архитектура сетей.	Решение учебных и практических задач. Сборка идеального компьютера «на бумаге». Разработка алгоритма для работы одной единицы техники	Повторение материалов по теме. Сборка идеального компьютера «на бумаге». Разработка алгоритма для работы одной единицы техники
Тема 1.2. Введение в разработку. Знакомство с Python	Понятие алгоритма и алгоритмического языка программирования. Синтаксис языка программирования Python. Базовые операторы. Переменные и их типы. Константы. Массивы. Управляющие конструкции. Условные операторы. Операторы инкремента, декремента. Циклы. Работа с операторами Switch, Break, Continue. Отладка кода.	Решение алгоритмических задач с помощью блок-схем и описание объектов. Решение практических задач с помощью языка программирования Python. Работа с условиями, циклами, массивами, операторами. Создание веб-приложения.	Повторение материалов по теме. Решение алгоритмических задач с помощью блок-схем и описание объектов. Решение практических задач с помощью языка программирования Python.
Промежуточная аттестация			

Модуль 2. Введение в работу с функциями на языке Python. Работа с данными.

Модуль знакомит с функциями, научатся создавать собственные функции и использовать библиотеки. Научатся работать с файлами для чтения и записи данных. Познакомятся с базами данных и языком запросов SQL. По окончании модулю спроектируют базу данных собственного сайта, смогут создавать сценарии исполнения запросов на создание, изменение, выборку и удаление данных, получать результаты запроса и вывод его на страницу сайта.

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы
Тема 2.1 Введение в работу с функциями на языке Python	Функции. Создание функции. Аргументы функции и аргументы по умолчанию. Область	Создание собственных функций. Подключение библиотек. Оптимизация кода. Использование	Повторение материалов по теме. Использование функций для решения практических задач. Использование

	видимости переменных. Статические переменные. Локальные и глобальные переменные. Обзор встроенных функций. Библиотеки. Полезные функции. Синхронные и асинхронные функции. Сортировка.	функций для решения практических задач.	продвинутых возможности функций для написания кода.
Тема 2.2 Работа с файлами	Функции для работы с файлами. Режимы работы. Чтение и запись. Функции для работы с директориями.	Работа с файлами. Организация чтения и записи. Решение учебных и практических задач	Повторение материалов по теме. Работа с файлами. Организация чтения и записи.
Тема 2.3 Основы работы с базами данных	Общие сведения о системах управления базами данных (СУБД). PostgreSQL. Программа PythonMyAdmin для работы с СУБД MySQL. Язык запросов SQL. Настройка окружения и PostgreSQL. Транзакции, понятие данных и метаданных. Создание базы данных и таблицы. Создание запросов на языке SQL. Соединение с сервером баз данных MySQL и выбор базы данных. Создание сценариев исполнения запросов на создание, изменение, выборку и удаление данных. Получение результата запроса и вывод его на страницу сайта.	Создание простых таблиц, заполнение их данными и обновление. Разработка кода на языке базы данных PostgreSQL — PL/SQL. Отправка данных на сервер. Создание схемы базы данных проекта.	Повторение материалов по теме. Работа с запросами на языке SQL.
Промежуточная аттестация			

Модуль 3. Основы объектно-ориентированного программирования. Работа с системой контроля версий.

Модуль знакомит с возможностями принципов объектно-ориентированного программирования для разработки приложений, учит работать с версионностью и применять системы контроля версий для обработки исходного текста программного кода, разрабатывать элементарный интерфейс приложений.

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы
Тема 3.1 Работа с Git. Версионность	Система контроля версий. Интерфейс Git. Возможности используемой системы	Применение системы контроля версий для обработки исходного текста программного	Повторение материалов по теме.

