

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Комитет по образованию Санкт-Петербурга
Администрация Центрального района Санкт-Петербурга
Государственное бюджетное общеобразовательное учреждение гимназия №155
Центрального района Санкт-Петербурга

Принята
Решением
Педагогического совета
Протокол № 11 от 10.06.2024

Утверждаю
директор гимназии №155
Шуйская О.Е.
Приказ № 28 -о от 10.06.2024

ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«ПРОГРАММИРОВАНИЕ»

срок реализации – 2 года
возраст учащихся –12-17 лет

разработчик программы:
Разумов Е.М,
педагог дополнительного образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Программирование» разработана на основе Федерального закона Российской Федерации № 273-ФЗ "Об образовании в Российской Федерации" от 29.12.2012; СанПиН 2.4.4.3172-14 "Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательной организации дополнительного образования детей" // Постановление Главного санитарного врача Российской Федерации от 04.07.2014 № 41; методических рекомендаций по проектированию дополнительных общеразвивающих программ в государственных образовательных организациях Санкт-Петербурга, находящихся в ведении Комитета по образованию // Распоряжение Комитета по образованию Санкт-Петербурга от 01.03.2017 № 617-р. и других нормативно-правовых документов, представленных в разделе программы «Литература и нормативно-правовые документы».

Уровень освоения программы: общекультурный.

Направленность программы: техническая.

Актуальность программы связана с современными требованиями воспитания всестороннего развития ребёнка с опорой на духовно-нравственное, гражданско-патриотическое воспитание, его интеллектуальное совершенствование, формирование общей культуры.

Актуальность программы состоит в том, что она составлена с учетом современных потребностей рынка в специалистах в области информационных технологий. 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. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Отличие данной программы заключается в нацеленности на коллективную творческую деятельность детей.

Данная программа:

- обеспечивает 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;
- предусматривает возможность индивидуальной работы с обучающимися.

Практическая значимость программы заключается в том, что она способствует более успешному овладению знаниями и умениями по предмету «Информатика» через развитие самостоятельности учащихся и оптимизацию средств и методов обучения.

Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как для учебных, так и прикладных задач.

Вместе с тем чистота и ясность его конструкций позволит учащимся потом с легкостью выучить любой другой язык программирования.

Новизна программы заключается в использовании различных современных образовательных технологий на занятиях, что в корне изменяет традиционный подход в

преподавании информатики. Современные технологии применяются педагогом как по отдельности, так и в различных сочетаниях (информационно-коммуникативные, здоровье сберегающие, игровые технологии, технологии творческого взаимодействия, творческих мастерских, проектные технологии, технологии интегрированного обучения и др.)

Содержание учебного материала направлено на:

- детальное изучение алгоритмизации;
- реализацию межпредметных связей;
- организацию проектной и исследовательской деятельности учащихся.

Педагогическая целесообразность. Педагогической целесообразностью является формирование мотивации учащихся к занятиям. Важным аспектом программы является самостоятельная работа над заданиями: обучающиеся учатся решать задачи самостоятельно без помощи преподавателя.

Целью реализации программы является формирование и развитие творчески развитой личности учащегося через создание условий для изучения методов программирования на языке Python; подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Для достижения поставленной цели перед обучающимися и педагогом стоят конкретные задачи:

Обучающие:

- закрепление навыков алгоритмического и логического мышления, грамотной разработки программ;
- закрепление навыков работы в интегральной среде разработки на языке Python;
- изучение конструкций языка программирования Python;
- расширения знакомства с основными структурами данных и типовыми методами обработки этих структур;
- приобретение навыков поиска информации в сети Интернет. Анализа выбранной информации на соответствие запросу, использование информации при решении задач.

