

Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

УТВЕРЖДАЮ
Директор филиала



Жигалова А. С.

СОГЛАСОВАНО

Начальник Управления общего и
дошкольного образования
Администрации города Норильска



Колин А. Г.

СОГЛАСОВАНО
И.д. директора МБОУ «СШ № 21»



Сасова Т.В.

СОГЛАСОВАНО
Директор МБОУ «СШ №33»

Ковалева Е. Н.



Дополнительная общеобразовательная общеразвивающая программа технической
направленности

«Python3 в web-разработке»

Срок реализации:

1 год

Возраст детей:

14–17 лет

Составитель программы:

Комаров Е. А.

г. Норильск 2022г.

СОДЕРЖАНИЕ

1. Пояснительная записка	3
1.1. Новизна ДООП	3
1.2. Актуальность ДООП	4
1.3. Педагогическая целесообразность	4
1.4. Цель ДООП	4
1.5. Возраст обучающихся, участвующих в ДООП	5
1.6. Условия вхождения в ДООП	5
1.7. Срок реализации ДООП	6
1.8. Режим занятий, формы и методы обучения	6
1.9. Ожидаемые образовательные результаты и эффекты, способы предъявления и отслеживания результатов	6
2. Учебно-тематический план	9
3. Содержание программы	12
4. Материально-техническое и информационно-методическое обеспечение	18
5. Рекомендуемая литература	19
6. Сведения о составителях ДООП	20
7. Приложение 1	21

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

На данный момент ни для кого не секрет, что в России активно строится информационное общество. Одним из критериев признания общества информационным является снижение занятости в сфере производства и увеличение в сфере услуг и информации (поскольку «сырьем» для нефизического труда является именно информация). Так, в Западной Европе, Японии и Северной Америки более 70% населения заняты в «информационной» сфере. Наша страна по этому критерию отстает.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне т. к. выделяется малое количество времени для изучения программирования. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является лишь формальное восприятие учащимися основ программирования и неумение применять полученные знания на практике.

Изучение программирования невозможно без регулярной практики написания программ на каком-либо языке. В данной программе для изучения выбран язык программирования Python. Этот выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это снижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических частях программирования, а не на заучивании синтаксиса языка. При это Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

1.1 Новизна ДООП

Предлагаемая программа нацелена на изучение синтаксиса Python и является основой для дальнейшего знакомства с этим языком программирования. Новизна данной программы заключается в практической направленности. Обучающиеся не просто слушают материал, но реализуют проекты начиная со второго занятия.

Программа ориентировано на создание необходимых условий для личностного развития обучающихся, позитивной социализации и профессионального самоопределения.

Программа имеет техническую направленность, в связи с этим рассматриваются следующие разделы для изучения:

1. Технологический. Данная программа рассматривается как средство, позволяющее развивать наиболее передовые на сегодняшний день технологии – информационные, которые включают в себя, как теоретические знания, так и практические навыки.

2. Общеразвивающий. Обучение по данной программе создает благоприятные условия для интеллектуального развития ребенка, профессионального самоопределения, развития познавательной деятельности и творческой самореализации обучающихся.

3. **Общеобразовательный.** Данная программа позволяет развить основные познавательные процессы, умение анализировать, выявлять взаимосвязи и зависимости, делать логические выводы, и опирается на такие дисциплины, как теория управления, программирование, теория информации.

Знания, полученные при прохождении данной программы, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по информатике, написании научной работы по физике, химии, биологии и другим наукам, а также являются фундаментом для дальнейшего совершенства мастерства программирования.

1.2 Актуальность ДООП

Современный уровень развития Информационных технологий предполагает освоение новых методов программирования. Программа нацелена на прививание интереса участников к информатике, а также к существующим задачам программирования.

1.3 Педагогическая целесообразность:

В связи с простой синтаксиса в сравнении с другими языками программами (ясность кода, быстрота реализации) при изучении Python есть возможность сформировать у обучающихся представление о базовых понятиях структурного программирования (данных, переменных, ветвлениях, циклах и функциях). В то же время Python является востребованным языком, он отлично подходит для концепции объектно-ориентированного программирования и активно применяется в различных областях от разработки веб-приложения до машинного обучения. Научившись программировать на языке Python, обучающиеся получают мощный и удобный в использовании инструмент для решения учебных задач и для создания собственных проектов. Вместе с тем, чистота и ясность его конструкций позволит в дальнейшем с легкостью выучить любой другой язык программирования.