	контроля версий и вспомогательных инструментальных программных средств для обработки исходного текста программного кода.	кода. Регистрация изменений исходного текста программного кода в системе контроля версий	
Тема 3.2 Введение в объектно-ориентированное программирование	Классы и объекты. Конструктор класса. Магические методы. Наследование классов. Инкапсуляция. Полиморфизм. Static, self, и final. Абстрактные классы. Анонимные классы. Перегрузка. Пространство имен. Классы DateTime. Замыкания. Генераторы. Итераторы.	Решение практических задач с использованием классов, объектов. Использование объектов для реализации программной логики. Защита состояния объектов от внешнего воздействия с помощью модификаторов доступа. Использование исключений.	Повторение материалов по теме. Решение практических задач.
Тема 3.3 Интерфейс приложения	Компоненты, из которых состоит приложение. Введение в UX/UI-дизайн. Генерация идей приложения.	Разработка прототипа для одной из выдвинутых идей приложения.	Повторение материалов по теме. Разработка прототипа для одной из выдвинутых идей приложения.
Промежуточная аттестация			

Модуль 4. Архитектура приложений. Тестирование

Модуль знакомит с шаблонами проектирования программного обеспечения, конфигурацией, стандартами кодирования и интерфейсами. Во время обучения, обучающиеся научатся изменять базовую конфигурацию без создания опасных ситуаций, разрабатывать полноценное приложение и выполнять его тестирование.

Название темы	Содержание		
	Лекций	Практических занятий	Самостоятельной работы
Тема 4.1 Архитектура приложений и шаблоны проектирования	Понятие MVC, Front Controller и настройка веб-сервера. Шаблоны проектирования программного обеспечения. Конфигурация.	Применение шаблонов проектирования при разработке веб-приложения. Реализация шаблона декоратор. Реализация произвольного шаблона. Переопределение базовой конфигурации без создания опасных ситуаций	Повторение материалов по теме. Реализация шаблонов проектирования.
Тема 4.2 Разработка полноценного приложения	Этапы разработки приложения. Разработка полноценного приложения. Повторение.	Разработка интерфейса и исходного кода приложения.	Повторение материалов по теме. Разработка приложения.
Тема 4.3 Тестирование приложений	Тестирование приложения: моки и стабы. Непрерывная доставка Phing. JMeter	Тестирование ошибок. Подготовка данных для тестов. Тестирование кода, взаимодействующего с	Повторение материалов по теме. Тестирование приложения.

	нагрузочные испытания. Автотесты	файлами. Работа с тестами.	
Промежуточная аттестация			
Итоговая аттестация			

ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

Кадровое обеспечение

Фамилия	Имя	Отчество	Уровень образования	Ученая степень	Ученое звание	Наименование основного места работы	Должность	Портфолио
Сосковец	Лариса	Александровна	высшее педагогическое	нет	нет	Университет Синергия	Старший преподаватель кафедры ГиЕД	https://docs.google.com/document/d/1az089wbADo1DxX156iG8waqmYUIWn40z/edit?usp=sharing&oid=103347011653736542142&rtpof=true&sd=true
Гедыгушева	Татьяна	Хаучьяновна	высшее	Кандидат педагогических наук		Университет Синергия	Заведующая кафедрой математических и естественно-научных дисциплин	https://disk.yandex.ru/d/RKE0HXb1mcg6gg

Адреса и координаты

№ п/п	Название адреса	Адрес	Код адреса	Долгота	Широта
1	Омский филиал Университета «Синергия»	г.Омск, проспект Карла Маркса, 18, корп. 10	52000000000	54.966280	73.382543
2	Карачаево- Черкесский филиал Университета «Синергия»	г.Черкесск просп. Ленина, 83	91000000000	44.217198	42.047749
3	Колледж экономики и права	г. Щелково, ул. Пионерская, д.19	46000000000	55.922706	37.969092

Методическое обеспечение. Методы, формы и технологии, применяемые при реализации программы

Обучение на курсе организовано в групповой и индивидуально-групповой форме.
Основная форма обучения: онлайн-лекции и практики, в том числе с демонстрацией экрана.

