

**БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ОРЛОВСКОЙ ОБЛАСТИ  
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ  
«ДВОРЕЦ ПИОНЕРОВ И ШКОЛЬНИКОВ ИМ. Ю.А. ГАГАРИНА»**

Рассмотрена и принята  
педагогическим Советом  
Протокол № \_\_ от «\_\_» \_\_\_\_\_ 2021 г.



Утверждена  
Директор Н.А. Марушкина  
«\_\_» \_\_\_\_\_ 2021 г.  
Приказ № \_\_\_\_ от «\_\_» \_\_\_\_\_ 2021 г.

Дополнительная общеобразовательная общеразвивающая программа  
технической направленности центра цифрового образования «IT-куб»

**«Программирование на Python»**

Возраст обучающихся: 10-18 лет

Срок реализации: 2 года

Наумкин С.В.

Ст. методист Терпячая Е.С.

## **Содержание:**

### **I. Комплекс основных характеристик программы**

1. Пояснительная записка
2. Цели и задачи общеразвивающей программы
  - 2.1. Цели и задачи «Базового уровня» (первый год обучения)
  - 2.2. Цели и задачи «Продвинутого уровня» (второй год обучения)
3. Содержание общеразвивающей программы
  - 3.1. Содержание учебного (тематического) плана 1-го года обучения (базовый уровень)
  - 3.2. Содержание учебного (тематического) плана 2-го года обучения (продвинутый уровень)
4. Планируемые результаты
  - 4.1. Планируемые результаты «Базового уровня»(первый год обучения)
  - 4.2. Планируемые результаты «Продвинутого уровня»(второй год обучения)

### **II. Комплекс организационно-педагогических условий реализации общеразвивающей программы**

1. Календарный учебный график на 2021-2022 учебный год
2. Условия реализации программы
3. Формы аттестации и оценочные материалы
4. Методические материалы

Список литературы

Приложение

## **I. Комплекс основных характеристик программы**

### **1. Пояснительная записка**

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие обучающимися основ современного программирования и неумение применять полученные знания на практике.

Программа составлена опираясь на уникальный опыт преподавания в Школе анализа данных Яндекса (АНО ДПО «ШАД»). В ней большое внимание уделяется практической и самостоятельной работе.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса.

Научившись программировать на языке Python, обучающиеся получат мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся в будущем с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в

олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

### ***Направленность программы***

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» имеет техническую направленность, что позволяет обучающимся приобщиться к инженерно-техническим знаниям в области инновационных технологий, сформировать техническое мышление.

Основанием для проектирования и реализации данной общеразвивающей программы служит ***перечень следующих нормативных правовых актов и государственных программных документов:***

- Федеральный закон от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;

- Приказ Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 "Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам";

- Приказ Минпросвещения России от 03.09.2019 г. №467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;

- Приказ Минпросвещения России от 2 декабря 2019 г. № 649 «Об утверждении Целевой модели цифровой образовательной среды»;

- Распоряжение Правительства Российской Федерации от 24 апреля 2015 года №729-р «Концепция развития дополнительного образования детей»;

- Распоряжение Правительства Российской Федерации от 29 мая 2015 года №996-р «Концепция развития дополнительного образования детей»;

- Постановление Главного государственного санитарного врача РФ от 04.07.2014 г. № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации

режима работы образовательных организаций дополнительного образования детей»;

- Методические рекомендации по созданию и функционированию центров цифрового образования «IT-куб» (Утверждены распоряжением Министерства просвещения Российской Федерации от 12 января 2021 г. № Р-5);

- Устав БУ ОО ДО «Дворец пионеров и школьников имени Ю. А. Гагарина».

### **Актуальность программы**

Актуальность программы обусловлена высоким интересом подростков к IT- сфере. Важнейшей характеристикой подрастающего поколения является активность в информационном пространстве, интернет-коммуникации. Python является очень востребованным языком программирования - он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Программа «Программирование на Python» в связи с востребованностью на рынке, является своего рода уникальным образовательным продуктом в области информационных технологий. В условиях динамично- развивающейся сферы информационных технологий, потребности кадров, развития инфраструктуры программа соответствует региональным социально- экономическим и социокультурным потребностям и проблемам региона.

### **Прогностичность программы**

Прогностичность дополнительной общеобразовательной общеразвивающей программы «Программирование на Python» заключается в том, что она отражает требования и актуальные тенденции не только сегодняшнего, но и завтрашнего дня и имеет междисциплинарный характер, что полностью отражает современные тенденции построения как дополнительных общеобразовательных программ, так и образования в целом.

Программа предполагает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне; имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту обучающегося; охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний; ориентирована на существующий парк вычислительной техники и дополнительные ограничения; допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня обучающихся (как группового, так и индивидуального), а также предусматривает возможность индивидуальной работы с обучающимися.

Практическая значимость курса заключается в том, что он расширяет возможности обучающихся в области программирования, создания программ, работой с файловой системой, базами данных и т. д., что позволяет применять язык программирования для обработки, хранения, изменения данных, используемых как в файловой системе, так и вебсайтах, анализе данных банковских структур.

Элементы программы курса могут быть рекомендованы для использования учителями информатики при проведении лабораторно-практических и практических занятий.

### ***Отличительная особенность программы***

Отличительная особенность дополнительной общеразвивающей программы «Программирование на Python» в том, что обучение происходит на базе образовательной платформы Яндекс.Лицей. На данной платформе представлены все теоретические материалы, библиотеки, практические и тестовые задания. Каждое занятие дополняется методическим сопровождением (видео, учебники). У каждого ребенка есть своя учётная запись, благодаря которой он может получить доступ к образовательной платформе с любого ПК и самостоятельно использовать материалы для повторения изученного материала и выполнения домашних работ. Педагог

получает статистику по уровню освоения программы каждым ребёнком.

Также программа является практико-ориентированной. Освоение обучающимися навыков программирования происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать IT- технологиями.

### ***Адресат общеразвивающей программы***

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» предназначена для детей в возрасте 10–18 лет, мотивированных к обучению и обладающих системным мышлением.

Формы занятий групповые. Группы формируются по возрасту: 10–14 и 15–18 лет. Количество обучающихся в группе – 12 человек. Состав групп постоянный.

### ***Возрастные особенности группы***

Содержание программы учитывает возрастные и психологические особенности детей в возрасте 10–18 лет, которые определяют выбор форм проведения занятий с обучающимися. Характерная особенность данного подросткового периода заключается в личном самосознании, сознательном проявлении индивидуальности. Ведущая потребность – самоутверждение. В подростковый период стабилизируются интересы детей. Основное новообразование – становление взрослости как стремление к жизни в обществе взрослых. Подростки этого возраста отличаются открытием своего внутреннего мира, внутреннего «Я». Главным измерением времени в самосознании является будущее, к которому он (она) себя готовит.

Также следует отметить, что подростки данной возрастной группы характеризуются такими психическими процессами, как стремление углублённо понять себя, разобраться в своих чувствах, настроениях, мнениях, отношениях. Это порождает у подростка стремление к самоутверждению, самовыражению (проявления себя в тех качествах, которые он считает наиболее ценными) и самовоспитанию. Эти процессы позволяют положить

начало созданию начального профессионального самоопределения обучающихся. Ведущая деятельность в этом возрасте – учебно-профессиональная, в процессе которой формируются такие новообразования, как мировоззрение, профессиональные интересы, самосознание, мечта и идеалы, поэтому основной формой проведения занятий выбраны практические занятия.

**Режим занятий, объём общеразвивающей программы:** длительность одного занятия составляет 2 академических часа, периодичность занятий – 2 раза в неделю.

**Срок освоения общеразвивающей программы** определяется содержанием программы и составляет 2 года (144 часа в год).

**Форма обучения:** очная, возможна реализация очно с применением электронного обучения и дистанционных образовательных технологий (Закон №273-ФЗ, гл.2, ст.17, п.2.).

***Виды занятий общеразвивающей программы:***

Основной тип занятий – комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является модуль. Каждый модуль охватывает отдельную информационную технологию или её часть. Каждая тема курса начинается с постановки задачи – характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.



Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый подросток получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес обучающегося к предмету, активность и самостоятельность обучающихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

По типу организации взаимодействия педагогов с обучающимися при реализации программы используются личностно-ориентированные технологии, технологии сотрудничества.

Реализация программы предполагает использование здоровьесберегающих технологий. Здоровьесберегающая деятельность реализуется:

- через создание безопасных материально-технических условий;
- включением в занятие динамических пауз, периодической смены деятельности обучающихся;
- контролем соблюдения обучающимися правил работы на ПК;
- через создание благоприятного психологического климата в учебной группев целом.

### ***Педагогическая целесообразность программы***

Программа «Программирование на Python» является целостной и непрерывной в течение всего процесса обучения, и позволяет обучающемуся шаг за шагом раскрывать в себе творческие возможности и самореализоваться в современном мире, развить компьютерную грамотность.

Базовый уровень предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка, гарантированно обеспечивает

трансляцию общей и целостной картины в рамках содержательно-тематического направления общеразвивающей программы. Продвинутый уровень предполагает использование форм организации материала, обеспечивающих доступ к сложным и нетривиальным разделам в рамках содержательно-тематического направления общеразвивающей программы. Также предполагает углублённое изучение содержания общеразвивающей программы и доступ к околопрофессиональным и профессиональным знаниям в рамках содержательно-тематического направления общеразвивающей программы. Осваивая данную программу, обучающиеся будут овладевать навыками востребованных на рынке труда. Практически для каждой перспективной профессии будут полезны знания и навыки, рассматриваемые в данной программе.

**Объём общеразвивающей программы:** общее количество учебных часов, запланированных на весь период обучения, необходимых для освоения программы: 288 часов.

По уровню освоения программа общеразвивающая, **разноуровневая** (базовый, продвинутый уровни). Она обеспечивает возможность обучения детей с первичными навыками программирования в том числе, одарённых детей.

#### ***«Базовый уровень» (первый год обучения)***

Рассчитан на детей в возрасте 10–16 лет, проявляющих интерес к IT-технологиям и приобретению навыков программирования.

Зачисление детей на первый год обучения производится по рейтингу. Рейтинг определяется путём сдачи предварительного тестирования.

К концу первого года обучающиеся получают первичные навыки объектно-ориентированного и функционального программирования, алгоритмизации, работы в различных интегрированных средах разработки на языке Python; изучат основные конструкции языка программирования; приобретут навыки поиска, анализа, использования информации в сети Интернет.

#### ***«Продвинутый уровень» (второй год обучения)***

Рассчитан на детей в возрасте 15–18 лет, проявляющих интерес к ИТ-технологиям, желающих совершенствовать свои навыки программирования, имеющих первичный опыт объектно-ориентированного и функционального программирования в различных интегрированных средах разработки на языке Python.

Зачисление детей на второй год обучения после завершения базового уровня производится по результатам успешной сдачи итогового тестирования. Дети, не освоившие базовый уровень, к зачислению на продвинутый уровень не допускаются.

Обучение по программе «Программирование на Python» на втором году нацелено на углубление и структурирование знаний основ современных языков программирования, умение на практике использовать сложные структуры данных, работать со специальными средствами и библиотеками языка Python.

К концу второго года обучения подростки способны самостоятельно: определять профессиональные задачи и пути решения; писать грамотный, красивый код; находить и обрабатывать ошибки в коде; разрабатывать эффективные алгоритмы и программы на основе изученного языка программирования Python; способны самостоятельно изучать новые технологии.

## **2. Цели и задачи общеразвивающей программы**

**Цель программы:** формирование творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Для успешной реализации поставленной цели необходимо решить ряд образовательных, развивающих и воспитательных задач:

## **Задачи:**

### ***Образовательные:***

- познакомить с конструкциями языка программирования Python;
- познакомить с принципами и методами функционального и объектно-ориентированного программирования;
- познакомить с основными структурами данных и типовыми методами обработки этих структур;
- сформировать навыки работы в интегрированной среде разработки на языке Python;
- сформировать навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- сформировать и развить навыки алгоритмического и логического мышления, грамотной разработки программ;

### ***Развивающие:***

- способствовать развитию навыков проектной деятельности;
- способствовать развитию навыков поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- сформировать у обучающихся интерес к программированию, самостоятельность и творческий подход к решению задач с использованием средств вычислительной техники;
- способствовать развитию навыков работы с различными источниками информации, необходимой для решения учебных задач;
- познакомить с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой;
- способствовать развитию умения планировать свои действия с учётом фактора времени, в обстановке с элементами конкуренции, предвидеть результат и достигать его, при необходимости вносить коррективы в первоначальный замысел.

### ***Воспитательные:***

- способствовать воспитанию этики групповой работы;
- сформировать коммуникативную культуру обучающихся, как внутрипроектных групп, так и в коллективе в целом;
- способствовать воспитанию упорства в достижении результата;
- сформировать целеустремлённость, организованность, равнодушие, ответственное отношение к труду и уважительное отношение к окружающим.

## **2.1. Цели и задачи «Базового уровня» (первый год обучения)**

**Цель уровня:** формирование познавательной активности обучающихся в области объектно-ориентированного, функционального программирования и алгоритмизации, а также получение первичного опыта работы с базовыми конструкциями языка программирования на основе работы в интегрированных средах разработки на языке Python.

Для успешной реализации поставленной цели необходимо решить ряд образовательных, развивающих и воспитательных задач:

### **Задачи:**

#### ***Образовательные:***

- сформировать базовые навыки работы с основными конструкциями языка программирования Python;
- познакомить с основными предметными понятиями и их свойствами;
- познакомить с комплексом базовых понятий и принципов функционального и объектно-ориентированного программирования (изучение структур данных, базовые принципы их обработки);
- сформировать навыки работы в интегрированной среде разработки на языке Python;
- сформировать необходимые навыки работы с информацией (поиск, анализ, использование информации в сети Интернет).

#### ***Развивающие:***

- сформировать и развить навыки работы с различными источниками информации, умение самостоятельно искать, извлекать и отбирать необходимую информацию;
- способствовать развитию самостоятельности и творческого подхода к решению задач;
- сформировать логическое и техническое мышление;
- сформировать и развить познавательную потребность в освоении технических знаний;
- познакомить с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

***Воспитательные:***

- способствовать воспитанию этики групповой работы, отношение делового сотрудничества, взаимоуважение;
- сформировать коммуникативную культуру обучающихся, как внутрипроектных групп, так и в коллективе в целом;
- сформировать целеустремлённость, организованность, неравнодушие, ответственное отношение к труду, толерантность и уважительное отношение к окружающим;
- сформировать стремление к получению качественного законченного результата.

**2.2. Цели и задачи «Продвинутого уровня» (второй год обучения)**

**Цель уровня:** углубление и структурирование знаний, полученных на первом году обучения навыков использования структур данных и программирования в различных интегрированных средах разработки на языке Python.

Для успешной реализации поставленной цели необходимо решить ряд образовательных, развивающих и воспитательных задач:

**Задачи:**

***Образовательные:***

- закрепить опыт объектно-ориентированного и функционального

программирования в различных интегрированных средах разработки на языке Python;

- отработать навыки программирования, углубить и структурировать знания основ современных языков программирования;

- сформировать навыки разработки эффективных алгоритмов и программ на основе изученного языка программирования Python;

- способствовать развитию навыков работы со специальными средствами и библиотеками языка Python;

- сформировать навыки написания грамотного, красивого кода, умение находить и обрабатывать ошибки в коде;

- научить на практике использовать сложные структуры данных.

***Развивающие:***

- сформировать навыки логического и алгоритмического мышления;

- способствовать развитию навыков проектной деятельности;

- способствовать развитию самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;

- сформировать общее представление об информационной картине мира, об информации и информационных процессах как элементах реальной действительности;

- способствовать развитию навыков работы с различными источниками информации, необходимой для решения учебных задач;

- сформировать навык планировать свои действия с учётом фактора времени, а также предвидеть результаты своей работы и оптимальные пути их достижения;

- познакомить с правилами индивидуального и коллективного безопасного поведения при работе с компьютерной техникой;

- способствовать развитию навыка самостоятельного изучения новых технологий.

