

Государственное автономное образовательное учреждение дополнительного образования
«Центр для одаренных детей «Поиск»

УТВЕРЖДЕНО
приказом Центра «Поиск»
№ 133 от 25 марта 2025 г.

Дополнительная общеобразовательная общеразвивающая программа
технической направленности

«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ. РАЗРАБОТКА НА PYTHON»

Направление: наука

Возраст обучающихся: 14-16 лет

Объем программы: 102 часа

Срок освоения: 1 год

Форма обучения: очная

Автор программы: Никотина Лидия Леонидовна, заместитель
директора по учебной работе ГАОУ ДО «Центр для
одаренных детей «Поиск»;
Ионисян Андрей Сергеевич, педагог
дополнительного образования ГАОУ ДО «Центр для
одаренных детей «Поиск»

Ставрополь, 2025

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ	0
ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	1
УЧЕБНЫЙ ПЛАН	7
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК	8
РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА «РАЗРАБОТКА НА PYTHON»	9
ОЦЕНОЧНЫЕ МАТЕРИАЛЫ	17
КАДРОВОЕ ОБЕСПЕЧЕНИЕ	21
МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ПРОГРАММЕ	22
УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ	24

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Сегодня во всех отраслях российской экономики происходят значительные изменения и перестройки. Особенно остро это касается ИТ-отрасли: производители зарубежного программного обеспечения уходят с рынка, российские серверы и сайты российских компаний подвергаются негативному внешнему воздействию. На этом фоне актуально быстрое и эффективное увеличение числа программистов, способных на федеральном уровне решать задачи государства.

Программирование занимает значительное место в системе подготовки профессионалов в области информационных технологий. В федеральном проекте «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика РФ» обозначена уникальность и ценность программирования как учебного вида деятельности. Готовить профессиональных программистов нужно уже в школе.

Школьники, умеющие разработать и реализовать проект компьютерной программы среднего уровня сложности с графическим пользовательским интерфейсом и доступом к базам данных и сети Интернет, способны стать лидерами в школе и в обществе. Они лучше успевают и по другим предметам, поскольку культура их мышления выше, а ассортимент методов выполнения различных работ богаче. В связи с этим представляется достаточно важным привить учащимся навыки промышленного программирования.

Вид программы – модульная.

Программа представляет собой совокупность 2-х модулей, которые реализуются в очной форме.

№	Название модуля	Форма обучения	Класс обучающегося
1.	Графический интерфейс PyQt, база данных SQLite, WEB-программирование.	очная	8-9
2.	Разработка практических приложений.	очная	8-9

1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.1. Направленность программы

Дополнительная общеобразовательная общеразвивающая программа «Информационные технологии. Разработка на Python» имеет техническую направленность, в связи с этим рассматриваются два актуальных аспекта изучения:

1) технологический: программирование рассматривается как средство формирования образовательного потенциала, позволяющего развивать наиболее передовые на сегодняшний день технологии – цифровые.

2) общеобразовательный: программирование рассматривается как средство развития основных познавательных процессов, умения анализировать, выявлять сущности и отношения, описывать планы действий и делать логические выводы.

1.2. Адресат программы

Программа адресована обучающимся от 14 до 16 лет.

Программа предназначена для школьников 8-9 классов с повышенным уровнем мотивации к обучению, желающих развить навыки XXI века, получить углубленные теоретические и практические знания и навыки по актуальным в настоящее время направлениям в сфере цифровой экономики.

1.3. Актуальность программы

Совершенствование технологических и программных средств привело к снижению количества часов, отводимых для изучения программирования в Программе среднего общего образования по информатике. Современные визуальные и мультимедийные пользовательские среды являются теми конкурентами, которые вытесняют разработку программ из сферы интересов школьников. Для работы за компьютером для поиска информации в сети пользователь имеет простые инструменты, не требующие мыслительных усилий при применении. Как следствие, в большинстве школ отсутствует системная работа по обучению школьников 7-10 классов языкам программирования.

Дополнительная общеобразовательная общеразвивающая программа «Информационные технологии. Разработка на Python» обеспечивает изучение востребованного языка программирования, нацелена на формирование математического аппарата описания и построения процессов обработки информации, в том числе человеком и технологическим устройством, создания и исследования числовых и нечисловых математических моделей, создания компьютерных программ с графическим пользовательским интерфейсом, доступом к базам данных и сети Интернет.

