

Центр цифрового образования детей «IT- куб»  
(структурное подразделение АНО ДТ «Красноярский Кванториум»)

РЕКОМЕНДОВАНО  
методическим советом

Протокол № 2  
от «1» 09 2022 г.

УТВЕРЖДАЮ  
Генеральный директор  
Кениг С.Р.

Приказ № 50  
от «5» 09 2022 г.



Дополнительная общеобразовательная общеразвивающая программа  
технической направленности

«Программирование на языке Python (2 год обучения)»

Срок реализации:  
1 год  
Возраст детей:  
11-17 лет  
Составитель программы:  
Артемьев И.В.

г. Красноярск, 2022 г.



## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python (2 год обучения)» технической направленности, для обучающихся от 11-17 лет, объем годовой нагрузки 144 часа, направлена на приобретение базовых навыков проектной деятельности и разработку собственных проектов.

Для освоения данной программы необходимы знания, полученные при изучении программы «Программирование на языке Python».

Сегодня важным звеном, а также ожидаемым карьерным развитием специалиста в любой компании и особенно в IT-сфере является деятельность по управлению проектами или «проектный менеджмент». Успех любого проекта в IT, будь то создание веб-сайта, интернет-магазина, работа или запуск нового мобильного приложения, зависит во многом от менеджера проекта (руководителя проектной команды), который наряду с глубокими знаниями в области технологий и программирования, должен владеть навыками, методами и инструментами управления проектом и его ресурсами.

Знания, полученные в результате изучения программы «Программирование на языке Python (2 год обучения)», помогут ученикам освоить базовые понятия в управлении проектами, а также на практике применить теоретические знания для решения собственных проектных задач.

### **Новизна программы**

Осознание уникальности и многогранности разработки проектов в сфере IT может стать мотивацией учеников к погружению в сферу профессиональной деятельности, а освоение ее современных подходов и инструментов станет выгодной инвестицией в их будущее, ведь решение любой жизненной задачи можно рассматривать как проект, а методологию управления проектами считать одним из самых эффективных способов успешного достижения поставленной цели.

Данная программа позволяет обучающимся пройти весь жизненный цикл IT-проекта от идеи до запуска продукта и получить необходимый опыт разработки.

Высокую вовлеченность обучающихся обеспечивает самостоятельный выбор темы проекта.

Программа рассчитана на четыре этапа обучения:

- 1) Закрепление базовых навыков
- 2) Управление проектами
- 3) Серверная разработка
- 4) Машинное обучение

Данная программа формирует учебные компетенции, которые позволят обучающимся в будущем создавать собственные программы, игры, сайты. Программа направлена на развитие у детей навыков по управлению проектами, совершенствование навыков программирования и разработки ПО, представления о профессии программиста и менеджера по проектам.

### **Актуальность программы**

Большинство курсов и образовательных программ по программированию нацелены только на изучение языка программирования. Данная программа нацелена на изучение программирования с точки зрения разработки конкретного проекта, и позволит обучающимся увидеть более полную картину работы программиста и разработки проекта.

Изучение основ управления IT-проектами способствуют формированию современного мировоззрения, развитию интеллектуальных способностей, развивает способности к самообразованию, умению ставить задачи, планировать их выполнение, умению разбивать решение задачи на подзадачи, способность работать с информацией: находить, оценивать и использовать информацию из различных источников. Обучение по данной программе дает обучающимся возможность реализовать собственные проекты.

### **Педагогическая целесообразность**

Над кодом мобильных приложений, игр, корпоративных информационных систем, созданием программно-аппаратных комплексов, цифровых гаджетов, web-сервисов и т.п. работает команда высококлассных профессионалов различной специализации: программисты и инженеры, дизайнеры и тестировщики и т.п., которые превращают идеи в реальные продукты, необходимые миллионам людей. Такими сложными командами необходимо уметь управлять. Поэтому важным звеном в деятельности любой IT-компании является управление проектами или «проектный менеджмент».

В термин «управление проектами» разные люди вкладывают разный смысл, зачастую не имея четкого представления о моделях, методах, инструментах и технологиях управления проектами. Между тем, управление проектами – это профессиональная деятельность, основанная на использовании современных научных знаний, навыков, методов, средств и технологий и ориентированная на эффективное получение желаемых результатов.