***Воспитательные:***

- способствовать воспитанию этики групповой работы, отношение

делового сотрудничества, взаимоуважение;

- сформировать целеустремлённость, организованность, равнодушие, ответственное отношение к труду, толерантность и уважительное отношение к окружающим;

- сформировать ответственность, равнодушие, взаимоуважение в процессе группового взаимодействия;

- сформировать коммуникативную культуру обучающихся, как внутрипроектных групп, так и в коллективе в целом;

- сформировать стремление к получению качественного законченного результата.

### 3. Содержание общеразвивающей программы

#### Учебный (тематический) план 1-го года обучения (базовый уровень)

Таблица 1

№ п/п	Название раздела, темы	Кол-во часов			Формы аттестации/контроля
		Всего	Теория	Практика	
<b>Базовый уровень</b>					
<b>Модуль 1. Введение в программирование.</b>		<b>26</b>	<b>7</b>	<b>19</b>	
1.1	Вводное занятие. Инструктаж по ТБ. Введение в программу 1-го года обучения	8	2	6	Знакомство. Устный опрос.
1.2	Интегрированные среды, исполнение кода и отладка	10	2	8	Устный опрос, беседа, решение задач
1.3	Переменные, основные операторы	6	2	4	Устный опрос, беседа, решение задач
1.4	Базовые типы данных, ветвления	1	1	0	Беседа. Устный опрос
1.5	Контрольная работа	1	0	1	Контрольное тестирование
<b>Модуль 2. Базовые конструкции в Python</b>		<b>42</b>	<b>6</b>	<b>36</b>	



2.1	Циклы, срезы, списочные выражения.	12	2	10	Устный опрос, беседа, решение задач
2.2	Методы списков и строк. Вложенные списки, знакомство со словарями	8	2	6	Устный опрос, беседа, решение задач
2.3	Решение задач по пройденным темам.	10	2	8	Практическая работа
2.4	Проектная деятельность	12	0	12	Разработка/Защита мини-проекта
<b>Модуль 3. Решение прикладных задачв Python.</b>		<b>76</b>	<b>28</b>	<b>48</b>	
3.1	Области видимости переменных, функции с переменным числом аргументов, функции как объект. Лямбда функции .	8	4	4	Устный опрос, беседа, решение задач
3.2.	Обработка коллекций. Поточковый ввод.	6	2	4	Решение задач
3.3.	Самостоятельная работа	4	2	2	Решение задач
3.4.	Библиотеки Python. Встроенные модули.	8	4	4	Беседа, практическая работа
3.5.	ООП: инкапсуляция, наследование, полиморфизм	6	2	4	Решение задач
3.6	Основы объектно-ориентированного анализа и дизайна, шаблоны проектирования	4	2	2	Устный опрос, беседа, Практическая работа
3.7.	Проектирование и разработка классов 1 часть.	4	2	2	Решение задач
3.8.	Проектирование и разработка классов 2 часть.	6	2	4	Решение задач
3.9.	Tkinter: создание графического интерфейса	8	4	4	Устный опрос, беседа, решение задач
3.10.	Итераторы, генераторы.	8	4	4	Устный опрос, беседа, решение задач
3.11.	Проектная деятельность	16	0	16	Разработка/защита индивидуального/группового проекта

<b>Итого:</b>	<b>144</b>	<b>41</b>	<b>103</b>	
---------------	------------	-----------	------------	--

### **3.1. Содержание учебного (тематического) плана 1-го года обучения (базовый уровень)**

#### **Модуль 1. Введение в программирование**

##### ***Тема 1.1 Вводное занятие. Инструктаж по ТБ. Введение в программу 1-го года обучения***

*Теория:* Инструктажи по технике безопасности с отметкой в журнале. Введение в программу «Программирование на Python» 1-го года обучения. Понятия кода, интерпретатора, программы. Простейшие программы с выводом на экран.

*Практика:* Знакомство с виртуальной средой взаимодействия – регистрация, организация личного кабинета, поиск и выкладывание материалов; знакомство с системой автоматизированной проверки задач и системой Яндекс.

##### ***Тема 1.2 Интегрированные среды, исполнение кода и отладка***

*Теория:* Интегрированные среды, исполнение кода. Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки.

*Практика:* Решение задач.

##### ***Тема 1.3 Переменные, основные операторы***

*Теория:* Условный оператор. Переменные и арифметика. Погружение в условия. Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

*Практика:* Решение задач.

##### ***Тема 1.4 Базовые типы данных, ветвления***

*Теория:* Знакомство со списками, строками, множествами и

кортежами в

Python.

### **Тема 1.5 Контрольная работа**

*Практика:* Выполнение контрольного тестирования.

## **Модуль 2. Базовые конструкции в Python**

### **Тема 2.1 Циклы, срезы, списочные выражения**

*Теория:* Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python. Знакомство со срезами и диапазонами. Равенство и совпадение объектов. Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python. Простейшие циклы и циклы с переменными.

*Практика:* Решение задач.

### **Тема 2.2 Методы списков и строк. Функции**

*Теория:* Списочные выражения. Методы split и join. Другие методы списков и строк. Знакомство с функциями. Области видимости переменных. Работа со списками, строками, множествами и кортежами в Python. Понятие итератора, подпрограммы, процедуры, функции. Функции в Python.

*Практика:* Решение задач.

### **Тема 2.3 Решение задач по пройденным темам.**

*Теория:* Повторение пройденных тем.

*Практика:* Практическая работа.

### **Тема 2.4 Проектная деятельность**

*Практика:* Защита мини-проекта.

## **Модуль 3. Решение прикладных задач в Python**

**Тема 3.1 Области видимости переменных, функции с переменным числом аргументов, функции как объект. Лямбда функции.**

*Теория:* Функции, функциональная парадигма программирования. Понятие лямбда функции и области ее применения.

*Практика:* Решение задач.

### ***Тема 3.2 Обработка коллекций. Поточковый ввод.***

*Теория:* Поточковый ввод sys.stdin, обработка коллекций, работа с потоковым вводом, обработка поточной информации.

*Практика:* Решение задач

### ***Тема 3.3 Самостоятельная работа***

*Практика:* Решение задач

### ***Тема 3.4 Библиотеки Python. Встроенные модули***

*Теория:* Использование встроенных библиотек (работа с графикой, аудио файлами, презентациями, текстовыми файлами, таблицами и диаграммами)

*Практика:* Создание собственного пакета модулей

### ***Тема 3.5 ООП: инкапсуляция, наследование, полиморфизм***

*Теория:* Причины появления, принципы и основные сущности объектно-ориентированного подхода к разработке ПО. Инкапсуляция, полиморфизм, наследование, композиция.

*Практика:* Решение задач.

### ***Тема 3.6 Основы объектно-ориентированного анализа и дизайна, шаблоны проектирования***

*Теория:* Восходящий и нисходящий способы разработки. Методика объектно-ориентированного дизайна приложения. Шаблоны проектирования.

*Практика:* Проектирование и реализация компьютерной игры

### ***Тема 3.7 Проектирование и разработка классов 1 часть.***

*Теория:* Разработка классов для игры «Шахматы», создание функции цвета фигур, реализация отрисовки поля.

*Практика:* Решение задач

### ***Тема 3.8 Проектирование и разработка классов 1 часть.***

*Теория:* Разработка классов фигур, реализация сруба фигур, реализация проверки ходов каждой фигуры в соответствие с правилами игры

*Практика:* Решение задач

### **Тема 3.9 Tkinter: создание графического интерфейса**

*Теория:* Знакомство с библиотекой tkinter, изучение основных команд, виджетов, упаковщиков и приемов работы.

*Практика:* Создание приложения с графическим интерфейсом.

### **Тема 3.10 Итераторы, генераторы.**

*Теория:* Принцип работы итераторов и генераторов, создание собственных итераторов и генерирующих функций

*Практика:* Решение задач

### **Тема 3.11 Проектная деятельность**

*Теория:* Презентация тем итоговых проектных работ. Принципы распределения ролей при выполнении проекта с применением нейронных сетей. Примерное содержание и регламент презентации работы, рекомендации по ее оформлению и выступлению. Критерии оценки итоговых проектов.