1.4. Отличительные особенности/новизна программы

Содержание программы «Информационные технологии. Разработка на Python» разработано с учётом современных требований, предъявляемых к уровню подготовки школьников к жизни и работе в цифровом мире в рамках ключевых направлений федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика РФ».

Программа способствует дополнительному образованию детей выступить «посредником» между сферой науки, общим образованием и бизнесом, привлекая необходимые научные кадры и организации для поддержки мотивации обучающихся к изучению программирования, развития профильной, учебно-исследовательской, самостоятельной творческой деятельности, способностей к рефлексии и оценке результатов обучения.

Уровень освоения программы – углубленное изучение языка программирования.

1.5. Объем и срок освоения программы

Объем программы – 102 часа.

Срок реализации программы – 1 год.

1.6. Цели и задачи программы

Цель программы – создание условий для повышения качества образования через обеспечение участников образовательного процесса углубленными знаниями и компетенциями из области программирования, востребованными современной цифровой экономикой, подготовка базы для последующего профессионального изучения программирования в высшей школе.

Задачи программы

1) Обучающие:

- формирование умений в области создания текстов программ для персональных компьютеров на основе глубоких знаний среды программирования и языков программирования;
- расширение и углубление представлений об автоматизированной обработке информации;
- освоение методов программирования: выдвижение и обоснование идеи решения задачи, структурирование этой идеи, формализация элементов полученной структуры средствами выбранного языка, анализ результатов решения задачи при различных значениях исходных данных.

2) Развивающие:

- формирование операционного мышления, направленного на выбор оптимальных решений;
- развитие способностей по самостоятельному приобретению знаний, умений, навыков, ускорение процесса перехода от обучения к научению, самообучению – наивысшей ступени образовательного процесса;
- развитие способностей эффективной работы в условиях ограничений;
- развитие умений эффективного использования возможностей информационной среды, защиты от негативных воздействий;
- формирование способностей выдвигать и доказывать гипотезы опытным путем, разрабатывать стратегию решения, прогнозировать результаты своей деятельности, анализировать и находить рациональные способы решения задачи путем оптимизации, детализации созданного алгоритма;
- формирование навыка рефлексивной деятельности.

3) Воспитательные:

- формирование определенного мировоззрения, связанного с устоями и обычаями, национальными и культурными традициями, историей Ставропольского края, межнациональной толерантностью;

- освоение информационной культуры: ответственного отношения к информации с учетом правовых и этических аспектов ее распространения, избирательного отношения к полученной информации;
- восприятие системы ценностей, принципов, правил, стереотипов информационного общества;
- формирование умений работать в команде;
- ранняя профориентация школьников через ознакомление с востребованными профессиями и видами профессиональной деятельности, связанными с цифровыми технологиями.

1.7. Планируемые результаты освоения программы

1. Предметные результаты:

- знание продвинутого программирования на языке программирования Python;
- владение приёмами обработки и хранения числовой, текстовой, графической информации;
- свободное использование стандартных библиотек языка программирования Python;
- умение подключать и работать с дополнительными модулями языка программирования Python;
- самостоятельная разработка приложений на языке программирования Python.

2. Метапредметные результаты:

- способность соотносить и оценивать результаты своей деятельности с поставленной целью;
- использование цифровых технологий в качестве инструмента достижения цели;
- понимание связи цифровых технологий с другими научными направлениями;
- осуществление саморефлексии и рефлексии деятельности группы, результатом которой будет опробование новых стратегий поведения внутри своих же ролей.

3. Личностные результаты:

- понимание и правильное оценивание своих возможностей;
- развитие навыков группового общения, умения работать в команде;
- обучение рациональному распределению времени работы;
- формирование способностей эффективно распределять роли в ходе выполнения командной работы.

2. Организационно-педагогические условия реализации программы

2.1. Язык реализации программы

Реализация дополнительной общеобразовательной общеразвивающей программы «Информационные технологии. Разработка на Python»

осуществляется на государственном языке Российской Федерации (на русском языке).

2.2. Форма обучения: очная.

2.3. Особенности реализации программы

Программа реализуется по модульному принципу.