Развивающие:

- формировать способность решать актуальные задачи в заданные сроки при разработке программ;
- развивать личностные компетенции такие, как: память, внимание, способность логически мыслить и анализировать, концентрировать внимание на главном при работе над творческими и научными проектами в области информатики;
- расширять круг интересов, критического и творческого мышления при работе в команде, проведении исследований, выполнении индивидуальных и групповых заданий;
- способствовать развитию творческих способностей учащихся, познавательных интересов, развитию индивидуальности и самореализации;
- расширять технологические навыки при подготовке различных информационных материалов;
- развивать познавательные способности учащихся, пространственное мышление, аккуратность и изобретательность при работе с техническими устройствами и выполнении учебных проектов;
- формировать творческий подход к поставленной задаче;

- развивать навыки инженерного мышления и эффективного использования электронного вычислительного оборудования.

Воспитательные:

- воспитывать дисциплинированность, ответственность, самоорганизацию, формировать организаторские и лидерские качества;
- воспитывать трудолюбие, уважение к труду;
- формировать чувство коллективизма и взаимопомощи;
- воспитывать чувство патриотизма, гражданственности, гордости за достижения отечественной науки и технике.
- воспитывать мотивацию учащихся к изобретательству, созданию собственных программных реализаций и электронных устройств;
- привить стремление к получению качественного законченного результата проектной деятельности;
- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

Условия реализации программы

Программа рассчитана на 2 года. В группу принимаются дети школьного возраста от 12 до 17 лет, желающие научиться программированию, не имеющие противопоказаний по состоянию здоровья.

Группа состоит из 12 человек.

Обучающиеся зачисляются в коллектив без конкурсного отбора по заявлению родителей (законных представителей).

Реализация программы предусматривает возможность обучения в дистанционном режиме без изменения календарно-тематического планирования. Для этого используются различные Интернет-платформы и средства дистанционных технологий (Zoom, Skype, группа ВКонтакте, Instagram, чат в WhatsApp).

Условиями успешной реализации программы являются:

- кадровое обеспечение – педагог дополнительного образования;
- материально-техническая база: персональный компьютер.
- научно-методическое обеспечение: пакет нормативно-правовых документов, регламентирующий деятельность педагога дополнительного образования, и пакет номенклатуры дел педагога.

Режим занятий

Занятия проводятся в трех группах, возможно разделение на подгруппы.

Занятия в группе проходят 3 раза в неделю по 1 часу, 108 часов за год.

Методы, технологии и формы обучения. На занятиях педагогом используются различные методы, технологии и формы обучения.

Методы обучения: в начале занятия учитель проводит демонстрацию презентации или показывает на практике реальные программы, а также готовые работы, выполненные на языке Python. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Формы обучения:

Фронтальная, круговая, групповая, парная, индивидуальная.

Технологии обучения:

Проектная деятельность, дифференцированное обучение, личностно-ориентированное обучение, развивающие и игровые технологии, технология творческого развития.

Формы проведения занятий. В зависимости от цели и задач конкретного занятия педагогом выбирается различная форма его проведения.

Формы занятий: контрольно-проверочные, сюжетно-игровые, самостоятельно-поисковые, проектно-творческие. Основной тип занятий – комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий программы выполняются самостоятельно с помощью персонального компьютера и необходимых программных средств. Каждый раздел охватывает отдельную информационную технологию или её часть. Внутри разделов разбивка по времени изучения производится педагогом дополнительного образования самостоятельно, но с учетом рекомендованного тематического плана. Каждая тема программы начинается с остановки задачи – характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить.

Планируемые результаты

Освоение данной программы обеспечивает достижение следующих личностных, метапредметных результатов и предметных результатов:

Личностные результаты:

- воспитание патриотизма, чувства гордости за свою Родину, российский народ и историю России;
- формирование целостного, социально ориентированного взгляда на мир в его органичном единстве и разнообразии природы, народов, культур и религий;
- формирование уважительного отношения к иному мнению, истории и культуре других народов;
- развитие самостоятельности и личной ответственности за свои поступки;
- формирование эстетических потребностей, ценностей и чувств;
- развитие навыков сотрудничества со взрослыми и сверстниками в разных ситуациях, умений не создавать конфликтов и находить выходы из спорных ситуаций;
- формирование установки на безопасный и здоровый образ жизни.

