

Автономная некоммерческая организация  
«Красноярский детский технопарк «Кванториум»

РЕКОМЕНДОВАНО  
методическим советом

протокол № 4  
от «30» мая 2019 г.

УТВЕРЖДАЮ

Генеральный директор  
Кениг С.Р.

Приказ № 25  
от «31» мая 2019 г.



Дополнительная общеобразовательная общеразвивающая программа  
технической направленности

«Основы программирования»

Срок реализации:

1 год

Возраст детей:

12-18 лет

Составитель программы:

Кондратов А.С.

г. Красноярск, 2019 г.

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования» технической направленности, для обучающихся от 12-18 лет, объем годовой нагрузки 144 часа, направлена на приобретение базовых навыков программирования, развитие основ алгоритмического мышления.

В настоящее время компьютеры окружают современного человека повсюду и не многие понимают то, как работают те сервисы и программы, которыми, порой, мы пользуемся ежедневно. Программирование - это не просто написание программ для компьютера, это целое искусство воплотить идею в свою программу. Знание основ программирования может помочь не только начинающим программистам продолжить свой путь в этой перспективной области, но и тем, чья жизнь в дальнейшем не будет тесно связана с написанием программ. Имея эти знания человек в дальнейшем будет способен автоматизировать решение рутинных задач и бытовых проблем, с которыми мы сталкиваемся в жизни. Не важно, чем человек занимается в жизни, так называемое «вычислительное мышление» поможет думать более методично и более осторожно, формулируя все проблемы как последовательность шагов к их решению. Работаете ли вы социологом, медиком, юристом, знание основ программирования поможет вам. Изучение программирования даёт детям новые возможности и инструменты для самовыражения. Изучение программирования не сложнее изучения иностранных языков, а порой, даже, намного легче.

### **Новизна программы**

Предлагаемая программа нацелена не на изучение конкретного языка программирования или одной области IT, а на обзор всех сфер IT, начиная от низкоуровневого программирования, знание которого позволит в будущем легче научиться работать с микроконтроллерами, например, Arduino или RaspberryPi, до высокоуровневого и web программирования. Ведь в самом начале становления программистом многим не ясно, какая область программирования их заинтересует больше всего, а если с самого начала углублённо изучать незнакомую область, то можно просто потерять впустую время, а, возможно, и полностью отбить желание заниматься этим. Обучение по программе позволяет получить практические навыки и знания, выходящие за рамки школьных программ по информатике и математике.

Программа рассчитана на четыре этапа обучения:

1. Освоение базовых принципов работы электронных вычислительных устройств и знакомство с визуальным программированием
2. Освоение базовых принципов программирования используя язык C++
3. Освоение базовых принципов компьютерных сетей, web-технологий и web-программирования

#### 4. Освоение базовых принципов программирования используя язык Python

Данная программа формирует профессиональные компетенции, которые позволят обучающимся в будущем создавать собственные программы, игры, сайты. Программа направлена на развитие у детей логического мышления, вычислительного мышления, совершенствование первичных навыков программирования, представления о профессии программиста и специалиста в области информационных технологий.

##### **Актуальность программы**

Большинство курсов и образовательных программ по программированию нацелены на изучение конкретной области, считая, что ученик уже знаком с тем, что ему придётся изучать в течение курса. Однако данная программа нацелена на людей, никогда не занимавшихся программированием, и позволит им увидеть всю многогранность компьютерных наук, однако даже те, кто изучает уже в школе программирование или изучал его на дополнительных курсах наверняка найдут что-то интересное в данной программе.

Изучение основ программирования способствуют формированию современного научного мировоззрения, развитию интеллектуальных способностей и познавательных интересов обучающихся, 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 попробовать силы в реализации собственных проектов, получить признание в своей социальной среде. Программирование - это не просто знание какого-то языка или технологии, это способность сформулировать проблему, найти возможные решения этой проблемы путём определения алгоритмов её решения и определить, по возможности, самый оптимальный алгоритм решения. После прохождения данной программы ребёнку будет проще определить наиболее интересующую его область программирования и в дальнейшем уже осознанно подходить к выбору дальнейшего направления обучения.

