

Государственное бюджетное общеобразовательное учреждение
гимназия №406 Пушкинского района Санкт-Петербурга

ПРИНЯТА

Решением Педагогического совета
ГБОУ гимназии № 406 Пушкинского
района Санкт-Петербурга
Протокол от 30.08.2024 № 1

УТВЕРЖДЕНА

Приказом директора от 30.08.2024 № 125
ГБОУ гимназии № 406 Пушкинского
района Санкт-Петербурга

_____ В.В. Штерн

Дополнительная общеразвивающая программа

«Программирование»

Срок освоения: 2 года

Возраст обучающихся: 13 - 17 лет

Разработчик: Пенин Андрей Семенович,
педагог дополнительного образования

Пояснительная записка

Направленность программы: техническая

Адресат программы: 13 - 17 лет.

Актуальность программы: соответствует государственной политике в области дополнительного образования. Язык программирования Python является современным объектно-ориентированным языком, который в настоящее время получил большое распространение благодаря своей направленности на создание кроссплатформенных, сетевых программных приложений, потому востребован среди подростков и их родителей.

Уровень освоения: общекультурный.

Объем и срок освоения программы: 432 часа, 2 года.

Цель программы: Закрепление теоретических знаний в области структурного программирования, расширение практических навыков программирования на языке Python с использованием стандартных средств разработки..

Задачи программы.

Обучающие:

- обучить структуре и принципам функционирования системы программирования;
- приобрести навык проектирования и создания компьютерной программы;
- обучить различным способам проектирования современных ОС;
- приобрести навык пользования основными объектами пользовательского интерфейса;
- познакомить с возможностями объектно-ориентированного языка Python.

Развивающие:

- формирование и развитие информационной культуры: умения работать с разными источниками;
- формирование исследовательских умений, умения общаться, умения взаимодействовать, умения доводить дело до конца;
- развитие памяти, внимательности и наблюдательности, творческого воображения;
- развитие информационной культуры за счет освоения информационных и коммуникационных технологий;
- развитие стратегического мышления;

Воспитательные:

- сформировать гражданскую позицию, патриотизм и обозначить ценность инженерного образования;
- воспитать чувство товарищества, чувство личной ответственности во время подготовки и защиты проекта, демонстрации моделей объектов;
- сформировать навыки командной работы над проектом;

Планируемые результаты.

Личностные:

- выражено умение работать в команде: работа в общем ритме, эффективное распределение задач и др.;
- сформировано умение ориентироваться в информационном пространстве, продуктивно использовать техническую литературу для поиска сложных решений;
- проявление технического мышления, познавательной деятельности, творческой инициативы, самостоятельности;
- сформировано способность творчески решать технические задачи;
- выражена готовность и способность применения теоретических знаний по физике, информатике для решения задач в реальном мире;
- сформирована способность правильно организовывать рабочее место и время для достижения поставленных целей.

Предметные:

- приобретено знание основ и принципов теории решения изобретательских задач, овладение начальными базовыми навыками инженерии;

- приобретено знание и понимание принципов проектирования в САПР, основ создания и проектирования 2D и 3D моделей;
- приобретено знание пользовательского интерфейса профильного ПО, базовых объектов инструментария.
- приобретены навыки проектной деятельности, навыки планирования работ и постановки задач;

Метапредметные:

- приобретен опыт поиска и преобразования оценки информации, необходимых качеств мышления, характерных для работы с большими данными.;
- сформировано умение ориентироваться в информационном пространстве, продуктивно использовать техническую литературу для поиска сложных решений;
- проявление технического мышления, познавательной деятельности, творческой инициативы, самостоятельности;

Организационно-педагогические условия реализации программы.

Язык обучения: русский

Форма обучения: очная / частично дистанционная.

Особенности реализации: Программа входит в блок программ Курчатовского класса. Реализация данной программы предполагает следующие формы обучения - очную, а также комбинированную очно – дистанционную. При реализации дистанционного обучения педагог использует образовательные платформы: ZOOM.... Сетевые ресурсы: SKYPE, Facebook, WS, VK. (см. раздел оценочные и методические материалы.).