Метапредметные результаты:

- овладение способностью принимать и сохранять цели и задачи учебной деятельности, поиска средств её осуществления;
- освоение способов решения проблем творческого и поискового характера;
- формирование умения планировать, контролировать и оценивать учебные действия в соответствии с поставленной задачей и условиями её реализации; определять наиболее эффективные способы достижения результата;
- формирование умения понимать причины успеха/неуспеха учебной деятельности и способности конструктивно действовать даже в ситуациях неуспеха;
- освоение начальных форм познавательной и личностной рефлексии;
- овладение логическими действиями сравнения, анализа, синтеза, обобщения, классификации;
- готовность слушать собеседника и вести диалог; готовность признавать возможность существования различных точек зрения и права каждого иметь свою; излагать своё мнение и аргументировать свою точку зрения и оценку событий;
- определение общей цели и путей её достижения; умение договариваться о распределении функций и ролей в совместной деятельности; осуществлять взаимный контроль в совместной деятельности, адекватно оценивать собственное поведение и поведение окружающих.

Предметные результаты:

- умение 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 высокого уровня;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение 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;
- формирование умения соблюдать нормы информационной этики и права.
- Компетентностей подход реализации программы позволяет осуществить формирование у обучающегося как личностных, так и профессионально-ориентированных компетенции через используемые формы и методы обучения, нацеленность на практические результаты.

Структура занятий

Каждое занятие состоит из трёх основных частей: вводная, основная и заключительная.

- Вводная часть – 5-10 минут. Организационный момент (мотивация детей к учебной деятельности, приветствие). Подготовка детей к основной части, эмоциональный настрой на занятие.
- Основная часть – 30-35 минут. В зависимости от цели и задач занятие может состоять из следующего: повторение пройденного материала на предыдущем занятии.
- Заключительная часть – 5 минут. Рефлексия, домашнее задание.

Формы подведения итогов:

В процессе обучения проводится на занятиях разные виды контроля результативности усвоения программного материала.

Контроль, оценка результатов обучения детей:

Текущий контроль проводится на занятиях в виде наблюдения за успехами каждого обучающегося, процессом формирования компетенций. 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.

Периодический контроль проводится по окончании изучения каждого раздела в виде представления практических результатов выполнения заданий.

Итоговый контроль проводится в виде педагогического анализа результатов выполнения учащимися диагностических заданий, участия учащихся в мероприятиях (викторинах, соревнованиях), защиты проектов, решения задач поискового характера. Итоги реализации программы могут подводиться в виде итоговой аттестации следующих форм: защита индивидуального или группового проекта в виде публичного выступления с демонстрацией или группового проекта в виде публичного выступления с демонстрацией проектной работы;

соревнование; взаимооценка обучающимися работ друг друга. В процессе проведения итоговой аттестации оценивается результативность освоения программы.

УЧЕБНЫЙ ПЛАН (1 год)