*Практика:* Выбор проектного задания. Поиск открытых данных по выбранной теме. Распределение ролей внутри проектной команды. Подготовка группового/ индивидуального проекта. Защита проекта.

## **Учебный (тематический) план 2-го года обучения (продвинутый уровень)**

Таблица 2

№п/п	Название раздела, темы	Кол-во часов			Формы аттестации/контроля
		Всего	Теория	Практика	
<b>Модуль 1. Программирование на Python</b>		<b>40</b>	<b>17</b>	<b>2 3</b>	
1.1	Вводное занятие. Введение в программу 2-го года обучения Инструктаж по ТБ	2	1	1	Опрос. Отметка о прослушанных инструктажах в журнале. Решение задач

1.2	Повторение. Решение задач на основные конструкции и структуры данных. Решение задач на классы и библиотеки	4	3	1	Опрос. Решение задач
1.3	Решение задач по теме «Повторение»	2	0	2	Решение задач
1.4	QT 1. Что такое QT и PyQT	2	2	0	Опрос. Беседа. Решение задач
1.5	QT 2. QtDesigner	4	2	2	Опрос. Беседа. Решение задач
1.6	QT 3. Файлы в Python. Типы файлов и работа с ними. Внутреннее устройство файлов	4	2	2	Опрос. Решение задач
1.7	QT 4. Диалоги, работа с изображениями	4	2	2	Опрос. Беседа. Решение задач
1.8	QT 5. Работа с простыми таблицами (csv). Работа с табличными данными в PyQT	4	2	2	Решение задач
1.9	QT 6. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 1	4	2	2	Решение задач
1.10	QT 7. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 2	4	0	4	Опрос. Беседа. Решение задач
1.11	QT 8. Обработка событий. Сборка независимого приложения	4	1	3	Практическая работа
1.12	Защита проекта QT	2	0	2	Защита проекта
<b>Модуль 2. PyGame</b>		<b>44</b>	<b>17</b>	<b>27</b>	
2.1	Введение в репозитории. Подключение PyCharm. Работа с удаленными репозиториями	4	1	3	Опрос. Беседа. Решение задач
2.2	PyGame 1. Введение	2	1	1	Опрос. Беседа. Решение задач
2.3	PyGame 2. Игровой цикл. События	4	2	2	Решение задач
2.4	PyGame 3. Основные команды при одиночной работе с Git	4	2	2	Решение задач
2.5	PyGame 4. Клеточное поле	4	1	3	Решение задач
2.6	PyGame 5. Классические игры на клеточном поле	4	2	2	Решение задач

2.7	PyGame 6. Изображения. Спрайты	2	1	1	Опрос. Беседа. Решение задач
2.8	PyGame 7. Столкновения и другие взаимодействия	4	2	2	Опрос. Беседа. Решение задач
2.9	PyGame 8. Игра в целом	6	2	4	Практическая работа
2.10	PyGame 9. Украшения игр	4	2	2	Практическая работа
2.11	Цели и подходы к тестированию. Создание "самодельных" тестов (без библиотек)	4	1	3	Практическая работа
2.12	Защита проекта PyGame	2	0	2	Защита проекта
<b>Модуль 3. Проекты WEB и API</b>		<b>60</b>	<b>22</b>	<b>38</b>	
3.1	WEB. Работа с популярными форматами файлов (json, xml)	4	2	2	Опрос. Беседа. Решение задач
3.2	WEB. Знакомство с API	4	2	2	Решение задач
3.3	WEB. Понятие исключения, обработка исключений. Собственные исключения.	4	2	2	Решение задач
3.4	WEB. Работа с командной строкой (скрипты, аргументы). Периодические задачи (модуль shedule)	4	2	2	Опрос. Беседа. Решение задач
3.5	WEB. Библиотека argparse. Задачи на создание скриптов с ее помощью	2	0	2	Опрос. Беседа. Решение задач
3.6	WEB. Работа с протоколом HTTP	4	2	2	Решение задач
3.7	WEB. Решение задач на API Яндекс-карт	4	2	2	Решение задач
3.8	WEB. Введение во flask. Обработка html-форм	4	2	2	Решение задач
3.9	WEB. Шаблоны. flask--wtf	4	2	2	Опрос. Беседа. Решение задач
3.10	WEB. Знакомство с flask-sqlalchemy	4	2	2	Опрос. Беседа. Решение задач
3.11	WEB. Flask-sqlalchemy	2	0	2	Решение задач
3.12	WEB. REST-API. Понятие. Делаем простое REST-API.	4	2	2	Решение задач
3.13	WEB. REST-API. Flask-restfull	2	0	2	Практическая работа
3.14	WEB. Разворачиваем проект в облаке. Дорешка.	4	0	4	Практическая работа

3.15	WEB. Алиса. Часть 1	4	2	2	Опрос. Беседа. Решение задач
3.16	WEB. Алиса. Часть 2	2	0	2	Решение задач
3.17	WEB. Защита проекта WebServer + API	2	0	2	Защита проекта
3.18	Разработка и защита итогового проекта	2	0	2	Защита индивидуального/ группового проекта
	<b>Итого программа года</b>	<b>144</b>	<b>56</b>	<b>88</b>	



## **3.2. Содержание учебного (тематического) плана 2-го года обучения**

**(продвинутый уровень)**

### **Модуль 1. Программирование на Python**

#### ***Тема 1.1 Вводное занятие. Введение в программу 2-го года обучения. Инструктаж по ТБ***

*Теория:* Вводное занятие. Инструктажи по технике безопасности с отметкой в журнале. Введение в программу 2-го года обучения. Общая характеристика программы. Цели и задачи. Тематические модули.

*Практика:* Решение задач

#### ***Тема 1.2 Повторение. Решение задач на основные конструкции и структуры данных. Решение задач на классы и библиотеки***

*Теория:* Повторение основных конструкций и структур данных. Повторение модулей стандартной библиотеки Python.

*Практика:* Решение задач

#### ***Тема 1.3 Решение задач по теме «Повторение»***

*Практика:* Решение задач

#### ***Тема 1.4 QT 1. Что такое QT и PyQT***

*Теория:* Обзор рынка фреймворков для разработки настольных приложений на Python. История, предназначение, преимущества и недостатки фреймворка Qt. Подключение и первые шаги с PyQt.

*Практика:* Решение задач

#### ***Тема 1.5 QT 2. QtDesigner***

*Теория:* Изучение основных элементов графического интерфейса и настроек QtDesigner, способы интеграции графического и текстового стиля создания настольных приложений.

*Практика:* Решение задач, связанных с созданием простейшего однооконного приложения.

#### ***Тема 1.6 QT 3. Файлы в Python. Типы файлов и работа с ними. Внутреннее устройство файлов***

*Теория:* Принципы работы с файловой системой средствами Python.

Чтение и запись информации в файл приложением, созданным при помощи QtDesigner.

*Практика:* Решение задач, связанных с работой с файлами из настольного приложения

#### ***Тема 1.7 QT 4. Диалоги, работа с изображениями***

*Теория:* Изучение набора встроенных в QtDesigner диалоговых окон. Способы чтения, модификации и внедрения графических файлов в настольное приложение.

*Практика:* Решение задач по обработке изображений

#### ***Тема 1.8 QT 5. Работа с простыми таблицами (csv). Работа с табличными данными в PyQT***

*Теория:* Формат данных csv, его преимущества и недостатки. Способы чтения и записи файлов в данном формате. Работа с табличными данными в PyQT в рамках собственного настольного приложения.

*Практика:* Решение задач по работе с таблицами

#### ***Тема 1.9 QT 6. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 1***

*Теория:* Предназначение и разновидности баз данных. Реляционные базы данных и язык запросов SQL. Обращение к базе данных из создаваемого при помощи PyQT настольного приложения.

*Практика:* Решение задач по работе с БД и SQL

#### ***Тема 1.10 QT 7. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 2***

*Теория:* Технология написания более сложных SQL запросов. Объектно-реляционное отображение.