Курс разработан с использованием следующих образовательных технологий:

- технология развивающего обучения,
- технология проблемного обучения,
- проектная технология,
- кейс-технология,
- обучение в сотрудничестве,
- здоровьесберегающие технологии.

При разработке материалов курса учитываются различные типы перцептивная модальности, поэтому используются словесные, наглядные, практические методы.

Ко всем необходимым материалам доступ обучающихся организуется через платформу обучения: презентациям, заданиям, записям прошедших вебинаров.

Все знания, полученные во время онлайн-лекций, отрабатывается вовремя онлайн практических занятий. Система заданий во время практики организована таким образом, что сначала обучающимся предлагаются задания на формирование и отработку конкретных умений через решение алгоритмических задач. После чего учащимся предлагается выполнение учебного проекта, например, создание сайта-визитки, новостной страницы, калькулятора, игры, создания формы обратной связи, формы регистрации, справочника, создания схемы базы данных собственного проекта и реализации работы веб-приложения с базой данных.

Материально-техническое обеспечение

- Персональный компьютер, который соответствует минимальным требованиям:
Для ПК на Windows: процессор i-серии (i3, i5, i7) или аналогичный от AMD (от 4 ядер), от 4G B RAM.
Для ПК на Mac OS: конфигурация Intel i7-2720QM/RAM 4 GB/HDD 750 GB., ОС Windows 10 (версии Home достаточно) либо Mac OS X;
- Высокоскоростной стабильный доступ в интернет.

УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

В процессе обучения по образовательной программе система заданий организована таким образом, что сначала обучающимся предлагаются задания на формирование и отработку конкретных умений через решение алгоритмических задач. После чего учащимся предлагается выполнение учебного проекта. Ниже представлены примеры некоторых из них.

Пример задания по Тема 1.2. Введение в разработку. Знакомство с Python

Задача: установка среды разработки и необходимого ПО на ПК

Процесс выполнения: Скачать дистрибутив с официального сайта Python 3.2, установить, следуя подсказкам преподавателя в необходимую директорию. Далее с официального сайта скачиваем дистрибутив Visual Studio Code, устанавливаем, согласно инструкциям преподавателя. Далее открываем VS Code и настраиваем редактор, согласно инструкциям.

Результат, который должен получить студент: иметь подготовленную среду для комфортной работы и изучения языка программирования.

Задача: Обучение студента работе с базовыми сущностями в языке Python

Процесс выполнения:

- 1) Открыть VS Code
- 2) В первой строке написать команду `print(5)`
- 3) Запустить выполнение кода, на выводе консоль выдаст цифру `5`.
- 4) Вывести строку, обернутую в кавычки с помощью: `print("Hello world!")`
- 5) Вывести в консоль комбинацию строк с помощью команды: `print("Hello", "world" + "!",)`
- 6) Ввести последовательность команд `print`, где одна из них будет пустой
- 7) Ввести команду `input`
- 8) В консоли ввести `5`
- 9) Теперь необходимо объединить команды `print` и `input`, ввести: `print(input())`
- 10) В консоли ввести любое число или строку
- 11) Получить на выводе введенное число или строку
- 12) Чтобы научиться работать с переменными студенту необходимо вызвать переменную и присвоить ей любое числовое или строковое значение.
- 13) Комбинировать с командой `print` и `input`

Результат, который должен получить студент: уметь работать с базовыми сущностями и выполнять базовые операции на языке Python

Пример проекта по Тема 2.1 Введение в работу с функциями на языке Python

Сделать игру, задача которой заключается в следующем:

У нас есть прямоугольное поле, на нем расположены деревья и реки.

Деревья периодически будут вырастать и периодически будут загораться.

Наша задача - вовремя их тушить с помощью вертолета.