2.4. Условия набора и формирования групп

На обучение зачисляются учащиеся 8-9 классов общеобразовательных организаций с повышенным уровнем мотивации к обучению:

- 1) подавшие заявку и прошедшие конкурсный отбор;
- 2) победители и призёры олимпиад и других интеллектуальных конкурсов по математике и/или информатике регионального и всероссийского уровней.

Условия конкурсного отбора гарантируют соблюдение прав учащихся в области дополнительного образования и обеспечивают зачисление наиболее способных и подготовленных обучающихся к освоению программы.

Количество обучающихся в группе – 12 человек.

Условия формирования групп:

- состав групп – разновозрастной.

2.5. Формы организации и проведение занятий

Формы организации занятий: аудиторные под непосредственным руководством преподавателя.

Формы проведения занятий: комбинированные, теоретические, практические, контрольные, решение кейса, защита проекта.

Формы организации деятельности обучающихся:

- фронтальная: работа педагога со всеми учащимися одновременно;
- групповая: организация работы в малых группах, в т.ч. в парах, для выполнения определенных задач; задание выполняется таким образом, чтобы был виден вклад каждого учащегося;
- коллективная: организация проблемно-поискового взаимодействия между всеми детьми одновременно;
- индивидуальная: организуется для работы с одаренными детьми по решению более сложных кейсов; для коррекции пробелов в знаниях и отработки отдельных навыков для учеников, пропустивших занятия.

Режим занятий. Программа реализуется в г. Ставрополе в очной форме один раз в неделю по три учебных часа.

2.6. Основные методы реализации содержания программы

Работа в рамках программы предполагает интеграцию следующих методик:

- 1) Кейс-метод;
- 2) Дизайн-мышление.

Кейс-метод

Кейс-метод – это метод проведения детального анализа конкретной ситуации, который используется для достижения определенных целей обучения.

Цель использования кейс-метода в рамках программы заключается в совместном анализе учащимися ситуации (кейса), возникающей у конкретного заказчика, оценке предложенных условий, выработке практического решения и нахождении лучшего выхода из ситуации в контексте поставленной проблемы.

Использование кейс-метода является основным в реализации программы, так как обеспечивает стимул для саморазвития, самообразования, саморегуляции своей образовательной деятельности.

Метод дизайн-мышления

Это метод создания продуктов/услуг, ориентированных на интересы пользователя. Любая идея здесь – это решение потребности человека.

Принципы дизайн-мышления:

- 1) Ошибайся раньше, ошибайся чаще.
- 2) Создай прототип вместо того, чтобы рассказать о продукте.
- 3) В первую очередь зафиксируй пожелания пользователя.
- 4) Делай продукт вместе с пользователем.

Три кита дизайн-мышления:

- 1) Процесс: есть алгоритм, интерактивность, смешанные команды.
- 2) Подход: человекоцентричность, эмпатия, культ быстрых ошибок.
- 3) Среда: осязаемый мыслительный процесс, возможность «думать руками».

УЧЕБНЫЙ ПЛАН

№	Наименование модуля	Контактная работа обучающихся с преподавателем, часов			Формы контроля / аттестации
		Теория	Практика	Всего	
1.	Графический интерфейс PyQt, база данных SQLite, WEB- программирование	15	30	45	презентация результатов кейса
2.	Разработка практических приложений		57	57	публичная презентация результатов кейса
Итого:		15	87	102	

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Наименование модуля	Год обучения	Дата начала обучения	Дата окончания обучения	Кол-во учебных недель	Кол-во учебных дней	Кол-во учебных часов	Режим занятий
Графический интерфейс PyQt, база данных SQLite, WEB-программирование	1-й	08.09. 2025	26.12. 2025	15	15	45	Один раз в неделю по три урока
Разработка практических приложений	1-й	12.01. 2026	28.05. 2025	19	19	57	Один раз в неделю по три урока

РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА «РАЗРАБОТКА НА PYTHON»

Курс «Разработка на Python» предназначен для учащихся 8-9 классов.

В курсе рассматриваются основные конструкции языка программирования Python, основы объектно-ориентированной технологии, создание графического пользовательского интерфейса PyQt, создание и взаимодействие с базой данных SQLite, создание и настройка WEB-сервера FLASK.