*Практика:* Решение задач на работу с БД из PyQT

#### ***Тема 1.11 QT 8. Обработка событий. Сборка независимого приложения***

*Теория:* Петля событий настольного приложения. Обработка событий стандартных устройств ввода (мышь, клавиатура). Обработка событий,

генерируемых модулями программы. Сборка интерактивного приложения.

*Практика:* Работа над проектом

### ***Тема 1.12 Защита проекта QT***

*Практика:* Защита проекта (настольное приложение)

## **Модуль 2. PyGame**

***Тема 2.1 Введение в репозитории. Подключение PyCharm. Работа с удаленными репозиториями.***

*Теория:* Разновидности и предназначение систем контроля версий. Репозитории для хранения и обмена данными. Использование среды разработки PyCharm с удаленными репозиториями.

*Практика:* Решение задач по работе с репозиторием

### ***Тема 2.2 PyGame 1. Введение***

*Теория:* Что такое PyGame, его место в экосистеме и какие игры можно создавать с его помощью.

*Практика:* Решение задач

### ***Тема 2.3 PyGame 2. Игровой цикл. События***

*Теория:* Создание игрового цикла, обработка событий стандартных устройств ввода.

*Практика:* Решение задач

### ***Тема 2.4 PyGame 3. Основные команды при одиночной работе с Git***

*Теория:* Система контроля версий Git. Использование git при разработке приложения в одиночку — графический и консольный интерфейс, основные команды.

*Практика:* Решение задач

### ***Тема 2.5 PyGame 4. Клеточное поле***

*Теория:* Реализация клеточного поля в PyGame. Способы хранения глобальных настроек и обмена данными между игровыми агентами.

*Практика:* Решение задач

### ***Тема 2.6 PyGame 5. Классические игры на клеточном поле***

*Теория:* Обзор классических игр на клеточном поле и вариантов их реализации при помощи PyGame

*Практика:* Решение задач по реализации игры на клеточном поле

### ***Тема 2.7 PyGame 6. Изображения. Спрайты***

*Теория:* Графические возможности PyGame, чтение, изменение и встраивание изображений. Техника работы со спрайтами.

*Практика:* Решение задач по работе с изображениями

### ***Тема 2.8 PyGame 7. Столкновения и другие взаимодействия***

*Теория:* Обсуждение основных видов взаимодействия между игроками и другими сущностями игры, а также способов программной реализации такого рода взаимодействий

*Практика:* Решение задач

### ***Тема 2.9 PyGame 8. Игра в целом***

*Теория:* Проектирование архитектуры игрового приложения.

*Практика:* Создание своей игры

### ***Тема 2.10 PyGame 9. Украшения игр***

*Теория:* Нюансы создания интерфейса игры, визуального оформления игровой механики, подключение звуковых эффектов.

*Практика:* Украшение своей игры

***Тема 2.11 Цели и подходы к тестированию. Создание «самодельных» тестов (без библиотек)***

*Теория:* как и зачем осуществляют тестирование в промышленной разработке. Подходы к тестированию программных продуктов. Технология создания тестов стандартными средствами Python.

*Практика:* Тестирование своего игрового приложения

### ***Тема 2.12 Защита проекта PyGame***

*Практика:* Защита проекта (игра)

## **Модуль 3. Проекты WEB и API**

***Тема. 3.1 WEB. Работа с популярными форматами файлов (json, xml)***

*Теория:* Структура форматов json и csv. Python-модули для работы с ними.

Основные сферы применения, практика применения в WEB.

*Практика:* Решение задач

### ***Тема 3.2 WEB. Знакомство с API***

*Теория:* Программный интерфейс приложения в разработке для основных существующих платформ. Особенности API в WEB-разработке.

*Практика:* Решение задач

### ***Тема 3.3 WEB. Понятие исключения, обработка исключений.***

***Собственные исключения.***

*Теория:* Программные ошибки, понятие исключительной ситуации.

Механизм исключений в Python, различные практики его использования.

*Практика:* Решение задач - самостоятельная работа

### ***Тема 3.4 WEB. Работа с командной строкой (скрипты, аргументы).***

***Периодические задачи (модуль schedule)***

*Теория:* Интерфейс командной строки основных операционных систем. Его применение в backend-разработке и администрировании.

*Практика:* Решение задач

***Тема 3.5 WEB. Библиотека argparse. Задачи на создание скриптов с ее помощью***

*Теория:* Парсинг аргументов командной строки средствами argparse.

*Практика:* Решение задач

### ***Тема 3.6 WEB. Работа с протоколом HTTP***

*Теория:* Протокол HTTP и клиент-серверная архитектура. Обмен данными с использованием данного протокола.

*Практика:* Решение задач. Самостоятельная работа

### ***Тема 3.7 WEB. Решение задач на API Яндекс-карт***

*Теория:* Картографические сервисы Яндекса для сайтов, мобильных

приложений и логистики. Работа с ними средствами Python применительно к WEB-проектам.

*Практика:* Решение задач

***Тема 3.8 WEB. Введение во flask. Обработка html-форм***

*Теория:* Фреймворк для создания WEB-приложений Flask, структура и базовая механика. Создание и обработка HTML-форм с его помощью.

*Практика:* Решение задач. Самостоятельная работа

***Тема 3.9 WEB. Шаблоны. flask--wtf***

*Теория:* Расширения фреймвора Flask. Предназначение и использование расширения Flask-WTF

*Практика:* Решение задач

***Тема 3.10 WEB. Знакомство с flask-sqlalchemy***

*Теория:* Предназначение и использование расширения (фреймворка Flask)

Flask-SQL-Alchemy

*Практика:* Решение задач

***Тема 3.11 WEB. Flask-sqlalchemy***

*Теория:* Нюансы Flask-SQL-Alchemy

*Практика:* Решение задач. Контрольная работа

***Тема 3.12 WEB. REST-API. Понятие. Делаем простое REST-API***

*Теория:* REST подход к организации взаимодействия компонентов распределенного приложения. REST-API.

*Практика:* Решение задач

***Тема 3.13 WEB. REST-API. Flask-restful***

*Теория:* Расширение (фреймворка Flask) Flask-RESTful для быстрого построения REST-API.

*Практика:* Работа над проектом — создание своего web-сервиса

***Тема 3.14 WEB. Разворачиваем проект в облаке.***

*Домашка Теория:* Размещение web-проекта в сети интернет.

*Практика:* Работа над проектом

***Тема 3.15 WEB. Алиса. Часть 1***

*Теория:* История развития и основные технологии, лежащие в основе голосовых помощников. Яндекс.Диалоги. Алиса.

*Практика:* Решение задач

***Тема 3.16 WEB. Алиса. Часть 2 Теория: Создание навыка для Алисы.***

*Практика:* Решение задач

***Тема 3.17 WEB. Защита проекта WebServer + API***

*Практика:* Защита своего web-проекта

***Тема 3.18 Разработка и защита итогового проекта***

*Практика:* Защита индивидуального/группового проекта

#### **4. Планируемые результаты**

***Предметные результаты:***

По окончании программы обучающийся:

- научится определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке;
- овладеет основными предметными понятиями («информация», «алгоритм», «исполнитель», «модель») и их свойств;
- овладеет логическими способностями и алгоритмическим мышлением, умением составить и записать алгоритм для конкретного исполнителя;
- научится выполнять пошагово (с использованием компьютера или вручную) алгоритмы управления исполнителями и анализа числовых и текстовых данных;

– получит навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ, выполнять созданные программы;

– научится разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;

***Личностные результаты:***

По окончании программы обучающийся:

– овладеет способностью к саморазвитию и самообразованию, будут развиты любознательность, внимательность и настойчивость при выполнении заданий практического характера;

– научится организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности;

– – будет сформировано ответственное отношение к учению, готовность и способность обучающихся к саморазвитию и самообразованию, средствами информационных технологий

– овладеет коммуникативной компетентностью в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности.

– овладеет базой целостного мировоззрения, соответствующего современному уровню развития информационных технологий.