Применение методологии управления проектами дает возможность четко определить цели и результаты проекта, дать им количественные характеристики, временные, стоимостные и качественные параметры проекта, создать четкий план проекта, выделить, оценить риски и предотвратить возможные негативные последствия во время реализации проекта.

На сегодняшний день методология управления проектами доказала свое право считаться одним из самых эффективных способов успешной реализации поставленных целей.

Проектная деятельность, связанная с непрерывным улучшением и изменением функциональности IT-продуктов, ближе к изобретательству и творчеству, чем к ремеслу, поэтому не всегда хорошо прогнозируется. Это в свою очередь, служит причиной сложности организации процесса взаимодействия между всеми заинтересованными сторонами: заказчиками,

инвесторами, участниками команды разработчиков, пользователями, увеличению количества связей и точек их пересечения, что объективно вызывает трудности в принятии решений, ведет к ошибкам в продукте, превышению стоимости разработки, затягиванию выполнения графиков работ.

Кроме того, важный фактор разработки программного обеспечения – потенциально огромное количество пользователей продукта труда разработчиков. Компьютерные программы разрабатываются для людей (заказчиков/пользователей) и, следовательно, всегда сопровождается постоянным социальным взаимодействием не только внутри команды, но и с людьми «снаружи».

Социальный аспект в разработке программного обеспечения – это очень важная составляющая многих IT-проектов, которая отличает их от проектов создания материальных вещей.

Осознание сложности и уникальности управления IT-проектами, в свою очередь, может служить и мотивацией учеников к погружению именно в эту сферу профессиональной деятельности, овладению искусством укладываться в сроки, бюджет проекта, рационально использовать ресурсы проекта, организовывать работу команды специалистов - с одной стороны; прогнозировать, каким должен быть и будет продукт, насколько он будет востребован - с другой.

В программе применены методики реальной разработки IT-проекта, что для традиционного обучения не свойственно. Программа не ограничивается разработкой только проектной идеи, она рассчитана на создание конкурентоспособного продукта.

### **Цель программы**

Цель программы.

Сформировать навыки работы с технологиями разработки и управления проектами

Задачи:

- 1) сформировать у обучающихся базовые представления о проектах и особенностях проектного подхода к организации деятельности на различных примерах
- 2) сформировать представление о специфике IT-проектов и их основных особенностях;
- 3) сформировать представление о методах управления проектами;
- 4) развить навыки коллективной разработки;
- 5) усовершенствовать навыки разработки, тестирования и отладки несложных программ;
- 6) сформировать самостоятельность и творческий подход к решению проектных задач;
- 7) сформировать навыки работы с документацией к языку программирования и программным продуктам.

8) расширить кругозор обучающихся в области программирования.

**Отличительные особенности данной дополнительной общеразвивающей программы от существующих образовательных программ**

Основной целью программы является создание условий для осознания учениками важности построения дальнейшей индивидуальной образовательной траектории и ранней профориентации через знакомство с перспективными направлениями развития IT-индустрии, спектром IT-профессий и возможностями профессиональной самореализации, спецификой деятельности различных специалистов.

Предметом и главной темой обсуждения в этой программе станет управление проектами - деятельность, направленная на достижение поставленных задач, реализацию определённых планов, используя имеющиеся ресурсы - время, капитал, людей, материалы и инструменты; минимизацию рисков и отклонений от плана, эффективного управления изменениями.

В ходе программы важно изменить сложившийся у большинства людей стереотип: чтобы состояться как специалист в IT-индустрии недостаточно научиться хорошо программировать.

Важно показать, что возрастает значение профессиональных компетенций коллективной разработки программного обеспечения, знание современных направлений, методов и технологий управления IT-проектами:

- 1) понимание обязанностей различных участников команды по разработке программного обеспечения: руководитель разработки программного обеспечения, руководитель технической группы (team leader), архитектор, программист, тестировщик, дизайнер, верстальщик, аналитик;
- 2) владение современными стратегиями и технологиями организации коллективной разработки программного обеспечения, включая системы управления версиями, процессы непрерывной интеграции, стандарты оформления кода и методы инспекции кода;
- 3) понимание основных направлений развития методов коллективной разработки, их отличий и целесообразности применения в зависимости от типа решаемых задач и требований организации;
- 4) владение методологиями разработки программных продуктов.

**Возраст обучающихся, участвующих в программе**

Программа «Программирование на языке Python (2 год обучения)» рассчитана на обучающихся 11-17 лет.