Условия набора и формирования групп: Разновозрастная группа. Программа предусматривает отбор мотивированных детей для участия в соревнованиях регионального и более высокого уровня. Программа не адаптирована для обучающихся с ОВЗ.

Формы организации и проведения занятий: очно-заочная, коллективная, групповая, фронтальная.

Материально-техническое оснащение: специализированный кабинет, компьютеры с выходом в Интернет ,раздаточный материал по теме занятия, включающий схемы, плакаты, сборники дидактических материалов, препараты.

Кадровое обеспечение: дополнительный узкопрофильный специалист не предусмотрен

Учебный план 1 год

№ п/п	Название раздела, темы	Количество часов			Формы аттестации, контроля
		всего	теория	практика	
1	Вводное занятие. Установка Версии Python (2 и 3). Установка Python. Установка Anaconda. Установка IDE PyCharm. Проверка работоспособности	4	2	2	Беседа по основным понятиям темы
2	Запуск программ на Python	12	2	10	Установка.зачет
3	Типы и модель данных	12	2	10	
4	Арифметические операции	12	2	10	Зачет
5	Условные операторы и циклы	12	2	10	
6	Работа с IPython и Jupyter Notebook	22	2	20	
7	Работа со списками (list)	12	2	10	Зачет
8	Кортежи (tuple)	12	2	10	

9	Словари (dict)	16	2	14	
10	Функции в Python	12	2	10	
11	Работа с исключениями	12	2	10	Зачет
12	Ввод-вывод данных. Работа с файлами	12	2	10	
13	Модули и пакеты	12	2	10	Зачет
14	Классы и объекты	12	2	10	Зачет
15	Итераторы и генераторы	12	2	10	Зачет
16	Установка пакетов в Python	12	2	10	
17	Виртуальные окружения	12	2	10	
18	Аннотация типов в Python		2	2	
	Итоговое занятие.	6	1		
	Итого	216	40	176	

Учебный план 2 год

№ п/п	Название раздела, темы	Количество часов			Формы аттестации, контроля
		всего	теория	практика	
1	Вводное занятие. Техника безопасности. Словари (dict)	4	2	2	Зачет
2	Функции в Python	22	2	20	
3	Работа с исключениями	26	6	20	
4	Ввод-вывод данных. Работа с документами.	26	6	20	Зачет
5	Модули и пакеты	22	2	20	
6	Классы и объекты	22	2	20	
7	Итераторы и генераторы	22	2	20	
8	Установка пакетов в Python	26	6	20	
9	Виртуальные окружения	22	2	20	Зачет
10	Аннотация типов в Python	22	2	20	Зачет
11	Итоговое занятие	6	2		
	ИТОГО	216	34	182	

Рабочая программа 1 год

Задачи программы

Обучающие:

- приобрести опыт взаимодействия со структурой и принципами функционирования системы программирования;
- приобрести опыт проектирования и создания компьютерной программы; познакомить с различными способами классификации и принципами проектирования современных ОС;
- приобрести опыт взаимодействия с основными объектами пользовательского интерфейса;
- приобрести опыт взаимодействия с основными понятиями, видами и характеристиками современного программного обеспечения технологии Python;
- приобрести опыт взаимодействия с возможностями объектно-ориентированного языка Python.

Развивающие:

- формирование и развитие информационной культуры: умения работать с разными источниками;
- развитие исследовательских умений, умения общаться, умения взаимодействовать, умения доводить дело до конца;
- развитие памяти, внимательности и наблюдательности, творческого воображения;
- развитие информационной культуры за счет освоения информационных и коммуникационных технологий;
- формирование технологической грамотности;
- развитие стратегического мышления;
- получение опыта решения проблем с использованием проектных технологий.

Воспитательные:

- сформировать гражданскую позицию, патриотизм и обозначить ценность инженерного образования;
- воспитать чувство товарищества, чувство личной ответственности во время подготовки и защиты проекта, демонстрации моделей объектов;
- сформировать навыки командной работы над проектом;
- Ориентировать учащихся на получение технической инженерной специальности;
- научить работать с информационными объектами и различными источниками информации;
- приобрести межличностные и социальные навыки, а также навыки общения.