***Метапредметные результаты:***

По окончании программы обучающийся:

– научится самостоятельно ставить и формулировать для себя новые задачи, планировать свои действия, планировать пути решения поставленной задачи для получения эффективного результата, корректировать свои действия;



- получит технические навыки и развитое логическое мышление;
- получит навыки исследовательской и проектной деятельности;
- научится работать с различными источниками информации, извлекать нужную информацию из открытых источников;
- усвоит правила индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

#### **4.1. Планируемые результаты «Базового уровня»(первый год обучения)**

##### ***Предметные результаты:***

По окончании программы обучающийся:

- Овладеет необходимой терминологией («информация», «алгоритм», «исполнитель», «модель»), смысла этих понятий и умение применять полученные знания на практике;
  - овладеет алгоритмическим (знакомство и навыки работы с простейшими алгоритмическими структурами – линейной, условной и циклической), а также логическим мышлением, что необходимо для грамотного составления алгоритмов, рассчитанных для конкретного исполнителя;
  - получит навыки пошагового выполнения алгоритмов управления исполнителями и анализа числовых и текстовых данных, умение осуществлять данные операции как вручную, так и с использованием компьютера;
  - познакомится с основами программирования и областями применения полученных навыков.

##### ***Личностные результаты:***

По окончании программы обучающийся:

- овладеет способностью к саморазвитию и самообразованию, будут развиты любознательность, внимательность и настойчивость при выполнении заданий практического характера;

– научиться организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности;

– будет сформировано ответственное отношение к учению, готовность и способность обучающихся к саморазвитию и самообразованию, средствами информационных технологий

– овладеет коммуникативной компетентностью в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности.

– овладеет базой целостного мировоззрения, соответствующего современному уровню развития информационных технологий.

#### ***Метапредметные результаты:***

По окончании программы обучающийся:

– научиться самостоятельно ставить и формулировать для себя новые задачи, планировать свои действия, планировать пути решения поставленной задачи для получения эффективного результата, корректировать свои действия;

– получит технические навыки и развитое логическое мышление;

– получит навыки исследовательской и проектной деятельности;

– научиться работать с различными источниками информации, извлекать нужную информацию из открытых источников;

– усвоит правила индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

#### **4.2. Планируемые результаты «Продвинутого уровня»(второй год обучения)**

##### ***Предметные результаты:***

По окончании программы обучающийся:

– получит опыт разработки и использования компьютерно-математических моделей, а также умение интерпретировать и анализировать

полученные результаты на предмет их соответствия или несоответствия реальному объекту или процессу;

- научиться выполнять созданные программы, осуществлять их разработку, тестирование и отладку, используя изученный язык программирования;

- получит опыт использования базовых управляющих конструкций объектно-ориентированного программирования;

- научиться самостоятельно создавать программы анализа данных на выбранном для этого универсальном языке программирования;

- получит навык работы со специальными средствами и библиотеками языка

Python;

- научиться определять результат выполнения алгоритма при заданных исходных данных.

***Личностные результаты:***

По окончании программы обучающийся:

- овладеет способностью к саморазвитию и самообразованию, будут развиты любознательность, внимательность и настойчивость при выполнении заданий практического характера;

- научиться организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности;

- будет сформировано ответственное отношение к учению, готовность и способность обучающихся к саморазвитию и самообразованию, средствами информационных технологий

- овладеет коммуникативной компетентностью в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности.

- овладеет базой целостного мировоззрения, соответствующего современному уровню развития информационных технологий.

***Метапредметные результаты:***

По окончании программы обучающийся:

- научиться самостоятельно ставить и формулировать для себя новые задачи, планировать свои действия, планировать пути решения поставленной задачи для получения эффективного результата, корректировать свои действия;
- получит технические навыки и развитое логическое мышление;
- получит навыки исследовательской и проектной деятельности;
- научиться работать с различными источниками информации, извлекать нужную информацию из открытых источников;
- усвоит правила индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

# I. Комплекс организационно-педагогических условий реализации общеразвивающей программы

## 1. Календарный учебный график на 2021-2022 учебный год

Таблица 3

№ п/п	Основные характеристики образовательного процесса	
1.	Количество учебных недель	36
2.	Количество учебных дней	72
3.	Количество часов в неделю	4
4.	Количество часов 1-ый год обучения	144
5.	Количество часов 2-ой год обучения	144
6.	Недель в I полугодии 1-ый год обучения	13
7.	Недель в I полугодии 2-ой год обучения	16
8.	Недель во II полугодии 1-ый год обучения	23
9.	Недель во II полугодии 2-ой год обучения	20
10.	Начало занятий 1-ый год обучения	4 октября
11.	Начало занятий 2-ой год обучения	6 сентября
12.	Выходные дни	1 января – 9 января
13.	Окончание учебного года	31 мая

## 2. Условия реализации программы

### *Материально-техническое обеспечение*

#### *Требования к помещению:*

- помещение для занятий, отвечающие требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочим местом для педагога.

#### *Оборудование:*

- Доска интерактивная;
- Клавиатура Logitech;
- Кулер для воды;
- Монитор Samsung 23.5;

- Мышь компьютерная Logitech;
- Наушники Pioneer;
- Ноутбук Lenovo ThinkPad P590;
- Потолочный кронштейн;
- Принтер Canon MF742CDw;
- Стойка для интерактивной доски;
- Телевизор Samsung 65;
- Тележка для ноутбуков;
- Тепловентилятор Polaris;
- Яндекс.Станция;
- HDMI-разветвитель (на 4 выхода).

*Расходные материалы:*

- whiteboard маркеры;
- бумага писчая;
- шариковые ручки;
- permanent маркеры.

***Информационное обеспечение:***

Операционная система Windows 7,8,10; поддерживаемые браузеры (для работы LMS): Yandex Browser, Chrome, Chrome Mobile, Firefox, Opera, Safari, Mobile Safari, Edge, Python; среда Wing IDE 101 (версии 6 или выше); среда PyCharm Community Edition; пакет PyQt4 (на Qt5); пакет библиотек со SciPy: numpy, scipy, matplotlib, ipython + ipythonnotebook, sympy, pandas; рекомендуется установить ПО Anaconda.

***Кадровое обеспечение:***

Программа реализуется Наумкиным С.В., педагогом дополнительного образования.

Реализовывать программу могут педагоги дополнительного образования, обладающие достаточными знаниями в области педагогики, психологии и методологии, знающие особенности технологии обучения основам программирования на языке Python.

### **3. Формы аттестации и оценочные материалы**

Система контроля знаний и умений учащихся представляется в виде учёта результатов по итогам выполнения практических заданий, демонстрации готовых проектов и посредством наблюдения, отслеживания динамики развития учащегося.

Система отслеживания результатов обучающихся выстроена следующим образом:

- определение начального уровня знаний, умений и навыков;
- промежуточный контроль;
- итоговый контроль.

Входным контролем при приёме на первый год обучения по данной общеразвивающей программе является успешное предварительное тестирование. Тестирование разрабатывается школой анализа данных Яндекса (Яндекс.Лицей).

Оценивая личностные качества воспитанников, педагог проводит наблюдение за обучающимися, отслеживание динамики изменения их творческих, коммуникативных и иных способностей, личностных качеств обучающихся (Приложение 1).

Система промежуточного и итогового контроля знаний и умений обучающихся представляется в виде учёта индивидуального результата по каждому контрольному мероприятию и подведения в итоге суммарного балла для каждого обучающегося.

Текущий контроль осуществляется регулярно в течение учебного года. Контроль теоретических знаний осуществляется с помощью педагогического наблюдения, тестов, опросов. В практической деятельности результативность оценивается качеством выполнения работ учащихся, где анализируются положительные и отрицательные стороны работ, корректируются недостатки.

Промежуточная аттестация проводится в форме учета результатов по итогам выполнения заданий отдельных модулей (Приложения 3, 4, 5, 6, 7) и посредством наблюдения (Приложения 2).

Оценка освоения программы в конце модуля осуществляется по 40-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице.