##### **Педагогическая целесообразность**

Выбор своего первого языка программирования — это очень сложная задача и поэтому курс построен таким образом, чтобы помочь учащимся заняться программированием, найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни и понять в каком направлении им хотелось бы развиваться дальше. Многие люди стремятся заняться информационными технологиями, не предполагая о всех возможностях этой области. Применение детьми на практике теоретических знаний, полученных в школе на математике или физике, ведет к более глубокому пониманию основ, закрепляет полученные навыки, формируя

понимание того, для чего в школе изучается теоретический материал на уроках, который не всегда очевидным образом применяется в реальной жизни и в реальных задачах.

Изучение основ программирования связано с целым рядом умений и навыков, которые могут помочь в дальнейшем в жизни не только будущим программистам, но и тем, чья жизнь не будет напрямую связана с написанием программ.

Изучая программирование, обучающиеся получают глубокое понимание принципов работы компьютера, организации ввода, вывода и хранения информации, принципов построения диалоговых приложений, познают азы профессии программиста.

В рамках данной программы обучающийся познакомится с визуальной объектно-ориентированной средой программирования Scratch, затем он познакомится, пожалуй, с одним из самых сложных, но самым универсальным языком программирования C++, после чего увидит, как создаются сайты и как на странице сайта появляется «динамика», не используя какие-либо конструкторы, а затем познакомится с таким современным языком программирования как Python. Все задачи, которые будут решать дети будут даваться в игровой форме, а в конце курса, в качестве итоговой аттестации, обучающиеся напишут свою небольшую игру, используя все знания, которые они получили во время прохождения курса.

Дополнительная общеразвивающая программа «Основы программирования» рассчитана на детей 12-18 летнего возраста.

Содержание программы раскрывает процесс достижения результатов обучения по всем видам деятельности обучающихся: развитие интеллектуальных способностей и познавательных интересов обучающихся через овладение технологиями создания программных продуктов.

Процесс обучения выстроен с учетом используемых современных образовательных технологий: информационные технологии обучения, технология проектного обучения, что отражено в дидактических принципах обучения (индивидуальность, доступность, преемственность, результативность и др.). Наряду с технологичностью обучения в образовательном процессе решаются задачи организации и управления. В соответствии с учебным планом программы на каждом этапе обучения разработаны занятия, наиболее эффективные для развития познавательного потенциала обучающихся. Виды деятельности обучающихся отражены в предлагаемых формах организации обучения, что и определяет продолжительность по времени, динамичную структуру занятия, способы организации процесса обучения и его методическое оснащение. В техническом творчестве и проектной технологии ведущее место занимают интерактивные и активные методы обучения, обеспечивающие эффективное достижение поставленной цели обучения (в данной предметной области) и отслеживание результативности обучения на всех этапах.

## **Цель программы**

Формирование навыков работы с технологиями программирования на языках Scratch, Python, C++, JavaScript для создания собственных проектов.

Задачи:

- сформировать у обучающихся базовые представления о языках программирования, алгоритме (программе), исполнителе, способах записи алгоритма;
- сформировать представление о профессии «программист»;
- сформировать навыки составления основных алгоритмических конструкций (линейных, разветвляющихся и циклических);
- сформировать навыки составления алгоритмов;
- сформировать навыки разработки, тестирования и отладки несложных программ;
- познакомить с понятием проекта и алгоритмом его разработки;
- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники
- знакомство с принципами и методами функционального программирования
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач
- расширение кругозора обучающихся в области программирования

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

Программа представляет четыре этапа, которые охватывают базовые особенности разных разделов программирования. Обучение начинается со знакомством с визуальным программированием с использованием Scratch, затем переходит в низкоуровневое и среднеуровневое, используя язык C++, затем web-программирование, используя HTML, CSS и JS и заканчивается программа языком Python. Каждый этап, кроме Scratch, предусматривает промежуточную аттестацию в форме представления и защиты проекта (программного продукта).