**Условия вхождения в программу:**

Набор на Программу осуществляется в соответствии с Положением о наборе в АНО ДТ «Красноярский «Кванториум».

Поступающий на программу должен успешно пройти ДООП «Программирование на языке Python».

**Срок реализации программы:**

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет 144 часа.

**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 40 минут) с обязательным перерывом, что определяется Санитарно-эпидемиологическими правилами и нормативами СанПиН 2.4.4.3172-14.

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

При проведении занятий используются следующие формы работы:

- 1) демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах;
- 2) фронтальная, когда обучающиеся синхронно работают под управлением педагога;
- 3) самостоятельная, когда обучающиеся выполняют индивидуальные задания в течение части занятия;
- 4) повторение и усвоение пройденного материала осуществляется через контрольные и проверочные работы, анализ полученных результатов;
- 5) закрепление знаний, умений и навыков через постановку задачи и самостоятельную работу обучающегося под руководством педагога;
- 6) применение полученных знаний и навыков через прикладную работу обучающегося, использующего на практике приобретенные компетенции;
- 7) мозговые штурмы для генерации идей и гипотез по проекту;
- 8) параллельная разработка, когда обучающиеся вместе работают над одним проектом, но разрабатывают разный функционал;
- 9) разработка в режиме спринтов, когда обучающиеся разрабатывают определенный объем работы, решают подзадачу в короткий интервал времени;
- 10) стендапы, когда обучающиеся собираются и отчитываются по результатам спринта, чтобы обсудить ход выполнения работы. Цель такого совещания заключается в выявлении проблем, которые могут повлиять на достижение цели спринта.

**Ожидаемые образовательные результаты и эффекты, способы предъявления и отслеживания результатов**

Личностное и профессиональное самоопределение (мотивация к получению профессий в наукоемких областях через интерес к организационно-управленческому аспекту в IT-индустрии) - один из важнейших планируемых образовательных результатов.

Поэтому, знакомя учеников с современным методами управления IT-проектами, которые призваны способствовать развитию экономики, улучшить качество жизни и работы людей, ускорить и изменить процессы и способы решения различных задач. Следует обратить внимание обучающихся на то, что процессы управления IT-проектами характеризуются многоплановостью и многофункциональностью, требуют от специалистов дополнительных навыков и компетенций, многие из которых связаны как с умением программировать, так и с навыками планирования, принятия решений и продуктивной коммуникации, высокой степенью личной ответственности и профессиональной исполнительности. Освоение начальных форм познавательной и личной рефлексии - одна из основных задач.

Отличительной особенностью образовательной программы является ориентация на кооперированную деятельность учеников, на развитие у них коммуникативных универсальных учебных действий:

- 1) умение организовывать продуктивное сотрудничество и совместную деятельность с одноклассниками;
- 2) работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов;
- 3) формулировать, аргументировать и отстаивать своё мнение.

Обучающиеся учатся работать самостоятельно и в команде, вступать в диалог и вести его, выполнять разные роли и обязанности. Приобретаемый ими социальный опыт и навыки практически значимы.

Учащиеся будут знать:

- 1) возможности современных и перспективных средств разработки программных продуктов, технических средств;
- 2) методологии разработки программного обеспечения и технологии программирования;
- 3) методологии управления проектами.

Учащиеся будут уметь:

- 1) вырабатывать варианты реализации программного обеспечения;
- 2) проводить оценку и обоснование выбранных решений;
- 3) разрабатывать алгоритм задачи;
- 4) работать в сети интернет для поиска информации;
- 5) работать в команде;
- 6) управлять собственными проектами.

Личностные результаты:

- 1) Учащиеся приобретут навыки самостоятельной организации своей деятельности, саморазвития и самовоспитания.



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

Метапредметные результаты:

- 1) умение планировать;
- 2) умение анализировать;
- 3) умение алгоритмизировать;
- 4) умение организовывать свой труд и труд команды.

Предметные результаты:

- 1) сформированы базовые представления о проектах и особенностях проектного подхода к организации деятельности на различных примерах
- 2) сформировано представление о специфике IT-проектов и их основных особенностях;
- 3) сформировано представление о методах управления проектами;
- 4) развиты навыки коллективной разработки;
- 5) усовершенствованы навыки разработки, тестирования и отладки несложных программ;
- 6) сформированы навыки работы с документацией к языку программирования и программным продуктам.
- 7) Получен опыт проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств;

#### **Механизм оценки результативности**

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

По итогам каждого этапа проводится итоговая аттестация в форме представления части проектной работы в форме презентации и демонстрации программной части.