1.4 Цель ДООП.

Дать Учащимся любого уровня подготовленности достаточный объём теоретических и практических знаний и области алгоритмизации и программирования, создание условий для социального, культурного и профессионального самоопределения, творческой самореализации.

Обучающие задачи:

- Формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- Приобретение навыков работы в интегрированной среде разработки на языке Python;
- Изучение конструкций языка программирования Python;
- Знакомство с понятием проекта и алгоритмом его разработки.

Воспитательные задачи:

- Развитие у учащихся инициативность и самостоятельность;
- Мотивация к созданию собственных проектов;

- Развитие стремление к получению качественного законченного результата в проектной деятельности;
- Воспитание социально значимые качества личности человека: ответственность, коммуникабельность, добросовестность, пытливость ума и критичность мышления.

Развивающие задачи:

- Развитие логического, абстрактного и образного типов мышления;
- Развитие творческих способностей.
- Формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- Приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач.

Программа представляет 3 этапа. Обучение начинается со знакомством языка Python и его синтаксисом. Затем изучаются множества и способы работы с ними. На третьем этапе знакомятся с функциями и готовят проект по итогам изученного материала.

Особенностью данной программы является привлечение детей среднего школьного возраста к изучению программирования при помощи языка Python, т. к. он обладает следующими достоинствами:

- Python – это текстовый язык программирования. Он универсален, пригоден для создания самых разнообразных программ, от текстовых процессоров до веб-браузеров;
- Python – простой и удобный язык. По сравнению со многими другими языками читать и составлять программы на Python несложно;
- В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро;
- Python используется как язык программирования крупными корпорациями, такими Google.

На первом этапе обучающиеся ознакомятся с основами синтаксиса языка Python, узнают о типах данных, переменных. Узнают о типах циклов и вложенных циклах. Научатся строить конструкции ветвлений if/else/elif. На этом этапе закладываются основы работы с Python. На втором этапе обучающиеся знакомятся с основными конструкциями языка, для обработки и хранения больших объемов данных. Узнают какие виды данных неизменяемые, как делать выборку по объектам. Научатся делать многомерные матрицы. На третьем этапе обучающиеся знакомятся и работают с функциями.

1.5 Возраст обучающихся, участвующих в ДООП

Программа «Python3. Синтаксис Python.» ориентирована на обучающихся от 14 до 17 лет. Занятия проводятся в группе из 12 человек.

1.6 Условия вхождения в ДООП

Поступающий на программу должен владеть базовыми умениями работы на компьютере (создание и удаление файлов; умение работать в простейшем текстовом и графическом редакторе; запуск, просмотр презентаций и видеороликов).

1.7 Срок реализации ДООП

Программа рассчитана на 1 учебный год. Нагрузка на обучающегося составляет 144 учебных часа.

1.8 Режим занятий, формы и методы обучения

Учебные занятия проходят по очной форме обучения. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час 45 минут) с обязательным перерывом, что определяется Санитарными правилами СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи.

При проведении занятий используются комбинированные занятия – изложение нового материала, проверка пройденного материала, закрепление полученных знаний, самостоятельная работа.

При проведении занятий используются три формы работы:

- демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах;
- фронтальная, когда обучающиеся синхронно работают под управлением педагога;
- самостоятельная, когда обучающиеся выполняют индивидуальные задания в течение части занятия;
- повторение и усвоение пройденного материала осуществляется через контрольные и проверочные работы, анализ полученных результатов;
- закрепление знаний, умений и навыков через постановку задачи и самостоятельную работу обучающегося под руководством педагога;
- применение полученных знаний и навыков через прикладную работу обучающегося, использующего на практике приобретенные компетенции.

1.9 Ожидаемые образовательные результаты и эффекты, способы предъявления и отслеживания результатов.