Отличительной особенностью программы «Основы программирования» является возможность освоения нескольких программных сред Scratch, C++, JS и Python.

На первом этапе обучающиеся знакомятся с языком программирования Scratch. Он особенно интересен для начального уровня изучения программирования и знакомством с алгоритмами, и базовыми языковыми конструкциями, такими как: переменные, условия, циклы и так далее. Обучение основам программирования в этой среде наиболее эффективно при выполнении небольших проектов. При этом естественным образом обучающийся овладевает интерфейсом новой для него среды, постепенно углубляясь как в возможности Scratch, так и в идеи программирования. На

втором этапе дети знакомятся с языком С++, который за минимальными исключениями знаком всем программистам. Несмотря на то, что этот язык появился уже очень давно, он до сих пор развивается и в наши дни, С++ программисты по-прежнему очень нужны. На третьем этапе дети узнают, как создаются сайты и какие для этого применяются технологии. Будут рассмотрены основы вёрстки сайтов, используя HTML и CSS и добавление «динамики» на сайт используя язык JavaScript. Последний этап рассматривает язык Python, который обладает рядом преимуществ по сравнению с С-подобными языками, такими как С, С++ или Java. Код написанный на Python весьма лаконичен, синтаксис понятен и относительно прост в изучении и освоении, он поддерживает основные модели программирования: объектно-ориентированное, функциональное и структурное.

### **Возраст обучающихся, участвующих в программе**

Программа «Основы программирования» рассчитана на обучающихся 12-18 лет.

### **Условия вхождения в программу:**

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

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).

### **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 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.

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

По результатам обучения, обучающиеся овладевают базовыми основами технологии программирования на языках C++, JavaScript, Python, а также Scratch, HTML, CSS для создания собственных проектов.

Учащиеся будут знать:

- возможности визуального программирования;
- основные отличия визуального программирования от традиционного;
- основные приемы написания программ-приложений;
- требования к написанию и оформлению программ-приложений;
- основные принципы работы компьютера и выполнения программы;
- основные алгоритмические конструкции;
- учащиеся знают простейшие типы данных и основные операторы языка программирования, что такое функции, параметры, глобальные и локальные переменные.

Учащиеся будут уметь:

- настраивать окружение интегрированной среды в соответствии с решаемой задачей;
- правильно интерпретировать получаемые результаты в ходе тестирования и отладки программ;
- пользоваться готовыми компонентами для разработки приложений;
- умеют разрабатывать алгоритм задачи;
- умеют работать в сети интернет для поиска информации;
- могут разработать консольное приложение из нескольких функций с использованием вложенных циклов и условий.

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.

**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 самообразовании; эффективной организации индивидуального информационного пространства.

**Механизм оценки результативности**

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

По итогам каждого этапа проводится итоговая аттестация в форме небольшой проектной работы, являющаяся промежуточной аттестацией по программе.



## УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика
<b>1</b>	<b>Знакомство с направлением обучения</b>	<b>2</b>	<b>2</b>	<b>0</b>
	Водный урок. Правила работы и техника безопасности при работе на компьютере	1	1	0
	Математика вокруг нас	1	1	0
<b>2</b>	<b>Вычислительное мышление и Scratch</b>	<b>6</b>	<b>3</b>	<b>3</b>
	Основы работы компьютера	2	1	1
	Scratch	4	2	2
<b>3</b>	<b>Язык C++</b>	<b>44</b>	<b>18</b>	<b>26</b>
	Visual Studio 2019, Базовые основы C++	2	1	1
	Основные операторы, булева алгебра	6	3	3
	Операторы цикла	6	2	4
	Массивы и указатели	6	2	4
	Функции	6	2	4
	Алгоритмы	4	2	2
	Строки	4	2	2
	Структуры	4	2	2
	Файловые потоки	4	2	2
	Промежуточная аттестация - тест	2	0	2
<b>4</b>	<b>HTTP, HTML, CSS, JS</b>	<b>42</b>	<b>17</b>	<b>25</b>
	Сети, TCP/IP HTTP, URLs, DNS	1	1	0
	Типы данных, переменные, массивы в JS	3	1	2
	Условия и циклы	4	2	2
	Функции	4	2	2
	Основы HTML, CSS, Forms	4	2	2
	Связь JavaScript с HTML	4	2	2
	Объекты	4	2	2
	DOM	2	1	1
	Элемент canvas	6	2	4
	Промежуточная аттестация - создание игры на JS	10	2	8
<b>5</b>	<b>Python</b>	<b>50</b>	<b>20</b>	<b>30</b>
	Введение в Python, сравнение с C++ и JS	1	1	0
	Типы данных в Python	3	2	1
	Строки, списки, кортежи и словари	4	2	2
	Условные конструкции	4	1	3
	Циклы	6	2	4
	Функции	4	2	2
	«Модули» Python	2	1	1
	Рисование с помощью «черепашки»	4	2	2
	«Исключения» в программировании	2	1	1
	Основы ООП в Python	2	1	1
	Создания графического интерфейса программ	6	3	3
	Итоговая аттестация - создание игры на Python	12	2	10
	<b>ИТОГ</b>	<b>144</b>	<b>60</b>	<b>84</b>

## Содержание программы

**Тема:** Математика вокруг нас.

**Теоретическая работа:** вводная лекция, которая покажет, как и где математика встречается в нашей жизни, в том числе и в программировании.

**Тема:** Основы работы компьютера.

**Теоретическая работа:** что такое информатика, системы счисления, представления данных, алгоритмы, псевдокод.

**Практическая работа:** работа с двоичной системой счисления, как соотносятся двоичная система счисления с десятичной. Перевод числа из одной системы счисления в другую. Абстракции в информатике. Как компьютер представляют изображения. Понятие алгоритма и оценка их сложности. Написание псевдокода.

**Тема:** Scratch.

**Теоретическая работа:** введение в визуальное программирование. Знакомство со средой Scratch. Установка и запуск Scratch. Интерфейс Scratch. Главное меню Scratch. Особенности графического редактора среды Scratch.

**Практическая работа:** знакомство со средой Scratch. Установка и запуск Scratch. Интерфейс Scratch. Главное меню Scratch. Примеры программ, созданные в Scratch. Работа со «строительными блоками» Scratch. Создание небольших игр.

**Тема:** Visual Studio 2019, базовые основы C++.

**Теоретическая работа:** что такое IDE и зачем она нужна. Чем визуальное программирование отличается от программирования на языке C++. Структура программы. Для чего используется C++. Ключевые слова в языке.

**Практическая работа:** установка IDE и её настройка. Структура программы. Написание первой программы на языке C++.

**Тема:** Основные операторы, булева алгебра.

**Теоретическая работа:** основные языковые конструкции C++. Понятие переменной и как она храниться в памяти компьютера. Зарезервированные слова. Ввод и вывод переменных. Проблема чисел с плавающей запятой. Условные операторы. Булева алгебра. Правила записи выражений и операторов. Основные операции языка. Приоритеты операций. 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:** использование переменных. Работа с различными типами данных. Что нужно знать про арифметику с плавающей

запятой. Команды if-else. Сравнение чисел. Булева алгебра в программировании.

**Тема:** Циклы.

**Теоретическая работа:** циклы. Циклические программы. Понятия итерации, счетчика и параметров цикла. Циклов со счетчиком. Циклы с предусловием. Циклы с постусловием. Синтаксис, реализация и способы применения всех видов циклов. Вложенные циклы. Операторы принудительного завершения итерации цикла. Оператор завершения цикла. Оператор безусловного перехода. Синтаксис, реализация и правила использования.

**Практическая работа:** циклы. Повтор команд. Переменные цикла. Вложенные циклы. Циклы while, for, do-while. Бесконечный цикл. Выход из цикла.