**В результате освоения учебного курса обучающийся должен:
знать:**

- основные принципы построения программ на языке;
- основные способы представления структур данных;
- базовые конструкции языка Python (операции присваивания, ветвления, цикла, ввода/вывода, описание данных, запись констант и выражений);
- основные методы работы с контейнерами данных (списками, строками, кортежами, словарями);
- алгоритмы обработки строковых переменных;
- понятие пользовательских функции, способы организации функций, принципы модульной технологии программирования;
- принципы подключения и использования дополнительных модулей языка;
- основы объектно-ориентированной технологии;
- принципы разработки GUI-интерфейса;
- основы языка SQL;
- принципы взаимодействия с сервером баз данных;
- принципы разработки веб-приложений.

уметь:

- формализовать поставленную задачу;
- тестировать и отлаживать задачи с целью повышения надёжности и эффективности;
- использовать современные готовые библиотеки классов, технологии и инструментальные средства.

ТЕМАТИЧЕСКИЙ ПЛАН КУРСА «РАЗРАБОТКА НА PYTHON»

№ темы	Наименование раздела, темы	Контактная работа обучающихся с преподавателем, часов		
		Теория	Практика	Всего
	Модуль 1. Графический интерфейс PyQt, база данных SQLite, WEB-программирование	15	30	45
1	Тема 1. Объектно-ориентированная технология в Python	3	6	9
2	Тема 2. Библиотека создания графического пользовательского интерфейса PyQt	3	6	9
3	Тема 3. Сервер баз данных SQLite	3	6	9
4	Тема 4. Обращение к серверам сети Интернет	3	6	9
5	Тема 5. Web-сервер FLASK	3	6	9
	Модуль 2. Разработка практических приложений		57	57
6	Тема 6. Разработка программ с графическим пользовательским интерфейсом PyQt		15	15
7	Тема 7. Разработка программ взаимодействия с базой данных SQLite		12	12
8	Тема 8. Создание WEB-сервера FLASK		15	15
9	Тема 9. Работа над проектом		15	15
Итого:		15	87	102

СОДЕРЖАНИЕ КУРСА «РАЗРАБОТКА НА PYTHON»

Модуль 1. Графический интерфейс PyQt, база данных SQLite, WEB-программирование

Тема 1. Объектно-ориентированная технология в Python

Теория. Классы и объекты. Примеры объектно-ориентированных иерархий. Наследование, инкапсуляция, полиморфизм. Реализация ООП в языке Python.

Практика. Создание классов и объектов в Python. Разбор решений объектно-ориентированных задач на программирование.

Форма подведения итогов: тестирование.

Тема 2. Библиотека создания графического пользовательского интерфейса PyQt

Теория. Каркас PyQt программы, основные виджеты (QWidget, QPushButton, QLabel, QLineEdit, QListWidget). Подключение обработчиков событий к виджетам. Создание и подключение GUI-интерфейса в программе QtDesigner.

Практика. Создание и изучение вычислительных программ с графическим пользовательским интерфейсом PyQt на языке программирования Python.

Форма подведения итогов: тестирование.

Тема 3. Сервер баз данных SQLite

Теория. Способы хранения больших объемов информации в ЭВМ. Реляционные базы данных. Основные команды языка SQL. Простой сервер баз данных SQLite. Программа быстрого создания, просмотра и модификации базы данных SQLiteStudio. Создание, подключение и работа с SQLite-базами данных в Python. Модуль sqlite3.

Практика. Создание базы данных в SQLiteStudio. Создание и изучение программ для доступа к SQLite-базе данных на языке Python (с текстовым и с графическим пользовательским интерфейсами).

Форма подведения итогов: тестирование.

Тема 4. Обращение к серверам сети Интернет

Теория. Общие принципы хранения информации в сети Интернет. IP-адрес, порт, доменное имя, сайт. Форматы XML и JSON обмена информации между серверами в сети Интернет. Модули json и request языка программирования Python. Создание подключения к удаленному серверу. Чтение и передача информации. Расшифровка JSON-пакетов.

Практика. Создание и изучение программ доступа к серверам сети интернет (прогноз погоды, Яндекс-карты) на языке программирования Python.

Форма подведения итогов: тестирование.