Учащиеся будут уметь:

- Объяснять и использовать на практике как простые, так и сложные структуры данных и конструкций для работы с ними;
- Искать и обрабатывать ошибки в коде;
- Разбирать решение задач на подзадачи;
- Писать грамотный и красивый код;
- Находить, оценивать, использовать информацию из различных источников, необходимую для решения профессиональных задач, в том числе на основе системного подхода;
- Работать с системой контроля версий Git и репозиториям GitHub;
- Базовые навыки работы с PostgreSQL;
- Грамотно работать в команде, в зависимости от целей и ситуации.

Учащиеся будут знать:

- Основные классические алгоритмы и способы их реализации;
- Синтаксис языка программирования Python;
- Основные элементы программирования: ввод/вывод информации, типы данных, операции с данными, условное выполнение, циклы;
- Основные команды работы с базами данных;
- Команды работы с репозиторием GitHub;
- Основы работы с фреймворком Django.

Сформированные навыки:

- Соблюдение требований техники безопасности;
- Работы в изучаемых программных средах;
- Навыки составления алгоритмов;
- Применения на практике основных команд и операторов изучаемых языков;
- Разработки, тестирования и отладки несложных программ;
- Навыки работы в сети Интернет для поиска информации.

Личностные результаты:

Учащиеся приобретут навыки самостоятельной организации своей деятельности; формирования основ саморазвития и самовоспитания.

У обучающихся сформируется готовность и способность к самостоятельной, творческой деятельности, к образованию, в том числе самообразованию; готовность к осознанному выбору будущей профессии.

Метапредметные результаты:

- Умение планировать; умение анализировать; алгоритмизировать.
- Коммуникативные навыки:
 - Умение договариваться с другими людьми;
 - Работать в команде;
 - Аргументировать свою позицию;
 - Развить эмоциональный интеллект – способность понимать чужие чувства и контролировать свои.
- Навыки самоорганизации и тайм-менеджмент;
- Нестандартное мышление, креативные навыки;
- Умение работать с информацией
 - Анализ информации;
 - Компьютерная грамотность;
- Стрессоустойчивость.

Опыт:

Проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; информационной деятельности в различных сферах; эффективного применения информационных образовательных ресурсов в учебной деятельности, в том

числе самообразования; эффективной организации индивидуального информационного пространства.

Механизм оценки результативности

Контроль качества образования осуществляется в форме тестов, выполнения практических и проектных работ.

В конце курса обучающиеся выполняют индивидуальный проект по созданию собственной веб-страницы

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	Основы языка программирования Python	54	23	31	
1.1	Переменные, числа, арифметика.	4	2	2	
1.2	Строки.	4	2	2	
1.3	Условные конструкции. Операции сравнения.	4	2	2	
1.4	Типы данных. Индексация и срезы строк.	2	2	2	
1.5	Списки. Кортежи.	2	1	1	
1.6	Циклы.	4	2	2	
1.7	Словари. Множества.	4	2	2	
1.8	Функции. Параметры функции.	4	2	2	
1.9	Области видимости переменных.	2	1	1	
1.10	ООП: объекты и классы. Взаимодействие между ними.	4	2	2	
1.11	ООП: наследование, инкапсуляция, полиморфизм.	4	2	2	
1.12	Работа с файлами.	2	1	1	
1.13	Работа с различными форматами данных: CSV, JSON, YAMML, XML	2	1	1	
1.14	Библиотека requests, http-запросы	2	1	1	

1.15	Проектная работа.	10	1	9	Защита проекта
2	Git-система контроля версий	6	3	3	
2.1	Введение в систему контроля версий Git.	2	1	1	
2.2	Внедрение системы контроля Git & GitHub	2	1	1	
2.3	История работы и ветки Git & GitHub	2	1	1	
3	Базы данных для python разработчиков	18	9	9	
3.1	Введение в базы данных. Типы баз данных	2	1	1	
3.2	Работа с PostgreSQL. Создание БД	2	1	1	
3.3	Select-запросы, выборки из одной таблицы	2	1	1	
3.4	Продвинутая выборка данных	4	2	2	
3.5	Работа с PostgreSQL из Python	4	2	2	
3.6	Python и БД. ORM	4	2	2	
4	Профессиональная работа с Python	14	7	7	
4.1	Модули, пакеты, импорты в Python	2	1	1	
4.2	Регулярные выражения	4	2	2	
4.3	Веб-скрапинг	4	2	2	
4.4	Итераторы, генераторы	2	1	1	
4.5	Декораторы	2	1	1	
5	Django: создание функциональных	16	8	8	