**Тема:** Массивы и указатели.

**Теоретическая работа:** понятие массива, элемента и его индекса. Одномерные массивы. Объявление одномерного массива и его представление в памяти. Алгоритмы работы с массивами: ввод и вывод элементов массивов, нахождение характеристик массивов. Выполнение поиска данных в массиве. Сортировка массива. Матрицы (двумерные массивы).

**Практическая работа:** Массивы. Размерности массивов. Указатели.

**Тема:** Функции.

**Теоретическая работа:** определение функции. Структура функции. Объявление и вызов функций. Параметры функции и возвращаемое функцией значение. Локальные и глобальные переменные. Формальные и фактические параметры. Правила работы с функциями. Способы передачи параметров в функцию. Прототип функции. Константные параметры. Параметр `__cplusplus` по умолчанию.

**Практическая работа:** изучение правил объявления и вызова функции в программе. Задание формальных параметров функции, структуры функции и возвращаемого функцией значения. Объявление локальных и глобальных переменных и работа с ними. Изучение способов передачи параметров в функцию. Составление программ с использованием функций. Реализация концепций структурного программирования с помощью функций. Демонстрация возможностей передачи параметров в функции. Передача в параметрах указателей, ссылочных данных и структур. Передача в функцию одномерных и многомерных массивов данных с помощью передачи указателя на их начало.

**Тема:** Алгоритмы.

**Теоретическая работа:** разделение задачи на подзадачи. Как оценивают алгоритмы. Какой алгоритм выбрать для решения задачи. Парадигма «разделяй и властвуй». Алгоритмы сортировки.

**Практическая работа:** составляем алгоритм решения задачи. Оцениваем полученные алгоритмы. Выбираем наиболее эффективный. Рассматриваем задачу поиска информации в отсортированном списке. Обзор алгоритмов сортировки.

**Тема:** Строки.

**Теоретическая работа:** объявление и инициализация строк. Способы ввода строк с клавиатуры и вывода их на экран. Специальные функции для работы со строками: сравнение строк, удаление из строки символа или подстроки, копирование, объединение и другие. Функции преобразования и классификации отдельных символов. Создание программы, реализующей работы со строковыми данными с помощью специальных функций.

**Практическая работа:** объявление и инициализация строк. способы ввода строк с клавиатуры и вывода их на экран. специальные функции для работы со строками: сравнение строк, удаление из строки символа или подстроки, копирование, объединение и другие. функции преобразования и классификации отдельных символов. создание программы, реализующей работы со строковыми данными с помощью специальных функций.

**Тема:** Структуры.

**Теоретическая работа:** понятие структуры и полей структуры. Объявление структуры и ее представление в памяти. Свойства структур. Синтаксис обращения к полям структур. Сравнение структур. Массив структур

**Практическая работа:** объявление структуры и экземпляра структуры. изучение правил обращения к полям структур. Работа со структурами и использование их для совместного хранения разнотипных данных. Объявление массивов структур и работа с ними. создание программ, выполняющих функции простейших баз данных.

**Тема:** Файловые потоки.

**Теоретическая работа:** понятие текстового файла. Объявление переменной типа файл. Связывание файловой переменной с физическим файлом. Открытие файла. Определение конца файла. Функции ввода и вывода данных из текстовых файлов. Потоки файлов. Входные и выходные потоки. Работа с файлами с помощью потоков.

**Практическая работа:** изучение структуры текстовых файлов и принципов работы с ними с помощью специальных функций. Открытие существующих файлов в программе. программное создание новых файлов. создание программ со считыванием начальных данных из файла и сохранением результатов работы в файл. отработка навыков работы с текстовыми файлами.

## **Промежуточная аттестация**

### **Практическая работа: тест**

**Тема:** Сети, TCP/IP HTTP, URLs, DNS.

**Теоретическая работа:** TCP / IP, как компьютеры могут взаимодействовать друг с другом. Серверы. Протоколы. UTL. DNS.

**Тема:** Типы данных, переменные, массивы в JS.

**Теоретическая работа:** что такое JavaScript. Сравнение с C++. Где применяется JavaScript. Как создавать программы на языке JavaScript.