Промежуточный контроль – защита проектной идеи, проводится через полгода после начала занятий. Итоговый – защита проекта, в конце учебного года.

## УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика
<b>1</b>	<b>Закрепление базовых навыков</b>	<b>20</b>	<b>5</b>	<b>15</b>
1.1	Вычисления	2	1	1
1.2	Условия	4	1	3
1.3	Циклы	4	1	3
1.4	Списки, кортежи, множества. Словари	4	1	3
1.5	Функции, библиотеки	6	1	5
<b>2</b>	<b>Управление проектами</b>	<b>34</b>	<b>13</b>	<b>21</b>
2.1	Методологии управления проектом	6	2	4
2.2	Как правильно определить тему проекта	2	1	1
2.3	Формулирование актуальности проекта	2	1	1
2.4	Поиск проблемы	2	1	1
2.5	Формулировка цели	2	1	1
2.6	Формулировка задач	2	1	1
2.7	Поиск аналогов. Анализ	2	1	1
2.8	Проработка технического решения	4	1	3
2.9	Изучение технологий производства	4	2	2
2.10	Составление плана работы	2	1	1
2.11	Оформление презентации	4	1	3
2.12	Промежуточная аттестация. Предзащита проектной идеи	2	0	2
<b>3</b>	<b>Серверная разработка</b>	<b>72</b>	<b>31</b>	<b>41</b>
3.1	Введение в серверную часть	2	2	0
3.2	Обзор Клиент-Сервера	2	1	1
3.3	Фреймворки серверной части	2	2	0
3.4	Безопасность веб-сайта	2	2	0
3.5	Введение в Flask	4	2	2
3.6	Настройка Flask	4	1	3
3.7	Создание скелета веб-сайта	4	1	3
3.8	Использование моделей	4	2	2

3.9	Flask admin веб-сайта	6	2	4
3.10	Создание главной страницы	6	2	4
3.11	Общий список и подробные представления	4	1	3
3.12	Структура сессий	4	1	3
3.13	Авторизация и права пользователей	8	3	5
3.14	Работа с формами	6	2	4
3.15	Тестирование веб-приложения Flask	6	2	4
3.16	Деплой Flask на продакшн	4	2	2
3.17	Безопасность веб-приложений Flask	4	3	1
<b>4</b>	<b>Машинное обучение</b>	<b>18</b>	<b>7</b>	<b>11</b>
4.1	Введение в Машинное Обучение	2	1	1
4.2	Основы Компьютерного Зрения	8	3	5
4.3	Введение в сверточные нейронные сети	6	3	3
4.4	Итоговая аттестация	2	0	2
<b>Итого</b>		<b>144</b>	<b>46</b>	<b>98</b>

## Содержание программы

### 1.1 Тема: Вычисления

**Теоретическая работа:** Простые типы данных: целый, вещественный, символьный, логический и их представление в ПК. Приведение типов. Работа с переменными и константами. Использование математических функций. Организация ввода и вывода данных.

**Практическая работа:** Работа с простыми типами данных: целым, вещественным, символьным и логическим. Организация ввода и вывода данных.

### 1.2 Тема: Условия

**Теоретическая работа:** Конструкция ветвления. Разветвляющиеся программы. Условные выражения. Операции отношения. Логические операции. Полная и сокращенная форма условного оператора. Блоки и составные операторы. Оператор множественного выбора.

**Практическая работа:** Работа с логическим типом данных. Объявление и использование в программе переменных, констант и выражений логического типа. Использование простых и составных условий. Полная и сокращенная формы условного оператора. Создание разветвляющейся программы с помощью условных операторов. Оператор множественного выбора. Организация множественного разветвления в программе.

### 1.3 Тема: Циклы

**Теоретическая работа:** Циклы. Циклические программы. Понятия итерации, счетчика и параметров цикла. Циклов со счетчиком. Циклы с предусловием.

**Практическая работа:** повтор команд. Переменные цикла. Операторы while. Оператор for. Бесконечный цикл. Выход из цикла.

### 1.4 Тема: Списки, кортежи, множества, Словари

**Теоретическая работа:** Основы множеств и алгебры логики. Круги Эйлера, списки. Индексация и обратная индексация, схожесть со строками. Отличия кортежей от списков. Практическое использование списков.