	веб-приложений				
5.1	Знакомство с Django. Подготовка и запуск проекта.	2	1	1	
5.2	Обработка запросов	2	1	1	
5.3	Динамическое формирование страниц на основе шаблонов	2	1	1	
5.4	Базы данных (2 части)	4	2	2	
5.5	Знакомство с API на примере Django REST framework	2	1	1	
5.6	Django REST framework: фильтрация, валидация, аутентификация	2	1	1	
5.7	Тестирование Django-приложений с использованием Pytest	2	1	1	
6	Python в веб-разработке	16	8	8	
6.1	Flask	2	1	1	
6.2	Docker	4	2	2	
6.3	CI/CD	2	1	1	
6.4	Asyncio	4	2	2	
6.5	Aiohttp	2	1	1	
6.6	Celery	2	1	1	
7	Итоговый проект	20	1	19	
7.1	Выполнение проекта	18	1	17	
7.2	Подведение итогов	2	0	2	Защита проекта

3. СОДЕРЖАНИЕ ПРОГРАММЫ

ТЕМА 1: Основы языка программирования Python

1.1 Переменные, числа, арифметика.

Теоретическая работа: Вводная лекция, знакомство с учениками. Проведение инструктажа по технике безопасности. Работа с числами. Арифметические действия. Работа с переменными.

Практическая работа: Знакомство с IDE. Арифметические действия для решения задач. Ввод и вывод информации.

1.2 Строки.

Теоретическая работа: Знакомство со строками. Функции и методы строк.

Практическая работа: Выполнение базовых операций над строками: конкатенация, дублирование, определение длины строки (функция len), доступ к символам по индексу. Методы строк.

1.3 Условные конструкции. Операции сравнения.

Теоретическая работа: Знакомство с логическими функциями и/или/не, использование логических элементов в ветвлении. Правила оформления блоков ветвления с помощью отступов.

Практическая работа: Создание ветвлений, использование множественных ветвлений при решении задач.

1.4 Типы данных. Индексация и срезы строк.

Теоретическая работа: Знакомство со строками. Функции и методы строк. Индексация. Операции над строками. Конкатенация. Повторение. Решение задач на извлечение среза из строки.

Практическая работа: Выполнение базовых операций над строками: конкатенация, дублирование, определение длины строки (функция len), доступ к символам по индексу.

1.5 Списки. Кортежи.

Теоретическая работа: Что такое список, его отличие от строки, различные операции над списками, методы специфичные для списка. Что такое кортежи, как и для чего их используют. В чем отличие кортежей от списков.

Практическая работа: Решение задач на арифметические действия со списками, поиск минимального или максимального числа в списке, добавить элемент или удалить элемент из списка.

1.6 Циклы.

Теоретическая работа: Знакомство с понятием цикла, общий вид цикла. Знакомство с командами остановки и прерывания цикла break, continue. Знакомство с функцией range в циклах for. Генерация ряда чисел в заданном диапазоне. Знакомство с вложенными циклами.

Практическая работа: Практическое использование циклов While и for. Практическое использование отладчика в Pycharm

1.7 Словари.

Теоретическая работа: Способы создания словарей. Генераторы словарей. Ключи. Способы создания. Операции над словарями. Генераторы словарей. Методы: Clear, сору и др.

Практическая работа: Выполнение заданий по созданию словарей. Поиску по ключу, изменение и вывод словарей.

1.8 Множества.

Теоретическая работа: Что такое множество, методы множества.

Практическая работа: Решение задач на создание, преобразование, удаление коллекций. Использование циклов для получения данных из коллекций.

1.9 Функции.

Теоретическая работа: Знакомство с функциями. Как возвращать значение из функции. Возвращения значений из функции. Инструкции return, def.

Практическая работа: С помощью инструкции def объявляем функцию, с помощью инструкции return возвращаем значение функции.

1.10 Параметры функции.

Теоретическая работа: Отличие между переменной и значением. Функции, изменяющие значение аргумента. Id Объектов. Изменяемость и неизменяемость объектов. Распаковка и запаковка значений. Аргументы по умолчанию. Именованные аргументы. Инструкция pass. Согласованность аргументов.