Тема 5. Web-сервер FLASK

Теория. Основные теги языка HTML. WEB-сервер. Модуль flask языка программирования Python, Создание простейшего локального WEB-сервера. Динамический доступ к страницам WEB-сервера. Методы GET и POST. Шаблоны HTML-страниц с динамическим содержимым. Доступ к локальному WEB-серверу из сети Интернет. Реальные IP-адрес, доменное имя и хостинг сайта.

Практика. Создание и отладка простого локального WEB-сервера на основе модуля FLASK. Проброс доступа к локальному WEB-серверу.

Форма подведения итогов: тестирование.

Модуль 2. Разработка практических приложений

Тема 6. Разработка программ с графическим пользовательским интерфейсом PyQt

Теория. Библиотека PyQt.

Практика. Создание вычислительных программ с графическим пользовательским интерфейсом PyQt на языке программирования Python.

Форма подведения итогов: демонстрация работы компьютерных программ.

Тема 7. Разработка программ взаимодействия с базой данных SQLite

Теория. Библиотека sqlite3.

Практика. Создание базы данных в SQLiteStudio. Создание программ для доступа к SQLite-базе данных на языке Python (с текстовым и с графическим пользовательским интерфейсами).

Форма подведения итогов: демонстрация работы компьютерных программ.

Тема 8. Создание WEB-сервера FLASK

Теория. Библиотеки request и flask.

Практика. Создание и отладка простого локального WEB-сервера на основе модуля FLASK. Проброс доступа к локальному WEB-серверу. Создание клиентской программы доступа к WEB-серверу.

Форма подведения итогов: демонстрация работы компьютерных программ.

Тема 9. Работа над проектом

Теория. Основные стадии создания индивидуального проекта. Техническое задание, компьютерная программа, пояснительная записка, презентация. Приемы защиты проекта.

Практика. Создание технического задания, компьютерной программы и презентации индивидуального проекта. Репетиция процесса защиты проекта.

Форма подведения итогов: защита индивидуального проекта.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Раздел, тема	Форма занятия	Приемы и методы организации образовательного процесса	Дидактический материал	Техническое оснащение	Форма подведения итогов
Тема 1. Объектно-ориентированная технология в Python	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет	Тестирование
Тема 2. Библиотека создания графического пользовательского интерфейса PyQt	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет	Тестирование
Тема 3. Сервер баз данных SQLite	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование
Тема 4. Обращение к серверам сети Интернет	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование
Тема 5. Web-сервер FLASK	Комбинированная	Объяснительно-иллюстративный.	«Изучаем Python. Программирование игр,	Проекционное оборудование,	Тестирование

		Частично-поисковый. Исследовательский	визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	ПК. Доступ к сети Интернет	
Тема 6. Разработка программ с графическим пользовательским интерфейсом PyQt	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет	Демонстрация работы компьютерных программ
Тема 7. Разработка программ взаимодействия с базой данных SQLite	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет	Демонстрация работы компьютерных программ
Тема 8. Создание WEB-сервера FLASK.	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет	Демонстрация работы компьютерных программ
Тема 9. Работа над проектом	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский	«Изучаем Python. Программирование игр, визуализация данных, веб-приложения». Эрик Мэтиз. – СПб.: Питер, 2017	Проекционное оборудование, ПК. Доступ к сети Интернет	Защита индивидуального проекта

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Оценочные материалы по программе «Разработка на Python» разрабатываются для осуществления следующих видов контроля.

1) Входной контроль

Цель входного контроля — оценка общего уровня подготовки обучающегося. Для входного контроля используется тестирование. Результаты входного контроля используются для вывода о целесообразности редактирования планирования. Для оценивания используется 100-балльная система.

Входной контроль проводится с каждым обучающимся индивидуально по двум параметрам – теоретический и практический.

Теоретический параметр проверяет базовые знания по основам программирования на языке Python. Его значение определяется по результатам тестирования.

Практический параметр демонстрирует степень владения языком Python путем создания несложной компьютерной программы на языке программирования Python. Его значение определяется по результатам практического тестирования.

Во время проведения входной диагностики педагог заполняет входные параметры информационной карточки «Учёт результатов обучения», пользуясь шкалой, представленной в таблице.