Вертолет будет перемещаться по карте. Когда вертолет пролетает над клеткой с водой, он берет ее себе в резервуар и может потушить с помощью этой воды дерево, если вовремя его не потушить - пожар распространиться дальше, а дерево сгорит. Вы будете терять очки если дерево сгорело, и получать очки если его успешно потушили.

Чтобы усложнить игру, нужно добавить погодные условия:

- облака, которые будут мешать видеть карту
- грозы в этих облаках, при попадании в которые вертолет будет терять жизни

Чтобы количество жизней можно было пополнять, нужен госпиталь в который можно залететь и купить за очки единицу здоровья.

Также можно добавить магазин улучшений, чтобы можно было улучшить вертолет и он мог брать сразу две ячейки воды и потушить два пожара.

Работа должна выполняться в Visual Studio CODE, с использованием эמודзи.

Пример заданий по Тема 4.2 Разработка полноценного приложения

План выполнения:

1. Откройте Visual Studio Code и перейдите в путь нашего проекта.

Вторым шагом давайте включим нашу виртуальную среду разработки с помощью команды `venv/Scripts/activate.bat` (`source venv/bin/activate` если у Вас macOS, !!! ВАЖНО!!! так же возможно, что Вам будет достаточно использовать команду `venv/scripts/activate` и на windows, если с файлом `activate.bat` вызываются проблемы). Не забывайте проверить стоит ли у Вас нужный интерпретатор.

2. Разберемся с Permissions.

- Для начала изучите материал который рассказывает спикер.
- Перейдем на админскую страницу и добавим новую группу с правами на `service` и названием `team_group`
- Теперь перейдем во вкладку пользователей и добавим пользователя в новую группу (если пользователя, кроме суперюзера нет, создайте нового) в моем случае пользователя зовут Artem. И не забудьте сохранить.
- Перейдем в `views.py` и пропишем/изменим код

- Ссылка на сайт с документацией по Django: <https://docs.djangoproject.com/en/4.0/>
- Перейдем на страницу админа, и пользователю Artem добавим галочку рядом с Staff Status. И не забудьте сохранить.
- Перейдем на страницу index.html и пропишем условия
- Теперь при переходе на страницу форума, у пользователя Artem будет доступ к созданию постов.
- Так мы научились создавать проверку на Permissions.

3. Так как этот урок более теоретический, то советую дальше просто повторять и изучать действия спикера.

Пример заданий по Тема 4.2 Разработка полноценного приложения

Цель задания: Изучить тему тестирования.

План выполнения:

1. Откройте Visual Studio Code и перейдите в путь нашего проекта.

Вторым шагом давайте включим нашу виртуальную среду разработки с помощью команды `venv/Scripts/activate.bat` (source `venv/bin/activate` если у Вас macOS, !!! ВАЖНО!!! так же возможно, что Вам будет достаточно использовать команду `venv/scripts/activate` и на windows, если с файлом `activate.bat` вызываются проблемы). Не забывайте проверить стоит ли у Вас нужный интерпретатор.

2. Выполним первый тест.

В папке `api` создадим новую папку с названием `test`, в ней создадим файл с названием `__init__.py` и перенесем в эту папку файл `tests.py` и переименуем его в `test_utils.py`

Теперь перейдем в `test_utils.py` и пропишем код

Теперь запустим в терминале команду `python manage.py test .` и получим такой результат

3. Напишем тест для API.

Создадим новый файл

Перейдем в `settings.py` и прокомментируем данную часть кода

Теперь перейдем в `test_api.py` и пропишем первый тест

Результат в терминале при запуске команды `python manage.py test api.tests.test_api`

Теперь перейдем в `test_api.py` и пропишем второй тест

Результат в терминале при запуске команды `python manage.py test api.tests.test_api`

4. Напишем тест для сериалайзеров.

Создадим новый файл для тестов с названием `test_serializer.py` и пропишем код

Результат в терминале при выполнении команды `python manage.py test api.tests.test_serializer`

5. Узнаем про Coverage.