№ п/п	Наименование темы	Количество часов			Формы контроля
		теория	практика	всего	
1.	Введение в программу. Организационная работа. Инструктаж по технике безопасности.	1	1	2	Знакомство. Устный опрос.
2.	Введение. Знакомство с Python	2	2	4	Устный опрос, беседа, решение задач
3.	Переменная	2	2	4	Устный опрос, беседа, решение задач
4.	Алгоритмы и их способы записи	2	2	4	Устный опрос, беседа, решение задач
5.	Условный оператор	4	4	8	Устный опрос, беседа, решение задач
6.	Цикл While	4	4	8	Устный опрос, беседа, решение задач
7.	Отладчик	4	4	8	Устный опрос, беседа, решение задач
8.	Цикл For	4	4	8	Устный опрос, беседа, решение задач
9.	True и False, break и continue	4	4	8	Устный опрос, беседа, решение задач
10.	Вложенные циклы	4	4	8	Устный опрос, беседа, решение задач
11.	Коллекции: множества	1	1	2	Устный опрос, беседа, решение задач
12.	Коллекции: строки	1	1	2	Устный опрос, беседа, решение задач
13.	Коллекции: списки	1	1	2	Устный опрос, беседа, решение задач
14.	Коллекции: кортежи	1	1	2	Устный опрос, беседа, решение задач
15.	Коллекции: словари	1	1	2	Устный опрос, беседа, решение задач
16.	Вложенные списки	2	2	4	Устный опрос, беседа, решение задач
17.	Функции	2	2	4	Устный опрос, беседа, решение задач
18.	Библиотеки	5	5	10	Устный опрос, беседа, решение задач

19.	Проектная деятельность	6	12	18	Разработка/ Защита мини-проекта
Итого:		52	56	108 ЧАСОВ	

УЧЕБНЫЙ ПЛАН (2 год)

№ п/п	Название темы	Количество часов			Формы контроля
		теория	практика	всего	
Программирование на Python (40 ч.)					
1.	Вводное занятие. Введение в программу 2-го года обучения Инструктаж по ТБ	1	1	2	Опрос. Отметка о прослушанн ых инструктажах в журнале.Решение задач
2.	Повторение. Решение задач наосновные конструкции и структуры данных. Решение задач на классы и библиотеки	1	3	4	Опрос. Решение задач
3.	Решение задач по теме «Повторение»	2	0	2	Решение задач
4.	QT 1. Что такое QT и PyQT	2	0	2	Опрос. Беседа. Решениезадач
5.	QT 2. QtDesigner	2	2	4	Опрос. Беседа. Решение задач
6.	QT 3. Файлы в Python. Типыфайлов и работа с ними. Внутреннее устройство файлов	2	2	4	Опрос. Решение задач
7.	QT 4. Диалоги, работа с изображениями	2	2	4	Опрос. Беседа. Решениезадач
8.	QT 5. Работа с простыми таблицами (csv). Работа с табличными данными в PyQT	2	2	4	Решение задач

9.	QT 6. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 1	2	2	4	Решение задач
10.	QT 7. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 2	4	0	4	Опрос. Беседа. Решение задач
11.	QT 8. Обработка событий. Сборка независимого приложения	3	1	4	Практическая работа
12.	Защита проекта QT	2	0	2	Защита проекта
Модуль 2. PyGame (44 ч.)					
13.	Подключение PyCharm. Работа с удаленными репозиториями	3	1	4	Опрос. Беседа. Решение задач
14.	PyGame 1. Введение	1	1	2	Опрос. Беседа. Решение задач
15.	PyGame 2. Игровой цикл. События	2	2	4	Решение задач
16.	PyGame 3. Основные команды при одиночной работе с Git	2	2	4	Решение задач
17.	PyGame 4. Клеточное поле	3	1	4	Решение задач
18.	PyGame 5. Классические игры на клеточном поле	2	2	4	Решение задач
19.	PyGame 6. Изображения. Спрайты	1	1	2	Опрос. Беседа. Решение задач
20.	PyGame 7. Столкновения и другие взаимодействия	2	2	4	Опрос. Беседа. Решение задач
21.	PyGame 8. Игра в целом	2	4	6	Практическая работа
22.	PyGame 9. Украшения игр	2	2	4	Практическая работа
23.	Цели и подходы к тестированию. Создание "самодельных" тестов (без библиотек)	1	3	4	Практическая работа
24.	Защита проекта PyGame	0	2	2	Защита проекта
Проекты WEB и API (24 ч.)					
25.	WEB. Работа с популярными форматами	2	2	4	Опрос. Беседа. Решение задач