Оценка параметров входного контроля

Наименование уровня	Результат диагностики, %
Элементарный уровень	0 – 54%
Низкий уровень	55 – 69%
Средний уровень	70 – 84%
Высокий уровень	85 – 100%

2) Текущий контроль

Осуществляется после каждой темы в форме решения кейса;

3) Промежуточная аттестация

Проводится в конце каждого модуля в форме закрытой презентации (внутри группы) решения кейса из реального сектора экономики.

4) Итоговая аттестация

Завершает программу, проводится в виде публичной (с приглашением представителей реального сектора экономики, родителей, гостей) групповой защиты решения кейса из реального сектора экономики.

Формы отслеживания результатов: наблюдение, тестирование, контрольный опрос (устный или письменный), собеседование, публичная презентация (защита кейса), психологический мониторинг.

Формы фиксации результатов: аналитическая справка, материалы тестирования и опроса, результаты психологического мониторинга, презентация, отчёт.

Документальной формой подтверждения итогов реализации программы является документ об обучении «Сертификат» (с оценкой) установленного Центром «Поиск» образца.

Степень выраженности компетенций имеет три уровня.

- низкий уровень: 55-69%
- средний уровень: 70-84%
- высокий уровень: 85-100%

Варианты контроля знаний описаны в Приложении 1.

Мониторинг результатов обучения

№	Показатель	Критерии	Степень выраженности оцениваемого качества	Возможное количество баллов	Методы диагностики
1	Теоретические знания по основным разделам учебно-тематического плана программы	Соответствие теоретических знаний ребенка программным требованиям, знание техники безопасности.	Низкий уровень: ребенок овладел менее чем $\frac{1}{2}$ объема знаний, предусмотренных программой, не обнаружил, что вторая половина ему нужна для решения задач.	55-69%	Наблюдение. Тестирование. Контрольный опрос. Публичная презентация решения кейса.
			Средний уровень: объем усвоенных знаний составляет более $\frac{1}{2}$, понимает, что оставшиеся знания тоже необходимы.	70-84%	
			Высокий уровень: ребенок освоил практически весь объем знаний, предусмотренных программой за конкретный период.	85-100%	
2	Владение понятийным аппаратом по	Осмысленность и правильность использования	Низкий уровень: ребенок сочетает специальную	55-69%	Собеседование. Опрос. Наблюдение.

	тематике программы	специальной терминологии	терминологию с бытовой.		Публичная презентация решения кейса.
			Средний уровень: объем усвоенных знаний составляет более $\frac{1}{2}$, понимает, что оставшиеся знания тоже необходимы.	70-84%	
			Высокий уровень: специальные термины употребляет осознанно и в полном соответствии с их содержанием.	85-100%	
3	Практические умения и навыки, предусмотренные основными разделами учебно-тематического плана программы	Соответствие практических умений и навыков программным требованиям, отсутствие затруднений в использовании специального оборудования, соблюдение техники безопасности	Низкий уровень: ребенок овладел менее чем $\frac{1}{2}$ объема знаний, предусмотренных программой, не обнаружил, что вторая половина ему нужна для решения задач.	55-69%	Контрольное задание. Практическая работа. Наблюдение. Публичная презентация решения кейса.
			Средний уровень: объем усвоенных знаний составляет более $\frac{1}{2}$, понимает, что оставшиеся знания тоже необходимы.	70-84%	
			Высокий уровень: ребенок освоил практически весь объем знаний, предусмотренных программой за конкретный период	85-100%	

КАДРОВОЕ ОБЕСПЕЧЕНИЕ

Обеспечение реализации программы, нацеленной на предоставление высокого качества обучения, планируется за счет штата, состоящего из высококвалифицированных специалистов, обладающих определенными компетенциями и выполняющими определенный функционал. Из них:

- учитель информатики высшей квалификационной категории – 2 чел.;
- педагог-психолог высшей квалификационной категории – 1 чел.;
- педагог-организатор высшей квалификационной категории – 1 чел.

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

Требования к зданию/помещению

Для реализации программы «Разработка на Python» помещение должно удовлетворять строительным, санитарным и противопожарным нормам.

Учебные кабинеты укомплектованы удобными рабочими местами за ученическими столами в соответствии с ростом обучающихся, состоянием их зрения и слуха.

Кабинеты информатики оборудованы в соответствии с гигиеническими требованиями, предъявляемыми к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы с ними. Используемые цифровые образовательные ресурсы, инструменты учебной деятельности (программные средства) лицензированы для использования во всём учреждении или на необходимом количестве рабочих мест. В работе используются комплекты лицензионного или свободно распространяемого программного обеспечения.