Зеркало сайта с урока: <https://evileg.com/ru/post/457/>

Для начала установим Coverage через терминал командой `pip install coverage`.

Затем повторите все действия за спикером, для ознакомления с данным инструментом.

ФОРМЫ АТТЕСТАЦИИ

Аттестация проводится в двух формах: промежуточная и итоговая.

Промежуточная аттестация. Данный вид аттестации проводится по итогам освоения каждого модуля.

Форма контроля - тестирование.

Показатели и критерии оценивания:

Оценка теста производится автоматически. Тест считается пройденным, если даны правильные ответы не менее, чем на 50% тестовых заданий.

Шкала оценивания

Нижнее значение	0%
Верхнее значение	100%
Минимальный проходной балл для успешной сдачи	50%

Итоговая аттестация. Данный вид аттестации проводится по итогам освоения всей Образовательной программы, включая освоение ее последнего модуля.

Форма контроля – защита проектов.

Показатели и критерии оценивания:

Проекты проверяются в соответствии с заданиями, описанными в материалах.

За каждое выполненное условие ставится от 0 до 2 баллов согласно критериям оценивания.

Критерии оценивания:

0 — условие не выполнено

1 — условие выполнено, частично

2 — условие выполнено, полностью

Шкала оценивания

Нижнее значение	0
Верхнее значение	10
Минимальный проходной балл для успешной сдачи	5

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ, ВКЛЮЧАЯ ПРИМЕРЫ КОНТРОЛЬНЫХ ЗАДАНИЙ

Для каждого вида аттестации предусмотрены свои оценочные материалы. Ниже представлены примеры некоторых из них.

Промежуточная аттестация по модулю 1

1. Как выполняются команды кода в Python?
а. В зависимости от того на каком расстоянии от левого края среды разработки они находятся. Самые близкие имеют приоритет. Затем выполняются остальные.

б. Код выполняется последовательно построчно за редкими исключениями.

в. Каждая команда имеет свой приоритет и выполняется в соответствии с установленными стандартами.

2. Какая команда позволяет ввести информацию из консоли?

а. input()

б. while()

в. print()

г. println()

3. В какой последовательности производятся арифметические операции в Python?

а. Строго в соответствии с написанием. Приоритет не соблюдается.

б. Так как положено по математическим законам.

в. Математические выражения обязательно нужно заключать в скобки, чтобы программа понимала как производить вычисление. В противном случае будет выведена синтаксическая ошибка.

4. Какие есть операторы сравнения в Python?

а. > больше, < меньше, = равно.

б. (a) + ==> больше либо равно, ==< меньше либо равно, != не равно.

в (a) + >= больше либо равно, <= меньше либо равно, != не равно.

5. Как можно изменить в строке орфографическую ошибку?

а. Найти индекс соответствующего символа и поменять букву, используя специальный метод.

б. Существует метод, который путём перебора символов может отбросить ненужный и вставить корректный. Для этого используют цикл for.

в. Строка - это неизменяемый тип данных. Для того чтобы исправить ошибку, нужно перезаписать строку, то есть создать новый объект.

Промежуточная аттестация по модулю 2

1. Назовите популярные реляционные и нереляционные базы данных.

- а) Реляционные - MySQL, NoSQL, PostgreSQL. Нереляционные - SQLite
- б) Реляционные - MySQL, SQLite, PostgreSQL. Нереляционные - NoSQL
- в) Реляционные - MySQL, NoSQL. Нереляционные - SQLite, PostgreSQL

2. В чём отличие РСУБД от НРСУБД?

- а) В отличие от реляционных, нереляционные СУБД не поддерживают SQL-запросы.
- б) Нереляционные СУБД, в отличие от реляционных, могут хранить информацию в табличном виде.
- в) Нереляционные БД не поддерживают изображения и видео форматы.

3. Где лучше использовать РСУБД?