	файлов (json, xml)				
26.	WEB. Знакомство с API	2	2	4	Решение задач
27.	WEB. Понятие исключения, обработка исключений. Собственные исключения.	2	2	4	Решение задач
28.	Разработка и защита итогового проекта	2	10	12	Защита индивидуального/ группового проекта
ИТОГО		54	54	108	

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Год обучения	Дата начала обучения по программе	Дата окончания обучения по программе	Всего учебных недель	Количество учебных часов	Режим занятий
1-й	01.09.2023	31.05.2024	36	108	3x1
2-ой			36	108	3x1

РАБОЧАЯ ПРОГРАММА

Год обучения	Задачи	Ожидаемые результаты
1-й	- Обеспечить возможность обучения детей с первичными навыками программирования в том числе, одарённых детей. - Сформировать интерес к IT-технологиям - Сформировать первичные навыки объектно - ориентированного и функционального программирования, алгоритмизации, работы в различных интегрированных средах разработки на языке Python - изучить основные конструкции языка программирования; -приобрести навыки поиска, обработки, анализа, а также умение эффективно использовать различные Интернет-ресурсы.	Личностные результаты: ● Формирование ответственного отношения к учению, способности довести до конца начатое дело; ● Формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой, благодаря иллюстративной среде программирования, мотивации к обучению и познанию; ● Развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам; ● Формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участие в конкурсах и конференциях различного уровня; ● Формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий; ● Формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;

		● Формирование ценности здорового и безопасного образа жизни; ● Усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой; Метапредметные результаты: ● Умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности; ● Умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности; ● Умение критически оценивать правильность решения учебно-исследовательской деятельности; ● Умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая; ● Учебно-исследовательские задачи; ● Умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями; ● Владение основами самоконтроля, способность к принятию решений; ● Умение создавать применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ; ● Формирование и развитие компетенции в области использования информационно-коммуникационных технологий (ИКТ-компетенция); ● Умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной учебно-исследовательской деятельности. Предметные результаты:
--	--	--

		● Умение 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 высокого уровня; ● Развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами; ● Умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления и анализ числовых и текстовых данных; ● Навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; ● Умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ. Выполнять созданные программы; ● Умение 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;
--	--	---

2-ой	- 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;	Личностные результаты: по окончании программы учащийся: • 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 учащийся: • 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. 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.
--	--	--

Содержание

1 год обучения

Тема № 1. Вводное занятие. Организационная работа. Инструктаж по технике безопасности

Теория: Инструктажи по технике безопасности с отметкой в журнале. Введение в программу «Программирование на Python» 1-го года обучения. Понятия кода.

Простейшие программы с выводом на экран. Знакомство с виртуальной средой взаимодействия – регистрация, организация личного кабинета, поиск и выкладывание материалов; знакомство с системой автоматизированной проверки задач и системой Яндекс.

Практика: решение задач.

Тема № 2. Введение. Знакомство с Python

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

Практика: решение задач.

Тема № 3. Переменная

Теория: Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python

Практика: решение задач.

Тема № 4. Алгоритмы и их способы записи

Теория: знакомство с понятие алгоритм. Виды алгоритмов.

Практика: разработка алгоритмов и программ, определение работоспособности разработанной программы.

Тема № 5. Условный оператор

Теория: Ввод-вывод в программе, условный оператор. Простейшие программы с использованием условного оператора if, и конструкцией if-else-elif.

Практика: решение задач.

Тема № 6. Цикл While

Теория: Цикл while. Считывание данных до стоп значения. Бесконечный цикл. Решение задач. **Практика:** решение задач по теме «Срезы и диапазоны».

Тема № 7. Отладчик

Теория: Визуальный отладчик. Основные функции отладки.

Практика: практическая работа.

Тема № 8. Цикл For

Теория: Цикл For (цикл с параметром). Функция range.

Практика: решение задач.