Содержание программы 1 год

Тема 1. Вводное занятие. Установка

Версии Python (2 и 3). Установка Python. Установка Anaconda. Установка IDE PyCharm.

Проверка работоспособности

Тема 2. Запуск программ на Python

В этом уроке мы рассмотрим два основных подхода к работе с интерпретатором Python – это непосредственная интерпретация строк кода, вводимых с клавиатуры в интерактивном режиме и выполнение файлов с исходным кодом в пакетном режиме. Также коснемся некоторых особенностей работы с Python в Linux и MS Windows.

Тема 3. Типы и модель данных

В данном уроке разберем как Python работает с переменными и определим какие типы данных можно использовать в рамках этого языка. Подробно рассмотрим модель данных Python, а также механизмы создания и изменения значения переменных.

Тема 4. Арифметические операции

Язык Python, благодаря наличию огромного количества библиотек для решения разного рода вычислительных задач, сегодня является конкурентом таким пакетам как Matlab и Octave. Запущенный в интерактивном режиме, он, фактически, превращается в мощный калькулятор. В этом уроке речь пойдет об арифметических операциях, доступных в данном языке.

Арифметические операции будем изучать применительно к числам, причем работу с комплексными числами разберем отдельно. Также, кратко остановимся на битовых операциях, представлении чисел в разных системах исчисления и коснемся библиотеки math.

Тема 5. Условные операторы и циклы

В этом уроке рассмотрим оператор ветвления `if` и операторы цикла `while` и `for`. Основная цель – это дать общее представление об этих операторах и на простых примерах показать базовые принципы работы с ними.

Тема 6. Работа с IPython и Jupyter Notebook

IPython представляет собой мощный инструмент для работы с языком Python. Базовые компоненты IPython – это интерактивная оболочка для широкого набора возможностей и ядро для Jupyter. Jupyter notebook является графической веб-оболочкой для IPython, которая расширяет идею консольного подхода к интерактивным вычислениям.

Основные отличительные особенности данной платформы – это комплексная интроспекция объектов, сохранение истории ввода на протяжении всех сеансов, кэширование выходных результатов, расширяемая система “магических” команд, логирование сессии, дополнительный командный синтаксис, подсветка кода, доступ к системной оболочке, стыковка с `pdb` отладчиком и Python профайлером.

IPython позволяет подключаться множеству клиентов к одному вычислительному ядру и, благодаря своей архитектуре, может работать в параллельном кластере.

В Jupyter notebook можно разрабатывать, документировать и выполнять приложения на языке Python, он состоит из двух компонентов: веб-приложение, запускаемое в браузере, и ноутбуки – файлы, в которых можно работать с исходным кодом программы, запускать его, вводить и выводить данные и т.п.

Тема 7. Работа со списками (list)

Одна из ключевых особенностей Python, благодаря которой он является таким популярным – это простота. Особенно подкупает простота работы с различными структурами данных – списками, кортежами, словарями и множествами. Сегодня мы рассмотрим работу со списками.

Тема 8. Кортежи (tuple)

Данный урок посвящен кортежам (tuple) в Python. Основное внимание уделено вопросу использования кортежей, почему иногда лучше применять их, а не списки, рассмотрены способы создания и основные приемы работы с кортежами. Также затронем тему преобразования кортежа в список и обратно.

Тема 9. Словари (dict)

Словари (dict) – это одна из наиболее часто используемых структур данных, позволяющая хранить объекты, для доступа к которым используется ключ. В этом уроке будут рассмотрены операции создания, удаления, работы со словарями и их методы.

Тема 10. Функции в Python

Урок посвящен созданию функций в Python и работе с ними (передача аргументов, возврат значения и т.п.). Также рассмотрены lambda-функций, их особенности и использование.

Тема 11. Работа с исключениями

Данный урок посвящен исключениям и работе с ними. Основное внимание уделено понятию исключения в языках программирования, обработке исключений в Python, их генерации и созданию пользовательских исключений.

Тема 12. Ввод-вывод данных. Работа с файлами

В уроке рассмотрены основные способы ввода и вывода данных в Python с использованием консоли, и работа с файлами: открытие, закрытие, чтение и запись.