- а) Если есть чёткая структура (например таблица), которую нужно будет изменять, форматировать и совершать с ней другие действия.
- б) Чёткая структура не прослеживается. Не прослеживаются структурные взаимосвязи между объектами.
- в) Удобно использовать в социальных сетях, где часто меняются пользователи, подписки и т.д.

4. Как установить MySQL на компьютер?

- а) Эта система по умолчанию установлена в некоторые браузеры. Достаточно просто вызвать её в командной строке, либо в терминале.
- б) Скачивание программы MySQL Community доступно только после регистрации и оформлении подписки. После этого на почту приходят установочные файлы, привязанные только к одному конкретному компьютеру.
- в) На официальной странице mysql.com в разделе download выбрать версию community. Далее, следуя подсказкам, устанавливаем в обычном режиме.

Промежуточная аттестация по модулю 3

1. Закончите предложение. Класс отличается от объекта тем, что...
 - А. класс - это переменная, объекта - это массив
 - В. класс - это экземпляр объекта
 - С. объект - это экземпляр класса
 - Д. ничем

2. Что означает final class? Выберите верный ответ.
 - А. он не является абстрактным
 - В. не содержит констант
 - С. он не может быть унаследован
 - Д. может быть создан лишь 1 его экземпляр

3. Что произойдет, если добавить в класс приватный конструктор и создать экземпляр класса?
 - А. fatal error
 - В. notice
 - С. ничего

D. warning

4. Что означает protected? Выберите верный ответ.

- A. свойства и методы доступны в экземплярах самого класса и в экземплярах наследников
- B. то же самое что private
- C. обратное к public
- D. свойства и методы доступны только в экземплярах наследника

Промежуточная аттестация по модулю 4

1. Выберите верный ответ. Какой шаблон позволяет складывать инициализированные объекты и доставать их оттуда при необходимости?
 - A. Реестр
 - B. Пул объектов
 - C. MVC
 - D. Singleton

2. Какие различают виды тестирования по степени подготовки? Выберите верный ответ.
 - A. Тестирование по документации
 - B. Альфа-тестирование
 - C. Компонентное тестирование
 - D. Эксплоринг

3. Закончите фразу. Регрессионные ошибки - это ошибки, которые возникают когда...
 - A. Функциональные возможности программного обеспечения, которые ранее работали, перестали работать
 - B. Новый функционал программного обеспечения не работает так, как планировалось
 - C. Старый функционал программного обеспечения не работает так, как планировалось

Итоговая аттестация

Для прохождения итоговой аттестации необходимо предоставить самостоятельно разработанное приложение. На экспертную оценку необходимо направить презентацию, содержащую информацию о вашем проекте и его компонентах. Проект должен соответствовать следующим требованиям:

- 1) Проект должен быть рабочим, без зависаний, лагов, соответствовать его описанию
 - 2) Должно присутствовать его текстовое описание (приложен файл tutorial.txt)
 - 3) Проект должен быть размещён на Github
 - 4) Код должен быть написан в строгом соответствии с принятыми стандартами соответствующего языка программирования, то есть, соблюден синтаксис, отступы, переменные должны иметь названия, логически отражающие контекст программы.
 - 5) Код должен быть рационален - то есть написан максимально оптимально с точки зрения объема, выполняться настолько это возможно быстро, не содержать бесконечных циклов.
- Внутри кода обязательно должны присутствовать комментарии, объясняющие ключевые моменты
 - Если проект содержит несколько файлов, то их названия должны иметь соответствующие расширения и их имена должны логически соответствовать их содержанию (например,

если в файле содержится код на языке javascript и он отвечает за выполнение математических вычислений, то название должно быть похожим на math.js и так далее).