**Практическая работа:** Работа типами данных: Кортежи, tuple().Списки. Организация ввода и вывода данных. Срезы. Встроенные методы.

### 1.5 Тема: Функции, библиотеки

**Теоретическая работа:** Создание функций с переменными аргументами. Лямбда-функции. Подходы к программированию. Как правильно разбивать программу на функции. Вывод из функции. Аргументы.

**Практическая работа:** Создание функций. Аргументы. Правильный вывод и ввод данных. создание глобальных и локальных переменных и структур данных. Вызов глобальных переменных и ограничения локальных. функция return. Присваивание функций.

### 2.1 Тема: Методологии управления проектом

**Теоретическая работа:** Что такое проект и командный проект. Отличия проекта от другой командной работы. Основные методологии управления проектом. Гибкие методологии. Scrum. Agile. Водопад. MVP.

**Практическая работа:** Работа на тренажере. Жизнь как проект. Разбор популярных сервисов на MVP и последующие версии.

## **2.2 Тема: Как правильно определить тему проекта**

**Теоретическая работа:** Откуда берется тема проекта. Проблематика. Виды проектов и зависимость темы проекта от вида.

**Практическая работа:** Распределение темы на виды проектов. Составление тем готовых проектов.

## **2.3 Тема: Формулирование актуальности проекта.**

**Теоретическая работа:** Обоснование актуальности проекта. Примеры актуальности проекта.

**Практическая работа:** Определение актуальности тестовых проектов.

## **2.4 Тема: Поиск проблемы**

**Теоретическая работа:** Проблемное интервью. HADI-циклы. Анализ проблемы

**Практическая работа:** Определение основных проблем, которые существуют в конкретной ситуации.

## **2.5 Тема: Формулировка цели.**

**Теоретическая работа:** На какой вопрос должна отвечать цель. Примеры целей проектов.

**Практическая работа:** Определение цели тестовых проектов.

## **2.6 Тема: Формулировка задач.**

**Теоретическая работа:** На какой вопрос должна отвечать задача. Примеры задач проектов. Отличия цели от задачи и их связь.

**Практическая работа:** Определение задач к целям тестовых проектов.

## **2.7 Тема: Поиск аналогов. Анализ.**

**Теоретическая работа:** Понятие аналога и анализа. Поиск аналогов и методы анализа информации.

**Практическая работа:** Построение сравнительной таблицы аналогов.

## **2.8 Тема: Проработка технического решения.**

**Теоретическая работа:** Сферы разработки проектов. Виды проектов в IT. Технические особенности каждого вида.

**Практическая работа:** Определение сферы проекта. Определение технического решения для конкретного проекта. Особенности реализации.

## **2.9 Тема: Изучение технологий производства.**

**Теоретическая работа:** Определение сложности проекта. Роли в проектах разных видов. Распределение компетенций.

**Практическая работа:** Распределение ролей в зависимости от компетенций. Определение сложности конкретного проекта.

## **2.10 Тема: Составление плана работы.**

**Теоретическая работа:** Основные шаги для составления плана работы.

**Практическая работа:** Примерный календарный план. Объем работ. Распределение ролей.

## **2.11 Тема: Оформление презентации.**

**Теоретическая работа:** Структура презентации. Сценарий. Содержание. Подача. Дизайн.

**Практическая работа:** Разработка сценария презентации. Разработка презентации в tilda или power point. Разработка и отработка выступления.

## **2.12 Промежуточная аттестация**

**Практическая работа:** Предзащита проектной идеи.

### **3.1 Тема: Введение в серверную часть.**

**Теоретическая работа:** «Что такое программирование серверной части?», «чем оно отличается от программирования клиентской части?» и «почему это настолько востребовано?».

### **3.2 Тема: Обзор Клиент-Сервера.**

**Теоретическая работа:** Структура клиент-серверного приложения. Динамические запросы.

**Практическая работа:** Построение структуры клиент-серверного приложения с базой данных.

### **3.3 Тема: Фреймворки серверной части.**

**Теоретическая работа:** Что такое фреймворк? Возможности фреймворков. Как выбрать фреймворк? Определение разницы между фреймворками и примеры работы. Объявление и обработка простых HTTP-запросов.