Тема 13. Модули и пакеты

Модули и пакеты значительно упрощают работу программиста. Классы, объекты, функции и константы, которыми приходится часто пользоваться можно упаковать в модуль, и, в дальнейшем, загружать его в свои программы при необходимости. Пакеты позволяют формировать пространства имен для работы с модулями.

Тема 14. Классы и объекты

Данный урок посвящен объектно-ориентированному программированию в Python. Разобраны такие темы как создание объектов и классов, работа с конструктором, наследование и полиморфизм в Python.

Тема 15. Итераторы и генераторы

Генераторы и итераторы представляют собой инструменты, которые, как правило, используются для поточной обработки данных. В уроке рассмотрим концепцию итераторов в Python, научимся создавать свои итераторы и разберемся как работать с генераторами.

Тема 16. Установка пакетов в Python

В процессе разработки программного обеспечения на *Python* часто возникает необходимость воспользоваться пакетом, который в данный момент отсутствует на вашем компьютере. О том, откуда взять нужный вам пакет и как его установить, вы узнаете из этого урока.

Тема 17. Виртуальные окружения

На данном уроке мы изучим, что такое виртуальные окружения в *Python*, зачем они нужны и как их использовать. Познакомимся с инструментами `virtualenv` и `venv` для создания виртуальных окружений.

Тема 18. Аннотация типов в Python

Этот урок посвящен аннотациям типов в *Python*. Рассмотрен вопрос контроля типов переменных и функций с использованием комментариев и аннотаций. Приведено описание *PEP*'ов, регламентирующих работу с аннотациями, и представлены примеры работы с инструментом *myru* для анализа *Python* кода.

Планируемые результаты:

Обучающие:

- познакомить со структурой и принципами функционирования системы программирования;
- познакомить процессом проектирования и создания компьютерной программы; познакомить с различными способами классификации и принципами проектирования современных ОС;
- познакомить с основными объектами пользовательского интерфейса;
- познакомить с основными понятиями, видами и характеристиками современного программного обеспечения технологии Python;

Развивающие:

- формирование и развитие информационной культуры: умения работать с разными источниками;
- развитие исследовательских умений, умения общаться, умения взаимодействовать, умения доводить дело до конца;
- развитие памяти, внимательности и наблюдательности, творческого воображения;
- развитие информационной культуры за счет освоения информационных и коммуникационных технологий;
- формирование технологической грамотности;
- развитие стратегического мышления;
- получение опыта решения проблем с использованием проектных технологий.

Воспитательные:

- сформировать гражданскую позицию, патриотизм и обозначить ценность инженерного образования;
- воспитать чувство товарищества, чувство личной ответственности во время подготовки и защиты проекта, демонстрации моделей объектов;
- сформировать навыки командной работы над проектом;
- Сориентировать учащихся на получение технической инженерной специальности;
- научить работать с информационными объектами и различными источниками информации;

Рабочая программа 2 год

Задачи программы:

Обучающие:

- приобрести навык пользования с основными объектами пользовательского интерфейса;
- научить пользоваться возможностями объектно-ориентированного языка Python.

Развивающие:

- формирование и развитие информационной культуры: умения работать с разными источниками;
- развитие исследовательских умений, умения общаться, умения взаимодействовать, умения доводить дело до конца;
- получение опыта решения проблем с использованием проектных технологий.

Воспитательные:

- сформировать гражданскую позицию, патриотизм и обозначить ценность инженерного образования;
- воспитать чувство товарищества, чувство личной ответственности во время подготовки и защиты проекта, демонстрации моделей объектов;
- сформировать навыки командной работы над проектом;

Содержание программы 2 год

Тема 1. Вводное занятие. Техника безопасности. Словари (dict)

Словари (dict) – это одна из наиболее часто используемых структур данных, позволяющая хранить объекты, для доступа к которым используется ключ. В этом уроке будут рассмотрены операции создания, удаления, работы со словарями и их методы.

Тема 2. Функции в Python

Урок посвящен созданию функций в Python и работе с ними (передача аргументов, возврат значения и т.п.). Также рассмотрены lambda-функций, их особенности и использование.