**Практическая работа:** реализация базовых конструкций языка. Работа с простыми типами данных: целым, вещественным, символьным и логическим.

**Тема:** Условия и циклы.

**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. Циклы. Циклические программы. Понятия итерации, счетчика и параметров цикла. Циклов со счетчиком. Циклы с предусловием. Циклы с постусловием. Синтаксис, реализация и способы применения всех видов циклов. Вложенные циклы. Операторы принудительного завершения итерации цикла. Оператор завершения цикла. Оператор безусловного перехода. Синтаксис, реализация и правила использования.

**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. Работа с различными формами циклических операторов: циклами со счетчиком, циклами с предусловием и с постусловием

**Тема:** Функции.

**Теоретическая работа:** определение функции. Структура функции. Объявление и вызов функций. Параметры функции и возвращаемое функцией значение. Локальные и глобальные переменные. Формальные и фактические параметры. Правила работы с функциями. Способы передачи параметров в функцию.

**Практическая работа:** изучение правил объявления и вызова функции в программе. Задание формальных параметров функции, структуры функции и возвращаемого функцией значения. Объявление локальных и глобальных переменных и работа с ними. Изучение способов передачи параметров в

функцию. Составление программ с использованием функций. Реализация концепций структурного программирования с помощью функций.

**Тема:** Основы HTML, CSS, Forms.

**Теоретическая работа:** как создаются сайты. Язык разметки HTML. Язык стилей CSS.

**Практическая работа:** изучаем основные тэги HTML. Пишем статический сайт. Добавляем стили для сайта.

**Тема:** как JavaScript связан с HTML

**Теоретическая работа:** что такое JavaScript. Сравнение с C++. Где применяется JavaScript. Как создавать программы на языке JavaScript.

**Практическая работа:** Основные конструкции языка JavaScript. Первая программа на JavaScript

**Тема:** Объекты.

**Теоретическая работа:** что такое объекты. Для чего нужны. Как ими пользоваться. Создание объектов. Прототипы. Получение и изменение свойств. Атрибуты свойств. Как работает this.

**Практическая работа:** составление программ использующие объекты. Получение и изменение свойств. Атрибуты свойств. Проверка существования свойств. Чтение и запись свойств. Атрибуты объекта. Удаление свойств.

**Тема:** DOM.

**Теоретическая работа:** модель DOM. Работа с DOM. Деревья. Стандарт JavaScript. Обход дерева. Поиск элементов в DOM. Изменение структуры DOM. Что именно мы получаем от DOM. Что именно мы получаем от DOM.

**Практическая работа:** составление программ использующие DOM. Получение элемента методом getElementById. Как задать атрибут методом setAttribute.

**Тема:** Элемент canvas.

**Теоретическая работа:** canvas. Для чего он нужен. Как пользоваться. Что можно делать. Настройки цвета и размера. Геометрические фигуры.

**Практическая работа:** работа с элементом HTML5 canvas. Применение стилей и цветов. Рисование текста. Использование изображений. Простые анимации.

**Промежуточная аттестация**

**Практическая работа:** создание игры на языке JavaScript.

**Тема:** Введение в Python, сравнение с C++ и JS.

**Теоретическая работа:** структура программы. Задание идентификаторов и операторов. Правила записи выражений и операторов. Основные операции языка.

**Практическая работа:** знакомство с интерфейсом и инструментами среды программирования. Реализация базовых конструкций языка. Компиляция, анализ и проверка программного кода. Создание линейной программы на изучаемом языке программирования.

**Тема:** типы данных в Python.

**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.

**Тема:** Строки, списки, кортежи и словари.

**Теоретическая работа:** новые типы данных. Для чего они нужны и как с ними работать.

**Практическая работа:** работа типами данных: Строки, списки, кортежи и словари. Организация ввода и вывода данных.

**Тема:** Условные конструкции.

**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.

**Тема:** Циклы.

**Теоретическая работа:** циклы. Циклические программы. Понятия итерации, счетчика и параметров цикла. Циклов со счетчиком. Циклы с предусловием. Циклы с постусловием. Синтаксис, реализация и способы применения всех видов циклов. Вложенные циклы. Операторы принудительного завершения итерации цикла. Оператор завершения цикла. Оператор безусловного перехода. Синтаксис, реализация и правила использования.