Тема № 9. True и False, break и continue

Теория: логические операции в Python. True и False, break и continue/

Практика: решение задач.

Тема № 10. Вложенные циклы

Теория: Инструкции управления циклом в Python. Вложенные циклы. Понятие итерация.

Практика: решение задач.

Тема № 11. Коллекции: множества

Теория: целочисленный тип данных int, числа с плавающей точкой float, троенные функции max, min, abs.

Практика: решение задач.

Тема № 12. Коллекции: строки

Теория: Представление строк в памяти компьютера. Таблица символов ASCII. Таблица символов Unicode. Функция ord(). Функция chr()

Практика: решение задач.

Тема № 13. Коллекции: списки

Теория: Создание списков. Пустые списки. Встроенная функция list(). Вывод списков

Практика: решение задач.

Тема № 14. Коллекции: кортежи

Теория: Определение кортежей. Способы применения.

Практика: решение задач.

Тема № 15. Коллекции: словари

Теория: Словари. Способы применения. Методы словарей.

Практика: решение задач.

Тема № 16. Вложенные списки

Теория: Методы списков. Вложенные списки.

Практика: решение задач.

Тема № 17. Функции

Теория: Функции, функциональная парадигма программирования.

Практика: решение задач.

Тема № 18. Библиотеки

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

Практика: решение задач.

Тема № 19. Проектная деятельность

Теория: Презентация тем итоговых проектных работ. Принципы распределения ролей при выполнении проекта с применением нейронных сетей. Примерное содержание 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.

2 ой год обучения

Тема № 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 № 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 № 3. Решение задач по теме «Повторение».

Практика: Решение задач

Тема №4. QT 1. Что такое QT и PyQt.

Теория: История, предназначение, преимущества и недостатки фреймворка Qt. Подключение и первые шаги с PyQt.

Практика: Решение задач

Тема №5. QT 2. QtDesigner

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

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

Тема №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 работай с файлами из настольного приложения.

Тема №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 №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 №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 №10. QT 7. Введение в БД, работа с SQL-таблицами и отображение данных в PyQt. Часть 2

Теория: Технология написания более сложных SQL запросов. **Практика:** Решение задач на работу с БД из PyQt

Тема №11. QT 8. Обработка событий. Сборка независимого приложения

Теория: Обработка событий стандартных устройств ввода (мышь, клавиатура). Обработка событий, генерируемых модулями программы. Сборка интерактивного приложения.

Практика: Работа над проектом.

Тема №12. Защита проекта QT

Практика: Защита проекта (настольное приложение)

Тема №13. 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 №14. PyGame 1. Введение

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

Практика: Решение задач

Тема № 15. PyGame 2. Игровой цикл. События

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

Практика: Решение задач

Тема №16. 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 №17. PyGame 4. Клеточное поле

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

Практика: Решение задач

Тема № 18. 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 № 19. 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 №20. 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 №21. PyGame 8. Игра в целом

Теория: Проектирование архитектуры игрового приложения.

Практика: Создание своей игры

Тема №22. PyGame 9. Украшения игр

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

Практика: Украшение своей игры

Тема №23. 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 №24. Защита проекта PyGame

Практика: Защита проекта (игра)

Тема. №25. 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 №26. WEB. Знакомство с API

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

Практика: Решение задач

Тема №27. 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 №28. Разработка и защита итогового проекта

Практика: Защита индивидуального/группового проект

Календарно-тематическое планирование
1 год обучения

№	Раздел, тема занятия	Кол-во часов	Примерная дата	Дата по факту
Организационная работа (2 ч.)				
1	Введение в программу. Организационная работа. Инструктаж по технике безопасности.	2		
Введение в программирование (12 ч.)				
2	Введение. Знакомство с Python.	4		
3	Переменная	4		
4	Алгоритмы и их способы записи	4		
Базовые конструкции в Python (48 ч.)				
5	Условный оператор	8		
6	Цикл While	8		
7	Отладчик	8		
8	Цикл For	8		
9	True и False, break и continue	8		
10	Вложенные циклы	8		
Решение прикладных задач в Python (46 ч.)				
11	Коллекции: множества	2		
12	Коллекции: строки	2		
13	Коллекции: списки	2		
14	Коллекции: кортежи	2		
15	Коллекции: словари	2		
16	Вложенные списки	4		
17	Функции	4		
18	Библиотеки	10		
19	Проектная деятельность	18		
ИТОГО		108 ЧАСОВ		