#### **Уровень освоения программы по модулю**

Таблица 4

<b>Баллы</b>	<b>Процент освоения программы</b>	<b>Уровень освоения</b>
1 – 15	0 – 30%	Низкий
16 – 34	31 – 69%	Средний
35 – 40	70 – 100%	Высокий

В конце каждого учебного года, обучающиеся проходят защиту индивидуальных/групповых проектов. Индивидуальный/групповой проект оценивается формируемой комиссией. Состав комиссии (не менее 3-х человек): педагог (в обязательном порядке), администрация учебной организации, приветствуется привлечение IT-профессионалов, представителей высших и других учебных заведений. Решение принимается коллегиально. Для оценки проекта членам комиссии рекомендуется использовать бланк оценки проектов (Приложение 8).

Итоговая аттестация осуществляется в форме презентации, защиты проекта и оценивается по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице:

#### **Уровень освоения программы по окончании обучения**

Таблица

<b>Баллы, набранные обучающимся</b>	<b>Уровень освоения программы</b>
1–39 баллов	Низкий
40–79 баллов	Средний
80–100 баллов	Высокий

Формы проведения итогов по каждой теме и каждому разделу общеразвивающей программы соответствуют целям и задачам ДООП.

## **4. Методические материалы**



Образовательный процесс осуществляется в очной форме с применением электронного обучения и дистанционных образовательных технологий.

В образовательном процессе используются следующие методы:

- объяснительно-иллюстративный;
- метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой);
- проектно-исследовательский;
- наглядный:
- демонстрация плакатов, схем, таблиц, диаграмм;
- использование технических средств;
- просмотр кино- и телепрограмм;
- практический:
- практические задания;
- анализ и решение проблемных ситуаций и т. д.

Выбор методов обучения осуществляется исходя из анализа уровня готовности обучающихся к освоению содержания модуля, степени сложности материала, типа учебного занятия. В связи с тем, что в работу по программе включаются подростки, уже имеющие первичные навыки программирования, поэтому вместе с объяснительно-иллюстративным используются, а также становятся ведущими частично-поисковый и проектно-исследовательский методы, на продвинутом уровне (2-ой год обучения) – проблемного изложения и проектно-исследовательский. На выбор методов обучения значительно влияет персональный состав группы, индивидуальные особенности, возможности и запросы детей.

Образовательный процесс строится на следующих **принципах**:

– **Принцип научности.** Его сущность состоит в том, чтобы ребёнок усваивал реальные знания, правильно отражающие действительность, составляющие основу соответствующих научных понятий.

**–Принцип наглядности.** Наглядные образы способствуют правильной организации мыслительной деятельности ребёнка. Наглядность обеспечивает понимание, прочное запоминание.

**–Принцип доступности,** учёта возрастных и индивидуальных особенностей детей в процессе обучения по программе. Предполагает соотнесение содержания, характера и объёма учебного материала с уровнем развития, подготовленности детей. Переходить от лёгкого к трудному, от известного к неизвестному. Но доступность не отождествляется с лёгкостью. Обучение, оставаясь доступным, сопряжено с приложением серьёзных усилий, что приводит к развитию личности.

**–Принцип осознания процесса обучения.** Данный принцип предполагает необходимость развития у ребёнка рефлексивной позиции: как я узнал новое, как думал раньше. Если ребёнок видит свои достижения, это укрепляет в нём веру в собственные возможности, побуждает к новым усилиям. И если ребёнок понимает, в чём и почему он ошибся, что ещё не получается, то он делает первый шаг на путь самовоспитанию.

**–Принцип воспитывающего обучения.** Обучающая деятельность педагога, как правило, носит воспитывающий характер. Содержание обучения, формы его организации, методы и средства оказывают влияние на формирование личности в целом.

#### **Формы обучения:**

– **фронтальная** – предполагает работу педагога сразу со всеми обучающимися в едином темпе и с общими задачами. Для реализации обучения используется компьютер педагога с мультимедиа проектором, посредством которых учебный материал демонстрируется на общий экран. Активно используются Интернет-ресурсы;

– **коллективная** – это форма сотрудничества, при котором коллектив обучает каждого своего члена и каждый член коллектива активно участвует в обучении своих товарищей по совместной учебной работе;

– **групповая** – предполагает, что занятия проводятся с подгруппой.

Для этой группы распределяется на подгруппы не более 6 человек, работа в которых регулируется педагогом;

– **индивидуальная** – подразумевает взаимодействие преподавателя с одним обучающимся. Как правило данная форма используется в сочетании с фронтальной. Часть занятия (объяснение новой темы) проводится фронтально, затем обучающийся выполняют индивидуальные задания или общие задания в индивидуальном темпе;

– **дистанционная** – взаимодействие педагога и обучающихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты. Для реализации дистанционной формы обучения весь дидактический материал размещается в свободном доступе в сети Интернет, происходит свободное общение педагога и обучающихся в социальных сетях, по электронной почте, посредством видеоконференции или в общем чате. Кроме того, дистанционное обучение позволяет проводить консультации обучающегося при самостоятельной работе дома.

#### ***Формы организации учебного занятия:***

В образовательном процессе помимо традиционного учебного занятия используются многообразные формы, которые несут учебную нагрузку и могут использоваться как активные способы освоения детьми образовательной программы, в соответствии с возрастом обучающихся, составом группы, содержанием учебного модуля: беседа, лекция, мастер-класс, практическое занятие, защита проектов.

***Методы воспитания:*** мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

***Педагогические технологии:*** индивидуализации обучения; группового обучения; коллективного взаимообучения; дифференцированного обучения; разноуровневого обучения; проблемного обучения; развивающего обучения; дистанционного обучения; игровой деятельности; коммуникативная технология обучения; коллективной творческой деятельности; решения изобретательских задач;

здоровьесберегающая технология.

***Методическое обеспечение:***

Методические пособия, разработанные преподавателем с учётом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности.

Электронное приложение к учебникам К. Ю. Полякова Информатика и ИКТ. Набор цифровых образовательных ресурсов – дидактические материалы, интерактивные тесты, анимационные плакаты.

**Список литературы**

***Нормативные документы:***

1. Федеральный закон «Об образовании в Российской Федерации» от 29.12.2012 N 273-ФЗ;
2. Федеральный закон от «Об основах охраны здоровья граждан в Российской Федерации», 2011г.;
3. Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребёнка в РФ»;
4. Распоряжение правительства РФ от 04.09. 2014 № 1726-р «Об утверждении Концепции развития дополнительного образования детей»;
5. Распоряжение Правительства Российской Федерации от 29 мая 2015 г. № 996-р. «Стратегия развития воспитания в Российской Федерации на период до 2025года»;
6. Приказ Министерства просвещения России от 09.11.2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
7. Приказ Министерства Просвещения Российской Федерации от 03.09.2019 г. №467 «Об утверждении Целевой модели развития

региональных систем дополнительного образования детей»;

8. Постановление Главного государственного санитарного врача РФ от 28.09.2020 г. № 28 «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

9. Письмо Министерства образования и науки РФ от 18.11.2015г. № 09-3242 «О направлении Методических рекомендаций по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;

#### **Список литературы, использованной при написании программы:**

1. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ.Лаборатория знаний, 2006;

2. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014;

3. Окулов С. М. Основы программирования. М.: Бином. Лаборатория знаний, 2012.;

4. Поляков К. Ю., Еремин Е. А. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.;

5. Эльконин, Д. Б. Детская психология: учеб. пособие для студ. высш. учеб. заведений / Д. Б. Эльконин; ред.сост. Б. Д. Эльконин. – 4-е изд., стер. – М.: Издательский центр «Академия», 2007. – 384 с;

6. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.

#### ***Литература, рекомендованная обучающимся:***

1. Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М.: Манн, Иванови Фербер, 2017 — 320 с.;

2. К. Вордерман и др. Программирование для детей: Иллюстрированное руководство по языкам Scratch и Python, 224 стр.

Издательство: Манн, Иванов и Фербер, 2017 г.;

3. Б. Пэйн. Python для детей и родителей, 352 стр. Издательство: Эксмо, 2017.

*Электронные ресурсы:*

1. Материалы и презентации к урокам в LMS Яндекс.Лицея;
2. Сайт «Python 3 для начинающих» – [pythonworld.ru](http://pythonworld.ru);
3. Сайт «Питонтьютор» – [pythontutor.ru](http://pythontutor.ru);
4. Лекции А. В. Умнова, прочитанные в Школе анализа данных Яндекса –

<https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh5OpdwBl>.