В целях организации антитеррористической защищённости охрана здания учреждения должна быть обеспечена системой наружного видеонаблюдения, пропускным режимом и штатными охранниками. Территория учреждения должна иметь периметральное ограждение и наружное освещение в темное время суток.

Материально-техническое обеспечение

Аудитории:

- аудитория для теоретических и практических занятий с необходимой ученической мебелью, пластиковой доской;
- компьютерный класс на 12 ученических и 1 учительское место;
- коворкинг-зона.

Технические средства и оборудование:

- проекционное оборудование;
- персональные компьютеры с выходом в сеть интернет и необходимым для стандартного функционирования программным обеспечением;
- принтер лазерный цветной;
- белая бумага для стандартной печати формата А4;
- маркеры для пластиковой доски.

Лицензионное программное обеспечение:

- Интегрированная среда для разработки компьютерных программ на языке PyCharm Community Edition (Версия не ниже 2020)
- Программа создания GUI-интерфейсов QtDesigner;
- Программа создания, просмотра и модификации баз данных SQLiteStudio.
- Язык программирования Python (версия не ниже 3.8) с библиотеками PyQt5, Flask и их зависимостями.

- Утилита установки дополнительных библиотек Pip.

Средства защиты:

- антибактериальные салфетки;
- антибактериальный спрей;
- огнетушитель;
- рециркулятор.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

1. Перечень литературы, необходимой для освоения программы

1.1. Перечень литературы, использованной при написании программы

1) М. Б. Лагутин. Алгоритмизация: учебное пособие – 7-е изд., испр. – М.: БИНОМ. Лаборатория знаний, 2019. – 472 с.

2) М.А. Гончарова, Н.В. Решетникова. Образовательные технологии в школьном обучении языкам программирования: Учебное пособие – Ростов н/Д: Феникс, 2020. – 264 с.

3) Шапиро И.М. Использование задач с практическим содержанием: Кн. для учителя. – М.: Просвещение, 2019. – 96 с.

1.2. Перечень литературы, рекомендованной обучающимся

1) Лутц М. Изучаем Python, том I, 5-е издание. – Пер. с англ. – СПб.: Диалектика, 2019.

2) Лутц М. Изучаем Python, том II, 5-е издание. – Пер. с англ. – СПб.: Вильямс, 2020.

3) Мэтиз Эрик. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – СПб.: Питер, 2017.

1.3. Перечень литературы, рекомендованной родителям

1) Кови С. «7 навыков высокоэффективных людей. Мощные инструменты развития личности» - Альпина Паблишер, 2019

2) Ицхак Пинтусевич «Действуй! 10 заповедей успеха» изд. Эксмо 2018 г.

3) Стивен Кови «Восьмой навык. От эффективности к величию» «Альпина Паблишер», 2020 г.

2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения программы

1) Сайт языка программирования Python <https://www.python.org>

2) Сайт IDE PyCharm Community Edition <https://www.jetbrains.com/ru-ru/pycharm/>

3) Сайт программы SQLiteStudio <https://sqlitestudio.pl/>

Приложение 1
к общеобразовательной общеразвивающей
программе «Информационные технологии.
Разработка на Python»

Примеры теоретического тестового задания

1. Что такое программирование?
2. Для чего нужен язык программирования Python?
3. Основные среды ввода и отладки компьютерных программ на языке Python.
4. Как создать проект и ввести компьютерную программу на языке Python в среде программирования PyCharm community edition?
5. Как запустить на выполнение компьютерную программу на языке Python в среде программирования PyCharm community edition?
6. Форматирование строк исходного кода (правило 4 отступов) программы на языке программирования Python.
7. Рекомендации PEP8 по оформлению исходного кода на языке программирования Python.
8. Комментирование строк исходного кода программы на языке программирования Python.
9. Оператор print языка программирования Python.
10. Оператор input языка программирования Python.
11. Типы переменных языка программирования Python.
12. Преобразование из строкового str типа в целый int и дробный float типы на языке программирования Python.
13. Преобразование из целого int и дробного float типов в строковый str тип на языке программирования Python.
14. Функция len языка программирования Python.
15. F-строки языка программирования Python.
16. Управление точностью вывода (количество знаков после запятой) дробных чисел.
17. Основные арифметические и логические операции языка программирования Python.
18. Оператор open языка программирования Python.
19. Оператор контекста with языка программирования Python.
20. Сохранение в файл и загрузка из файла результатов работы компьютерной программы на языке Python.
21. Перечислить базовые алгоритмические структуры (следование, ветвление, цикл) и кратко их назначение.
22. Алгоритмическая структура Следование.
23. Алгоритмическая структура Ветвление.
24. Алгоритмическая структура Цикл.
25. Логический тип данных языка Python.
26. Оператор if языка программирования Python.