Тема 3. Работа с исключениями

Данный урок посвящен исключениям и работе с ними. Основное внимание уделено понятию исключения в языках программирования, обработке исключений в Python, их генерации и созданию пользовательских исключений.

Тема 4. Ввод-вывод данных. Работа с файлами

В уроке рассмотрены основные способы ввода и вывода данных в Python с использованием консоли, и работа с файлами: открытие, закрытие, чтение и запись.

Тема 5 Модули и пакеты

Модули и пакеты значительно упрощают работу программиста. Классы, объекты, функции и константы, которыми приходится часто пользоваться можно упаковать в модуль, и, в дальнейшем, загружать его в свои программы при необходимости. Пакеты позволяют формировать пространства имен для работы с модулями.

Тема 6. Классы и объекты

Данный урок посвящен объектно-ориентированному программированию в Python. Разобраны такие темы как создание объектов и классов, работа с конструктором, наследование и полиморфизм в Python.

Тема 7. Итераторы и генераторы

Генераторы и итераторы представляют собой инструменты, которые, как правило, используются для поточной обработки данных. В уроке рассмотрим концепцию итераторов в Python, научимся создавать свои итераторы и разберемся как работать с генераторами.

Тема 8. Установка пакетов в Python

В процессе разработки программного обеспечения на Python часто возникает необходимость воспользоваться пакетом, который в данный момент отсутствует на вашем компьютере. О том, откуда взять нужный вам пакет и как его установить, вы узнаете из этого урока.

Тема 9. Виртуальные окружения

На данном уроке мы изучим, что такое виртуальные окружения в Python, зачем они нужны и как их использовать. Познакомимся с инструментами virtualenv и venv для создания виртуальных окружений.

Тема 10. Аннотация типов в Python

Этот урок посвящен аннотациям типов в *Python*. Рассмотрен вопрос контроля типов переменных и функций с использованием комментариев и аннотаций. Приведено описание *PEP*'ов, регламентирующих работу с аннотациями, и представлены примеры работы с инструментом *myru* для анализа *Python* кода.

Планируемые результаты:

Обучающие:

- познакомить со структурой и принципами функционирования системы программирования;
- познакомить процессом проектирования и создания компьютерной программы; познакомить с различными способами классификации и принципами проектирования современных ОС;
- познакомить с основными объектами пользовательского интерфейса;
- познакомить с основными понятиями, видами и характеристиками современного программного обеспечения технологии Python;
- познакомить с возможностями объектно-ориентированного языка Python.

Развивающие:

- формирование и развитие информационной культуры: умения работать с разными источниками;
- развитие исследовательских умений, умения общаться, умения взаимодействовать, умения доводить дело до конца;
- развитие памяти, внимательности и наблюдательности, творческого воображения;
- развитие информационной культуры за счет освоения информационных и коммуникационных технологий;
- формирование технологической грамотности;
- развитие стратегического мышления;
- получение опыта решения проблем с использованием проектных технологий.

Воспитательные:

- сформировать гражданскую позицию, патриотизм и обозначить ценность инженерного образования;
- воспитать чувство товарищества, чувство личной ответственности во время подготовки и защиты проекта, демонстрации моделей объектов;
- сформировать навыки командной работы над проектом;
- Сориентировать учащихся на получение технической инженерной специальности;
- научить работать с информационными объектами и различными источниками информации; приобрести межличностные и социальные навыки, а также навыки общения.

Методическое обеспечение

Критерии дополнительного образования:

- иметь ярко выраженный развивающий характер и основываться на выявлении природных способностей и интересов ребенка;
 - включать разнообразные формы и типы; опираться на существующие сертифицированные программы образования;
 - использовать развивающие методики в работе с детьми;
 - принимать во внимание потребности общества;
1. Методы обучения: словесный, практический, наглядный; эвристический или частично-поисковый, проблемный, исследовательский, репродуктивный, объяснительно-иллюстративный, закрепления материала(беседа, опрос, работа с книгой); методы самостоятельного усвоения материала(работа с учебником: вдумчивое изучение и осмысление материала, лабораторные занятия); методы проверки и оценки (текущая - в ходе учебных занятий, годовая - по итогам года).
 2. Формы занятий: лекции - изложение преподавателем предметной информации; семинары и коллоквиумы - заранее подготовленные сообщения и выступления в группе и их обсуждение; дискуссии - постановка спорных вопросов с целью отработки умения отстаивать и

аргументировать свою точку зрения; конференции - совещания для обсуждения различных тем и выработки решений; экскурсии, научные экспедиции - поездки с ознакомительными и информационными задачами. К нетрадиционным формам учебных занятий можно отнести: презентацию - публичное представление определенной темы или предмета; защиту проекта - обоснование проделанной работы; круглый стол - неформальное обсуждение выбранной тематики; мозговую атаку - коллективное решение нестандартных задач.