ПЕРЕЧЕНЬ ИСТОЧНИКОВ ИНФОРМАЦИОННОГО СОПРОВОЖДЕНИЯ

- Эрик Мэтиз Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – СПб.: Питер, 2017. – 496 с: ил – (Серия “Библиотека программиста”)
- Марк Лутц Изучаем Python, том1,5-еизд.:Пер.сангл.—
СПб.:ООО“Диалектика”,2019.—832с.:ил.—Парад,тит.англ
- Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель. SQL. Полное руководство. - Диалектика, 2019. – 960 с.
- Дэниэл Кляйн, Брэнд Хант, Кевин Кляйн, SQL. Справочник. – 3-е изд. – СПб: Символ-Плюс, 2010. – 656 с.

КОМПЕТЕНЦИИ, ФОРМИРУЕМЫЕ В ПРОЦЕССЕ ОБУЧЕНИЯ

Наименование компетенции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующие компетенции
Способен создавать программы (кодировать) на языке программирования Python	общефессиональные	<p>Понятие алгоритма и алгоритмического языка программирования.</p> <p>Назначение Python и его сфера применения.</p> <p>Основы синтаксиса языка программирования Python.</p> <p>Особенности программирования на языке Python.</p> <p>Базовые операторы.</p> <p>Переменные и их типы.</p> <p>Константы.</p> <p>Массивы.</p> <p>Управляющие конструкции.</p> <p>Условные операторы.</p> <p>Операторы инкремента, декремента.</p> <p>Циклы.</p> <p>Работа с операторами Switch, Break, Continue</p> <p>Функции.</p> <p>Создание функции.</p> <p>Аргументы функции и аргументы функции по умолчанию.</p> <p>Область видимости переменных.</p>	<p>Установка и настройка среды разработки Python.</p> <p>Применять язык программирования Python для написания программного кода для решения учебных и практических задач.</p> <p>Разрабатывать программный код приложений с использованием языка программирования Python.</p>	<p>- Средствами разработки: Visual Studio Code, PyCharm</p> <p>- Командной строкой (cmd)</p>

		<p>Статические переменные. Локальные и глобальные переменные. Стандартные библиотеки языка программирования Python Синхронные и асинхронные функции.</p>		
Способен работать с базами данных с помощью языка программирования SQL	общепрофессиональные	<p>Реляционные базы данных. СУБД. Модели данных, основные операции и ограничения. Технология установки и настройки сервера баз данных. Язык запросов SQL Настройка окружения и PostgreSQL. Транзакции, понятие данных и метаданных. Особенности выбранной среды программирования DBeaver и системы управления базами данных.</p>	<p>Проектировать и создавать базы данных. Создавать сценарии исполнения запросов на создание, изменение, выборку и удаление данных на языке SQL. Получать результаты запроса и выводить его в приложение. Использование выбранной среды программирования и средств системы управления базами данных для организации функционирования веб-приложения.</p>	<p>- Средствами разработки: Visual Studio Code, PyCharm - Менеджер базы данных: DBeaver, MySQL</p>
Способен осуществлять проверку и отладку программного кода	общепрофессиональные	<p>Основные принципы отладки и тестирования программных продуктов. Типы тестирования. Методы и приемы отладки программного кода. Типы и форматы сообщений об ошибках, предупреждения. Принципы работы современных компиляторов, отладчиков и оптимизаторов программного кода</p>	<p>Выполнять отладку и тестирование программы на уровне модуля. Выявлять ошибки в программном коде. Интерпретировать сообщения об ошибках, предупреждения. Использовать инструментальных средств для отладки и тестирования программного продукт.</p>	<p>- Средствами разработки: Visual Studio Code, PyCharm - Командной строкой (cmd) - Инструментом нагрузочного тестирования Apache JMeter</p>

Способен работать с системой контроля версий Git	общепрофессиональные	Система контроля версий. Интерфейс Git. Возможности используемой системы контроля версий и вспомогательных инструментальных программных средств для обработки исходного текста программного кода.	Применение системы контроля версий для обработки исходного текста программного кода. Регистрация изменений исходного текста программного кода в системе контроля версий	- GitHub
--	----------------------	---	---	----------