**Практическая работа:** циклы. Повтор команд. Переменные цикла. Вложенные циклы. Операторы while, for. Бесконечный цикл. Выход из цикла.

**Тема:** Функции

**Теоретическая работа:** Параметры функций. Локальные переменные. Зарезервированное слово «global». Зарезервированное слово «nonlocal». Значения аргументов по умолчанию. Ключевые аргументы. Переменное число параметров. Оператор «return». Строки документации. Аннотации.

**Практическая работа:** импортирование модулей. Работа с модулями

**Тема:** «Модули» Python.

**Теоретическая работа:** что такое модуль. Функции модулей. Импортирование модуля. Стандартные модули.

**Практическая работа:** импортирование модулей. Работа с модулями

**Тема:** Рисование с помощью «черепашки».

**Теоретическая работа:** Знакомство с модулем Turtle.

**Практическая работа:** Работа с модулем Turtle. Рисование простых геометрических фигур. Рисование сложных геометрических фигур.

**Тема:** «Исключения» в программировании.

**Теоретическая работа:** что такое исключения. Для чего они нужны. Когда появляются. Как их обрабатывать. Ошибки. Try . Finally. Оператор with.

**Практическая работа:** работа с исключениями.

**Тема:** Основы ООП в Python.

**Теоретическая работа:** понятия объекта и класса. Поля и методы класса. Иерархия классов. Принципы инкапсуляции, наследования и полиморфизма. Преимущества и недостатки объектно-ориентированного подхода. Этапы объектно-ориентированного программирования.

**Практическая работа:** изучение синтаксиса описания класса, его компонентов и спецификаторов доступа к ним. Изучение правил описания и вызова конструкторов, деструкторов и других методов класса. Объявление объектов и работа с ними. Создание программы, выполняющей работу с классами и объектами.

**Тема:** создание графического интерфейса программ

**Теоретическая работа:** Понятия GUI. Модуль Tkinter и библиотека PyQt.

**Практическая работа:** создание своего первого графического интерфейса. Создание виджета Label. Настройка размера и шрифта текста. Настройка размеров окна приложения. Добавление виджета Button.

**Итоговая аттестация**



**Практическая работа:** создание игры на Python.

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

Помещения для лабораторных занятий и самостоятельной работы обучающихся должны быть оснащены персональными компьютерами с установленной средой разработки программного обеспечения на языке объектно-ориентированного программирования Microsoft Visual Studio Community 2019 (и выше), IDE PyCharm 2018.3.5 (и выше), Python 3.7.4 (и выше), Visual Studio Code 1.35.0 (и выше), расширение для VSCode - Live Server, расширение для VSCode - CSS Peak, расширение для VSCode - Prettier, Microsoft Office 2013 (и выше), с возможностью подключения к сети Интернет и доска с маркером или интерактивная доска. Помещение для занятий лекционного типа должно быть оснащено мультимедийным проектором и персональными компьютерами с установленной средой разработки программного обеспечения на языке объектно-ориентированного программирования Microsoft Visual Studio Community 2019 (и выше), Microsoft Office 2013 (и выше), с возможностью подключения к сети Интернет, доска с маркером или интерактивная доска.

## **СПИСОК ЛИТЕРАТУРЫ**

1. Борович П.С. Бутко Е.Ю., 2011 г. Учебное пособие «Среда программирования Scratch»
2. . Прата, С. Язык программирования C++: Лекции и упражнения / С. Прата. – М.: Вильямс, 2005. – 1097 с.
3. Шилдт, Г. Теория и практика C++ / Г. Шилдт. – СПб.: ВНУ, 1996. – 416 с.
4. <http://scratch.mit.edu>
5. <https://en.cppreference.com/w/>
6. <https://www.python.org>
7. <https://learn.javascript.ru>