### **3.4 Тема: Безопасность веб-сайта.**

**Теоретическая работа:** Первые шаги для повышения устойчивости сайта. Основные угрозы. Распространенные атаки.

### **3.5 Тема: Введение в Flask.**

**Теоретическая работа:** Что такое Flask, какие функции он предоставляет, и основные строительные блоки приложения Flask.

**Практическая работа:** Установка Flask, создание проекта, отправка запроса и обработка запроса.

### **3.6 Тема: Настройка Flask.**

**Теоретическая работа:** Работа с базами данных. Какую базу данных выбрать и для каких задач.

**Практическая работа:** Компиляция простых программ и запуск уже готовых проектов на сервере. Обработка запросов.

### **3.7 Тема: Создание скелета веб-сайта.**

**Теоретическая работа:** Какие инструменты есть в Flask. Настройка всей структуры приложения целиком.

**Практическая работа:** Создание проекта. Создание приложения. Регистрация приложения. Настройка базы данных. Подключение базы данных. Запуск сайта.

### **3.8 Тема: Использование моделей.**

**Теоретическая работа:** Классификация моделей. Поля. Аргументы полей. Типы полей. Управление моделью.

**Практическая работа:** Составление минимальной модели. Определение моделей. Создание и изменение записей. Поиск записей.

### **3.9 Тема: Flask admin веб-сайта.**

**Теоретическая работа:** Преимущества и ограничения админ-панели Flask.

**Практическая работа:** Регистрация моделей. Создание супер-пользователя. Вход в админ-панель и ее использование. Конфигурация.

### **3.10 Тема: Создание главной страницы.**

**Теоретическая работа:** Как создавать простые url-преобразования. Как получать данные из моделей и создавать шаблоны. URL-адреса страниц

**Практическая работа:** Определение URL-адреса страниц. Создание главной страницы сайта. Шаблоны.

### **3.11 Тема: Отображение списков и детальной информации**

**Теоретическая работа:** Обобщенные базовые классы отображения. Паттерны URL-адресов для передачи информации в отображения

**Практическая работа:** Создание URL-преобразования, отображения и шаблоны страниц. Шаблон условного оператора и цикла. Доступ к переменным.

### **3.12 Тема: Структура сессий**

**Теоретическая работа:** Понятие сессии. Принцип работы сессий.

**Практическая работа:** Подключение сессий. Применение сессий. Хранение данных сессий. Разбор примера - подсчет числа визитов.

### **3.13 Тема: Авторизация и права пользователей**

**Теоретическая работа:** Авторизация и аутентификация и их отличия. Механизм разграничения прав пользователя.

**Практическая работа:** Подключение аутентификации. Создание пользователей и групп. Настройка представлений проверки. Шаблон входа/выхода. Форма сброса пароля. Права доступа.

### **3.14 Тема: Работа с формами**

**Теоретическая работа:** Создание форм. Формы HTML. Класс Form. Класс ModelForm.

**Практическая работа:** Процесс управления формой в Flask. Класс Form и функция отображения. Поля формы. Валидация. Копирование класса формы.

### **3.15 Тема: Тестирование веб-приложения Flask**

**Теоретическая работа:** Типы тестирования. Юнит-тесты. Что необходимо тестировать. Структуры тестов.

**Практическая работа:** Запуск готовых тестов. Определение результатов по тестовой информации. Тестирование LocalLibrary. Инструменты для тестирования.

### **3.16 Тема: Развертывание сайта на сервере**

**Теоретическая работа:** Где и как можно развернуть Flask-приложение для публичного доступа. Что такое окружение развертывания. Выбор хостинг провайдера.

**Практическая работа:** Подготовка сайта к публикации. Установка проекта на Heroku. Создание репозитория приложения на GitHub. Настройка базы данных.

### **3.17 Тема: Безопасность веб-приложений Flask**

**Теоретическая работа:** Понять, что нужно делать (или наоборот не делать), для обеспечения безопасности вашего веб-приложения. Распространенные угрозы/методы защиты.

**Практическая работа:** Межсайтовый скриптинг (XSS). Межсайтовая подделка запроса (CSRF). Разбор других атак.

## **4.1 Тема: Введение в Машинное Обучение**

**Теоретическая работа:** Что такое машинное обучение и искусственный интеллект. Какие задачи решает машинное обучение.

#### **4.2 Тема: Основы Компьютерного Зрения**

**Теоретическая работа:** Знакомство с TensorFlow. Принцип тренировки нейросети.

**Практическая работа:** Простейший граф и нейрон на TensorFlow. Работа с набором данных Fashion MNIST для распознавания элементов гардероба.