## Бланк наблюдения за динамикой личностного развития обучающихся

№ Группы \_\_\_\_\_

Дата \_\_\_\_\_

№ п/п	ФИО	ПОКАЗАТЕЛИ КАЧЕСТВА ЛИЧНОСТНОГО РАЗВИТИЯ ОБУЧАЮЩИХСЯ					Итого
		Во время занятий проявляет устойчивый интерес и инициативу при освоении программы	Использует в общении базовую систему понятий	Проявляет инициативу и интерес к проектной деятельности, активно включается в групповую работу	Активно сотрудничает со сверстниками, уважительно относится к мнению окружающих	Аккуратно относится к материально- техническим ценностям	
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

3 балла – качество проявляется систематически

2 балла – качество проявляется ситуативно

1 балл – качество не проявляется

**Лист оценки работы обучающихся в процессе разработки**  
(максимум – 20 баллов)

№ группы: \_\_\_\_\_

Дата: \_\_\_\_\_

№ п/п	ФИ обучающегося	Сложность языка разработки (по шкале от 0 до 5 баллов)	Соответствие написания программного кода поставленной задаче (по шкале от 0 до 5 баллов)	Презентация модели по плану. Степень владения специальными терминами (по шкале от 0 до 5 баллов)	Степень увлеченности процессом и стремления к оригинальности при выполнении заданий (по шкале от 0 до 5 баллов)	Кол-во вопросов и затруднений (шт. за одно занятие)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						



**Пример контрольной работы по модулю «1. Введение в программирование.»**  
**(стартовый уровень)**  
 (максимум – 40 баллов)

№	Вопросы	Верный ответ
1.	<p><b>Что выведет следующий фрагмент кода?</b>  <code>x = 4.5</code>  <code>y = 2 print(x // y)</code>  <b>Варианты ответов:</b>                      1) 2.0                      2) 2.25                      3) 9.0                      4) 20.25                      5) 21</p>	2.0
2.	<p><b>Что выведет следующий код, при его исполнении?</b>  <b>Используется Python 3.x.</b>  <code>print(type(1 / 2))</code> <b>Варианты ответов</b>  <code>class 'int'</code>  <code>class 'number'</code>  <code>class 'float'</code>  <code>class 'double'</code>  <code>class 'tuple'</code></p>	class 'float'
3.	<p><b>Что будет напечатано?</b>  <code>kvps = {"user", "bill", "password", "hillary"}</code>  <code>print(kvps['password'])</code>  <b>Варианты ответов:</b>                      1) user                      2) bill                      3) password                      4) hillary                      5) Ничего. TypeError.</p>	Ничего. TypeError.
4.	<p><b>Что будет напечатано?</b>  <code>name = "snow storm" print("%s" % name[6:8])</code> <b>Варианты ответов:</b>                      1) st                      2) sto                      3) to                      4) Syntax Error</p>	to

5.	<p><b>Что напечатает следующий код:</b>  word = 'foobar' print(word[3:] + word[:3])</p> <p><b>Варианты ответов:</b>  1) foobar  2) obarof  3) barfoo  4) SyntaxError</p>	barfoo
6.	<p><b>Что выведет следующая программа?</b>  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)</p> <p><b>Варианты ответов:</b> 1) 1  2) 2  3) 3  4) 4</p>	3

7	<p><b>Что выведет следующая программа?</b></p> <pre>a = [1,2,3,None(),[],] print(len(a))</pre> <p><b>Варианты ответов</b></p> <ol style="list-style-type: none"> <li>1) 4</li> <li>2) 5</li> <li>3) 6</li> <li>4) 7</li> </ol>	6
8	<p><b>Имеем следующую последовательность действий,</b></p> <p><b>чему равна переменная L2 ?:</b></p> <pre>&gt;&gt;&gt; L1 = [2, 3, 4] &gt;&gt;&gt; L2 = L1 &gt;&gt;&gt; L1[0] = 24 &gt;&gt;&gt; L1 [24, 3, 4] &gt;&gt;&gt; L2</pre> <p><b>Варианты ответов:</b></p> <ol style="list-style-type: none"> <li>1) [2,3,4]</li> <li>2) [24,3,4]</li> <li>3) [2,3,24]</li> <li>4) [3,4,2]</li> </ol>	[24,3,4]
9	<p><b>Что покажет этот код?</b></p> <pre>for i in range(5):     if i % 2 == 0:         continue     print(i)</pre> <p><b>Варианты ответов:</b></p> <ol style="list-style-type: none"> <li>1) Ошибку, так как i не присвоена</li> <li>2) Ошибку из-за неверного вывода</li> <li>3) Числа: 1, 3 и 5</li> <li>4) Числа: 0, 2 и 4</li> <li>5) Числа: 1 и 3</li> </ol>	Числа: 1 и 3

10.	<p><b>Что покажет этот код?</b></p> <pre>for j in 'Hi! I\'m mister     Robert':if j == "\":     print     ("Найдено")     break else:     print ("Готово")</pre> <p><b>Варианты ответов:</b></p> <ol style="list-style-type: none"><li>1) Ошибку в коде</li><li>2) "Найдено" и "Готово"</li><li>3) "Готово"</li><li>4) "Найдено"</li></ol>	"Найдено"
-----	--	-----------

**Пример промежуточных проектных работ по образовательным  
модулям(базовый уровень)**

Работа над персональным проектом — предполагает построение цели, задач для ее достижения, и выполнение по персональному ТЗ.

Предлагаемые темы работ:

1. Разработка игры «Шахматы» для двух игроков с консольным интерфейсом
2. Разработка игры «Змейка» с графическим интерфейсом (Tkinter)
3. Разработка игры «Футбол» с графическим интерфейсом
4. Создание приложения для автоматической генерации писем и документов по шаблону MS Word
5. Создание консольного графического редактора (применение фильтров, изменение размеров, повороты изображения и пр.)

**Пример промежуточных проектных работ по образовательным  
модулям(продвинутый уровень)**

Работа над персональным проектом — предполагает построение цели, задач для ее достижения, и выполнение по персональному ТЗ.

Предлагаемые темы работ:

1. Разработка приложения «Текстовый редактор»
2. Разработка приложения «Графический редактор»
3. Разработка игры «Гексагон» с графическим интерфейсом
4. Создание чат бота
5. Разработка игры для трех игроков с возможностью добавления ботов
6. Создание WEB-проекта, предоставляющего для команд пользователей совместное рабочее пространство

**Лист оценивания промежуточных проектных работ обучающихся  
(базовый уровень)**

(максимум – 20 баллов)

№ группы: \_\_\_\_\_

Дата: \_\_\_\_\_

№ п/п	ФИО обучающегося	Соответствие теме (по шкале от 0 до 5 баллов)	Сложность алгоритмов, структур данных (по шкале от 0 до 5 баллов)	Качество реализации (по шкале от 0 до 5 баллов)	Степень владения материалом (по шкале от 0 до 5 баллов)	Презентация (шт. за одно занятие)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						

**Лист оценивания промежуточных проектных работ обучающихся  
(продвинутый уровень)  
(максимум – 20 баллов)**

№ группы: \_\_\_\_\_

Дата: \_\_\_\_\_

№ п/п	ФИО обучающегося	Соответствие теме (по шкале от 0 до 5 баллов)	Сложность функционала (по шкале от 0 до 5 баллов)	Качество кода (по шкале от 0 до 5 баллов)	Удобство использования (шт. за одно занятие)	Презентация (шт. за одно занятие)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						



**Бланк оценки итоговых  
индивидуальных/групповых проектов  
(максимум – 20 баллов)**

№ Группы \_\_\_\_\_

Дата \_\_\_\_\_

№ п/п	ФИО	Название проекта	Актуальность проекта (0–3 балла)	Постановка проблемы (0–3балла)	Целеполагание (0–3 балла)	Качество результата (0–3 балла)	Практическая реализация (0–3 балла)	Оригинальность и творческий подход (0-2 балла)	Защита проекта (представление проекта, Работоспособность ) (0-3 балла)	ИТОГО
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										

\_\_\_\_\_ /  
подпись

\_\_\_\_\_ /  
расшифровка