3. Типы занятий: комбинированные - изложение материала, проверка изученного; подача нового материала; повторение и усвоение пройденного - анализ полученных результатов; закрепление знаний, умений и навыков - постановка задачи и самостоятельная работа ребенка под руководством педагога; применение полученных знаний и навыков - прикладная деятельность ребенка, использующего на практике приобретенные знания.
4. Технологии занятий: информационно – коммуникационная, развития критического мышления, проектная, развивающего обучения, проблемного обучения, модульная, технология уровневой дифференциации, групповая.
5. Цель оценки достижений воспитанника учреждения дополнительного образования: создание условий для организации деятельности воспитанников, которая обеспечивает реальные возможности разностороннего личностного роста, удовлетворения постоянно изменяющихся индивидуальных потребностей детей и их родителей; для создания психологически комфортной атмосферы для занятий, благоприятных условий для саморазвития, самоопределения детей.
6. Формы подведения итогов: оформление исследования, создание презентации и доклада по изученной теме, публичное выступление перед обучающимися объединения, участие в научно-практических конференциях.
7. Критерии результативности:
 - уровни образованности: элементарная грамотность, функциональная грамотность, компетентность;
 - результативность воспитательной деятельности: разностороннее личностное проявление - положительная динамика проявлений ценностно-значимых качеств личности, обогащение личного опыта социально и личностным содержанием; продуктивность деятельности, выражаемой в соответствующих предметно-практических достижениях (личных, групповых, коллективных).
8. Компоненты личностных достижений воспитанника учреждения дополнительного образования:
 - мотивационно-ценностные (потребность в самореализации, саморазвитии, самосовершенствовании, мотивация достижения, ценностные ориентации);
 - когнитивные (знания, рефлексия деятельности);
 - операциональные (умения, навыки);
 - эмоционально-волевые (уровень притязаний, самооценка, эмоциональное отношение к достижению, волевые усилия).
9. Показатели результативности:
 - низкий уровень: обучающийся имеет частично представление о методах и приемах исследовательской деятельности, не может самостоятельно сформулировать проблему исследования, поставить и описать эксперимент, сделать выводы;
 - средний уровень: обучающийся под руководством учителя может сформулировать проблему исследования, поставить и описать эксперимент, сделать выводы, подвести итог исследовательской работы, оформить реферат;
 - продвинутый: у обучающегося развита мотивация к познанию и творчеству, к получению основных знаний, умений, навыков в области методики и методов научного исследования. Он знает, как самостоятельно формировать и выявлять проблему исследования, как правильно поставить и описать эксперимент, как обеспечить получение надёжных результатов, умеет подвести итоги исследовательской работы, оформить реферат, написать статью и т.д.

10. Технологии отслеживания результатов:

- когнитивный, операциональный компоненты могут быть диагностированы при помощи диагностических исследований результативности освоения программы, содержащихся в самих программах и в методическом обеспечении программы; карты экспертной оценки педагогом компетентности обучающегося; карты самооценки компетентности обучающегося;
- мотивационно-ценностный компонент (потребность в самореализации, саморазвитии, самосовершенствовании, мотивация достижения, ценностные ориентации) - методом наблюдения и анкетирования;

4.Оценочные и методические материалы

4.1.Система текущего контроля и промежуточной аттестации обучающихся

С целью диагностики успешности проектной работы педагог осуществляет текущий контроль в виде scrum- и sprint-сессий. Текущий контроль успеваемости носит безотметочный характер и служит для определения педагогических приемов и методов для индивидуального подхода к каждому обучающемуся, корректировки плана работы с группой.