Практическая работа: Написание программ с получением функцией различных значений, и преобразование значений согласно коду программы. Работа с функциями. Получение неизвестного количества именованных и неименованных аргументов.

1.11 Области видимости переменных.

Теоретическая работа: Области локальной и глобальной видимости, области нелокальной видимости. Операторы global, nonlocal.

Практическая работа: Использование инструкций global и nonlocal, при решении задач с функциями.

1.12 ООП: объекты и классы. Взаимодействие между ними.

Теоретическая работа: Знакомство с особенностями ООП в общем и его реализацией в языке Python. Классы. Методы. Вызов методов.

Практическая работа: Создание собственных методов и классов.

1.13 ООП: наследование, инкапсуляция, полиморфизм.

Теоретическая работа: Знакомство с наследованием параметров класса. Оператор class. Наследование. Дерево атрибутов.

Практическая работа: Написание программы для регистрации учеников группы.

1.14 Работа с файлами.

Теоретическая работа: Способы создания файла. Экспорт данных из файла. Чтение файла. Сохранение в файл. Библиотека OS. Относительный и абсолютный адрес файла.

Практическая работа: Получение информации из текстового файла, обработка информации и сохранение данных.

1.15 Работа с различными форматами данных: CSV, JSON, YAML, XML.

Теоретическая работа: Типы файлов для хранения данных. Библиотеки для работы с файлами. Получение данных и сохранение данных в файл.

Практическая работа: Получение словаря из json файла. И обработка данных из словаря.

1.16 Библиотека requests, http-запросы.

Теоретическая работа: Работа с библиотекой requests. Что такое запросы. Возвращаемые коды. Знакомство с API.

Практическая работа: Получение данных с сайта по API запросу.

1.17 Проектная работа.

Теоретическая работа: Устройство API яндекс.диска. Работа с API VK.

Практическая работа: Написание программы для резервного копирования фотографий из VK на ЯД.

ТЕМА 2: Git-система контроля версий

2.1 Введение в систему контроля версий Git.

Теоретическая работа: Теория о системе контроля версий. Отличие Git от других VCS.

Практическая работа: Установка Git на Linux, macOS, Windows.

2.2 Внедрение системы контроля Git & GitHub.

Теоретическая работа: Работа в локальном репозитории, фиксация изменений, игнорирование файлов.

Практическая работа: Регистрация на Github, создание репозитория, создание файла readme.md.

2.3 История работы и ветки Git & GitHub.

Теоретическая работа: Ветки. Теги. Работа с историей. Откат изменений.

Практическая работа: Работа по слиянию веток репозитория, откат изменений и создание новых веток.

ТЕМА 3: Базы данных для python разработчиков

3.1 Введение в базы данных. Типы баз данных.

Теоретическая работа: БД и СУБД. Типы БД. Таблицы, атрибуты, кортежи. Первичный ключ и связи между таблицами. Нормальные формы

Практическая работа: Проектирование схемы для музыкального сайта.

3.2 Работа с PostgreSQL. Создание БД

Теоретическая работа: Что такое SQL. Типы запросов. Типы полей. Связи между отношениями.

Практическая работа: Создание схемы с различными типами связей между отношениями. Создание SQL-Запросов для создания БД.

3.3 Select-запросы, выборки из одной таблицы

Теоретическая работа: запросы: INSERT, UPDATE, DELETE. SELECT-запросы и арифметические операторы.

Практическая работа: Заполнение базы данных с помощью INSERT запросов. Создание SELECT – запросов

3.4 Продвинутая выборка данных

Теоретическая работа: Агрегирующие функции и операторы. Объединение таблиц с помощью JOIN. Индексы.

Практическая работа: Написание продвинутых SELECT запросов.

3.5 Работа с PostgreSQL из Python

Теоретическая работа: Работа с библиотекой psycopg2. Подключение к БД. Создание SELECT-запросов.

Практическая работа: Создание программу для управления клиентами на Python.

3.6 Python и БД. ORM

Теоретическая работа: Концепт ORM. Абстракция ORM.

Практическая работа: С помощью SQLAlchemy реализовать БД, по заданной схеме. Составить запросы выборки данных.