2 год обучения

№	Раздел, тема занятия	Кол-во часов	Примерная дата	Дата по факту
Программирование на Python (40 ч.)				
1.	Вводное занятие. Введение в программу 2-го года обучения Инструктаж по ТБ	2		
2.	Повторение. Решение задач на основные конструкции и структуры данных. Решение задач на классы и библиотеки	4		
3.	Решение задач по теме «Повторение»	2		
4.	QT 1. Что такое QT и PyQT	2		
5.	QT 2. QtDesigner	4		
6.	QT 3. Файлы в Python. Типы файлов и работа с ними. Внутреннее устройство файлов	4		
7.	QT 4. Диалоги, работа с изображениями	4		
8.	QT 5. Работа с простыми таблицами (csv). Работа с табличными данными в PyQT	4		
9.	QT 6. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 1	4		
10.	QT 7. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 2	4		
11.	QT 8. Обработка событий. Сборка независимого приложения	4		
12.	Защита проекта QT	2		

Модуль 2. PyGame (44 ч.)				
13.	Подключение PyCharm. Работас удаленными репозиториями	4		
14.	PyGame 1. Введение	2		
15.	PyGame 2. Игровой цикл. События	4		
16.	PyGame 3. Основные команды при одиночной работе с Git	4		
17.	PyGame 4. Клеточное поле	4		
18.	PyGame 5. Классические игры на клеточном поле	4		
19.	PyGame 6. Изображения. Спрайты	2		
20.	PyGame 7. Столкновения и другие взаимодействия	4		
21.	PyGame 8. Игра в целом	6		
22.	PyGame 9. Украшения игр	4		
23.	Цели и подходы к тестированию. Создание "самодельных" тестов (безбиблиотек)	4		
24.	Защита проекта PyGame	2		
Проекты WEB и API (24 ч.)				
25.	WEB. Работа с популярными форматами файлов (json, xml)	4		
26.	WEB. Знакомство с API	4		
27.	WEB. Понятие исключения,обработка исключений. Собственные исключения.	4		
28.	Разработка и защита итогового проекта	12		
ИТОГО		108 ЧАСОВ		

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Методическое обеспечение образовательной программы

Основными, в освоении образовательной программы, являются принципы:

- “от простого к сложному”,
- “от медленного к быстрому”,
- “посмотри и повтори”,
- “осмысли и выполни”,
- “от эмоций к логике”,
- “от логики к ощущению”.

В зависимости от психофизиологических способностей детей педагог, на своё усмотрение, может изменить уровень сложности освоения данной Образовательной программы.

Успешное освоение всех рекомендаций должно сочетаться с созданием на занятии такого психологического микроклимата, когда педагог, свободно общаясь с детьми на принципах сотрудничества, содружества, общего интереса к делу, четко реагирует на восприятие учащихся, поддерживая атмосферу радости, интереса и веселья, побуждает их к творчеству.

В целях создания положительной мотивации используются игровые моменты, ролевые игры, направленные на переключение внимания, разгрузку и отдых. На занятии каждый учащийся может почувствовать себя настоящим программистом – каждое занятие у учащихся есть возможность написать свою собственную программу, найти в ней ошибку и оптимизировать код.

Для успешной работы коллектива необходимо:

1. Материально-техническая база:

- Персональные компьютеры с выходом в сеть Интернет
- Доступ к платформе Stepik