**Практическая работа:** Работа с TensorFlow.

#### **4.3 Тема: Введение в сверточные нейронные сети**

**Теоретическая работа:** Что такое свертка. Чем сверточные нейронные сети сильны в задачах компьютерного зрения.

**Практическая работа:** Нейросетевая библиотека Keras. Классификация изображения.

#### **4.4 Тема: Итоговый контроль**

**Практическая работа:** Защита проекта.



## МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ПРОГРАММЕ

№ п/п	Наименование оборудования (ФПО)	Примерная модель (РВПО)	Единица измерения	Количество
1	<b>"Презентационное оборудование"</b>			
1.1	Напольная мобильная стойка для интерактивных досок или универсальное настенное крепление	ONKRON TS1330	шт	1
1.2	Моноблочное интерактивное устройство	SMART SBID-MX265-V2	шт	1
2	<b>"Дополнительное оборудование"</b>			
2.1	Комплект кабелей и переходников	Atcom High speed HDMI - HDMI MOST Lite LRG FAZA FOP-05GS-500	шт	4
2.2	Флипчарт магнитно-маркерный на треноге	Attache	шт	1
3	<b>"Профильное оборудование"</b>			
3.1	WEB-камера (рабочее место педагога)	logitech C922	шт	1
3.2	Наушники (рабочее место обучающегося)	SVEN AP-320M/321M	шт	12
3.3	МФУ (принтер, сканер, копир) (рабочее место педагога)	Epson L14150	шт	1
3.4	Ноутбук тип 2 (рабочее место обучающегося)	Acer TravelMate P2 TMP215-52	шт	12
3.5	Ноутбук тип 1 (рабочее место педагога)	Acer TravelMate P2 TMP215-52	шт	1

## СПИСОК ЛИТЕРАТУРЫ

1. Гаско, Рик. Простой Python просто с нуля [Электронный ресурс] / Рик Гаско. – М.: СОЛОН-Пресс, 2019.
2. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] / Д.М. Златопольский. – М.: БИНОМ. Лаборатория знаний, 2007.
3. Златопольский, Д. М. Сборник задач по программированию [Электронный ресурс] / Д.М. Златопольский. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011.
4. Меньшиков, Ф.В. Олимпиадные задачи по программированию [Электронный ресурс] / Ф.В. Меньшиков. – СПб.: Питер, 2006.
5. Мюллер, Джон Пол. Python для чайников [Электронный ресурс] / Джон Пол Мюллер. - 2-е изд.: Пер с англ. – СПб.: ООО «Диалектика», 2019.
6. Окулов, С.М. Основы программирования: учебное пособие [Электронный ресурс] / С.М. Окулов. – 10-е изд., электрон. – Москва : Лаборатория знаний, 2020.
7. Прохоренок, Н.А. Python 3. Самое необходимое [Электронный ресурс] / Н. А. Прохоренок, В. А. Дронов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2019.
8. Пупышев, В.В. 128 задач по началам программирования [Электронный ресурс] / В.В. Пупышев. – М.: БИНОМ. Лаборатория знаний, 2009.
9. Пэйн, Брайсон. Python для детей и родителей [Электронный ресурс] / Брайсон Пейн [пер. с англ. М.А. Райтмана]. - М.: Издательство «Э», 2017.
10. Свейгарт, Эл. Учим Python, делая крутые игры [Электронный ресурс] / Эл. Свейгарт. - М: Эксмо, 2018.
11. Седер, Наоми. Python. Экспресс-курс [Электронный ресурс] / Наоми Седер. - 3-е изд. – СПб.: Питер, 2019.
12. Столяров, А.В. Оформление программного кода [Электронный ресурс] / А.В. Столяров. – 2-е изд., испр. и доп. – Москва: МАКС Пресс, 2019.
13. Шень, А. Программирование: теоремы и задачи [Электронный ресурс] / А. Шень. - 6-е изд., дополненное. М.: МЦНМО, 2017.
14. Шуман, Х.Г. Python для детей [Электронный ресурс] / Х.Г. Шуман [пер. с нем. М.А. Райтман]. – М.: ДМК Пресс, 2019.
15. <https://www.python.org>
16. <https://learn.javascript.ru>
17. <https://developer.mozilla.org/ru/>
18. <https://www.tensorflow.org/>