ТЕМА 4: Профессиональная работа с Python

4.1 Модули, пакеты, импорты в Python.

Теоретическая работа: Модули. Пакеты. Импорты. Декомпозиция. PIP

Практическая работа: Разработка структуры программы по шаблону. Знакомство с модулем datetime.

4.2 Регулярные выражения.

Теоретическая работа: Что такое регулярные выражения. Синтаксис регулярных выражений. Модуль re в Python. Примеры использования.

Практическая работа: Применение регулярных выражений для форматирования адресной книги.

4.3 Веб-скрапинг

Теоретическая работа: Что такое web-scraping. Проблемы при скрапинге. Инструменты. Извлечение информации.

Практическая работа: Поиск статьи на сайте. Сохранение лога данных.

4.4 Итераторы, генераторы

Теоретическая работа: List и память. Как работает цикл for. Итераторы. Генератор и Yield.

Практическая работа: Написание итератора и генератора для получения данных из списка.

4.5 Декораторы

Теоретическая работа: Паттерны проектирования. Как изменить функцию. Декораторы. Область видимости. Параметризованный декоратор.

Практическая работа: Написание декоратора-логгера.

ТЕМА 5: Django: создание функциональных веб-приложений

5.1 Знакомство с Django. Подготовка и запуск проекта.

Теоретическая работа: Что такое Django. Создание проекта. Функции. URL.

Практическая работа: Реализация view функций и настройка URL.

5.2 Обработка запросов.

Теоретическая работа: Лендинг. Пагинация. Работа с файловым менеджером.

Практическая работа: Тестирование дизайна лендингов. Релизация пагинации данных.

5.3 Динамическое формирование страниц на основе шаблонов.

Теоретическая работа: Создание шаблонов. Хед-контейнер. Фут-контейнер.

Практическая работа: Создание статистических таблиц по данным с сайта.

5.4 Базы данных.

Теоретическая работа: Работа с ORM в Django.

Практическая работа: Создание онлайн-библиотеки.

5.5 Знакомство с API на примере Django REST framework.

Теоретическая работа: Создание, чтение, обновление, удаление сущностей по API.

Практическая работа: Создание программы для получения данных с API обработка и сохранение в БД.

5.6 Django REST framework: фильтрация, валидация, аутентификация.

Теоретическая работа: Работа с модулем валидации Django REST framework.

Практическая работа: Регистрация и аутентификация пользователей на сайте с помощью Django REST framework.

5.7 Тестирование Django-приложений с использованием Pytest.

Теоретическая работа: Библиотека Pytest. Создание декораторов для запуска. Сравнения результата.

Практическая работа: По готовому приложению создать тесты.

ТЕМА 6: Python в веб-разработке

6.1 Flask.

Теоретическая работа: Знакомство с Flask, POST, GET, DELETE запросы.

Практическая работа: Создание системы авторизации сайта с помощью Flask.

6.2 Docker.

Теоретическая работа: Знакомство с Docker. Работа с контейнерами.

Практическая работа: Создание собственного контейнера Docker для Flask приложения.

6.3 CI/CD.

Теоретическая работа: Непрерывная интеграция. Непрерывное тестирование.

Практическая работа: Запуск собственного приложения на Flask на Heroku. Запустить процессы CI/CD.

6.4 Asyncio.

Теоретическая работа: Знакомство с модулем Asyncio. Корутины. Футуры. Блокирующие функции.

Практическая работа: Создание асинхронных функций для выгрузки данных по API.

6.5 Aiohttp.

Теоретическая работа: Создание проекта. Структура проекта. View. Route. Создание шаблона. Запуск приложения.

Практическая работа: Переписать Flask приложение на Aiohttp.

6.6 Celery.

Теоретическая работа: Долго-выполняющиеся фоновые задачи. Работа с Celery.

Практическая работа: Создание асинхронного метода API для рассылки сообщений.

ТЕМА 7: Итоговый проект

7.1 Выполнение проект.

Теоретическая работа: Постановка задачи. Распределение на команды.

Практическая работа: Работа по выполнению проекта, по поставленной задаче.