С целью определения уровня достижения планируемых компетенций в процессе освоения образовательной программы проводится промежуточная аттестация. Промежуточная аттестация обучающихся проводится комплексно по освоению программы два раза в течение учебного года: в декабре по итогам полугодия и в июне по итогам учебного года; а также по итогам освоения каждого краткосрочного модуля. Формы промежуточной аттестации определены учебным планом. Конкретные проверочные задания промежуточной аттестации разрабатывает педагог по каждой дисциплине с учетом заявленных требований к знаниям и умениям. В ходе промежуточной аттестации педагог определяет уровень достижения планируемых предметных результатов: высокий, средний, низкий. Успешная проектная работа в группе под руководством педагога и демонстрация результатов в ходе конкурсных мероприятий служит основанием для перевода обучающихся с одного уровня образовательной программы на следующий.

В случае завершения обучения по программе (не ранее чем после базового уровня) промежуточная аттестация обучающегося является итоговой. Свидетельство об освоении программы может быть выдано обучающимся, успешно прошедшим итоговую аттестацию.

Таблица 1

Промежуточная и итоговая аттестация обучающихся

№п/п	ФИО	Посещаемость в % соотношении	Кол-во набранных баллов по итогам работы над кейсами (от 1 до 10 баллов)	Сформир-ть softskills согласно перечня рабочей программы (от 1 до 10 баллов)	Сформир-ть hardskills согласно перечня рабочей программы (от 1 до 10 баллов)	Участие в конкурсах и соревнованиях

Уровень	Описание поведенческих проявлений
1 уровень - недостаточный	Обучающийся не владеет навыком, не понимает его важности, не пытается его применять и развивать.
2 уровень – развивающийся	- Обучающийся находится в процессе освоения данного навыка. - Обучающийся понимает важность освоения навыков, однако не всегда эффективно применяет его в практике.
3 уровень – опытный пользователь	- Обучающийся полностью освоил данный навык. - Обучающийся эффективно применяет навык во всех стандартных, типовых ситуациях.

4 уровень – продвинутый пользователь	- Особо высокая степень развития навыка. - Обучающийся способен применять навык в нестандартных ситуациях или ситуациях повышенной сложности.
5 уровень – мастерство	- Уровень развития навыка при котором обучающийся становится авторитетом и экспертом в среде сверстников. - Обучающийся способен передавать остальным необходимые знания и навыки для освоения и развития данного навыка.

Участие в мероприятиях ДТ «Кванториум», в том числе соревнования, выставки, олимпиады.

4.2. Информационные источники, используемые при реализации программы

1. Аллен Б. Дауни - Think DSP. Цифровая обработка сигналов на Python - Издательство "ДМК Пресс" - 2017 - 160с.
 2. Арсак Ж. Программирование игр и головоломок: Пер. с франц.- М.: Наука. Гл. ред. физ.-мат. лит., 1990.— 224 С.
 3. Бизли Д. Python. Подробный справочник. - Пер. с англ. - СПб.: Символ-Плюс, 2010. - 864 с, ил.
 4. Брукс Ф., Мифический человеко-месяц, или Как создаются программные системы, М.: Символ-Плюс, 2010.
 5. Буйначев, С. К. Основы программирования на языке Python [Электронный ресурс]: учебное пособие / С. К. Буйначев, Боклаг Н. Ю. – Электрон. текстовые дан. – Издательство Уральского университета, 2014.
 6. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
 7. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: – 2006.
 8. Доусен М. Програмуємо на Python (Python Programming for the Absolute Beginner)//изд. «Питер», серия Бестселлеры O'Reilly, 2016,- 416с.
- ПРИЛОЖЕНИЕ 1

Приложение 1. Утверждается ежегодно.
УТВЕРЖДЕН

Приказом директора от
ГБОУ гимназии № 406 Пушкинского района
Санкт-Петербурга

_____ В.В. Штерн

1. Календарно учебный график.
2. Календарно-тематический план

Календарный учебный график						
Год обучения	Дата начала обучения по программе	Дата окончания обучения по программе	Всего учебных недель	Всего учебных дней	Количество учебных часов	Режим занятий