7.2 Подведение итогов

Теоретическая работа: Защита работ учеников.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Материально – техническое обеспечение:

1. Стол преподавателя;
2. Стул преподавателя;
3. Стол обучающегося;
4. Стул обучающегося;
5. Ноутбук преподавателя HP Victus;
6. Ноутбук обучающегося HP Victus;
7. Интерактивная панель Classic Solution;
8. Точки подключения к электрической сети;
9. Подключение к интернету.

Программное обеспечение:

10. Операционная система Windows 10
11. Пакет программ MS OFFICE
12. PyCharm

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

- М. Лутц. Изучаем PYTHON. СПб.: Символ-Плюс, 2019
- Б. Адитья. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. СПб.: Питер, 2019
- В. Дронов. Django 2.1. Практика создания веб-сайтов на Python. СПб.: БХВ-Петербург 2019;
- С. Чаконн, Б. Штрауб. Pro Git, APRESS, Нью-Йорк, 2020

Электронные ресурсы:

- Сайт pythonworld.ru – «Python 3 для начинающих»
- Сайт pythontutor.ru – «Питонтьютор»

6. СВЕДЕНИЯ О СОСТАВИТЕЛЕ ДООП

Комаров Е.А. педагог дополнительного образования, преподаватель по направлениям «Python для новичков». Образование высшее. ФГБОУ ВО «Казанский государственный энергетический университет» диплом по направлению «Электроэнергетика и электротехника» квалификация «Бакалавр». АНО ДПО «ФИПКИП» квалификация «учитель информатики»

ПРИМЕРЫ ЗАДАНИЙ И КРИТЕРИИ ОЦЕНИВАНИЯ

Примеры заданий для проверочной работы:

1. Что будет напечатано при исполнении следующего кода? `print(type(1/2))`

<input type="radio"/> class <'int'>	<input type="radio"/> class <'float'>
<input type="radio"/> class <'number'>	<input type="radio"/> class <'tuple'>

2. Что выведет следующая программа?

`x = 4.5`

`y = 2`

`print(x // y)`

<input type="radio"/> 2	<input type="radio"/> 9
<input type="radio"/> 2.25	<input type="radio"/> 20.25

3. Что будет напечатано при запуске программы?

```
x = True
y = False
z = False
if not x or y:
    print(1)
elif not x or not y and z:
    print(2)
elif not x or y or not y and x:
    print(3)
else:
    print(4)
```

<input type="radio"/> 1	<input type="radio"/> 3
<input type="radio"/> 2	<input type="radio"/> 4

4. Как обозначается тип данных целых чисел?

<input type="radio"/> Float	<input type="radio"/> Int
<input type="radio"/> Str	<input type="radio"/> Vool

5. Что будет выведено на экран?

`a=5`

`b=6`

`print(a)`

<input type="radio"/> 5	<input type="radio"/> 11
<input type="radio"/> 6	<input type="radio"/> 1

6. Чему будет выведено на экран после выполнения кода?

a = 5

b = 10/2

print(a==b)

<input type="radio"/> True	<input type="radio"/> False
<input type="radio"/> ошибка	<input type="radio"/> 0

7. Чему равно c?

a=2

b=5

if a>b:

 c=a

else:

 c=b

<input type="radio"/> 2	<input type="radio"/> 0
<input type="radio"/> 5	<input type="radio"/> Нет правильного ответа

8. До каких пор будут выполняться операторы в теле цикла: while x<100:

<input type="radio"/> Пока x меньше или равен 100	<input type="radio"/> Пока x больше 100
<input type="radio"/> Пока x меньше 100	<input type="radio"/> Пока x равен 100

9. Какие значения будет принимать переменная i, при организации цикла:

For i in range(12,3,-3):

<input type="radio"/> 9, 6, 3	<input type="radio"/> 9, 6
<input type="radio"/> 12, 9, 6, 3	<input type="radio"/> 12, 9, 6

10. Дана программная строка: for i in range(0,10,2): , число 2 - это...

<input type="radio"/> Начало цикла	<input type="radio"/> Счетчик
<input type="radio"/> Конец цикла	<input type="radio"/> Шаг

Критерий оценивания.

Оценка за контроль ключевых компетенций учащихся производится по пятибалльной системе. При выполнении заданий ставится отметка:

«3» - за 50-70% правильно выполненных заданий,

«4» - за 70-85% правильно выполненных заданий,

«5» - за правильное выполнение более 85% заданий.