27. Оператор while языка программирования Python.
28. Оператор for языка программирования Python.
29. Итератор range языка программирования Python.
30. Функция len языка программирования Python.
31. Выход из середины цикла командой break языка программирования Python.
32. Переход к следующей итерации цикла командой continue языка программирования Python.
33. Вложенные циклы на языке программирования Python.
34. Организация потока ввода информации в программу на языке Python при помощи циклов (без использования контейнеров данных).
35. Вывод табличной информации (без использования контейнеров данных).
36. Алгоритм поиска минимального и максимального чисел из потока вводимых данных (без использования контейнеров данных).
37. Алгоритм расчета суммы чисел из потока вводимых данных (без использования контейнеров данных).
38. Алгоритм расчета произведения чисел из потока вводимых данных (без использования контейнеров данных).
39. Алгоритм расчета среднего арифметического чисел из потока вводимых данных (без использования контейнеров данных).
40. Алгоритм обмена содержимого двух переменных.
41. Ручная сортировка трех переменных.
42. Хранение строк в языке программирования Python.
43. Основные методы работы со строками в языке программирования Python.
44. Множества set в языке программирования Python.
45. Основные методы работы с множествами в языке программирования Python.
46. Списки list в языке программирования Python.
47. Основные методы работы со списками в языке программирования Python.
48. Генераторы списков в языке программирования Python.
49. Двумерные списки, их создание и инициализация в языке программирования Python.
50. Отличия списков от массивов.
51. Кортежи tuple в языке программирования Python.
52. Основные методы работы с кортежами в языке программирования Python.
53. Словари dict в языке программирования Python.
54. Основные методы работы со словарями в языке программирования Python.
55. Организация ввода информации в список на языке Python при помощи циклов.
56. Вывод содержимого списка на экран ЭВМ в языке Python при помощи цикла.
57. Вывод содержимого двумерного списка (таблицы) на экран ЭВМ в языке Python при помощи вложенных циклов.
58. Вывод содержимого двумерного списка (таблицы) на экран ЭВМ в языке Python при помощи цикла (не вложенного).
59. Алгоритм поиска минимального и максимального элементов списка в языке

Python (встроенные функции и при помощи цикла).

60. Алгоритм расчета суммы элементов списка в языке Python (встроенная функция и при помощи цикла).

61. Алгоритм расчета произведения элементов списка в языке Python.

62. Алгоритм расчета среднего арифметического элементов списка в языке Python.

63. Сортировка элементов списка оператором sorted в языке Python.

64. Алгоритм пузырьковой сортировки.

65. Алгоритм сортировки выбором.

66. Алгоритм быстрой сортировки.

67. Алгоритм пирамидальной сортировки.

68. Ручная сортировка трех переменных.

69. Линейный поиск элемента в массиве (списке).

70. Бинарный поиск элемента в массиве (списке).

71. Поиск k-го по величине элемента в массиве (списке).

72. Создание независимой копии списка в языке Python (встроенная функция-метод и при помощи циклов).

73. Распаковка и запаковка списков в языке программирования Python.

74. Функция join в языке программирования Python.

75. Функция split в языке программирования Python.

76. Что такое вспомогательный алгоритм (подпрограмма)?

77. Как создать подпрограмму-функцию на языке программирования Python?

78. Как передать аргументы (вспомогательные параметры) функции на языке программирования Python?

79. Как вернуть вычисленное функцией значение в вызывающую (главную) функцию на языке программирования Python?

80. Локальные переменные функции в языке программирования Python.

81. Глобальные переменные программы в языке программирования Python.

82. Статические переменные функции в языке программирования Python.

83. Ключевое слово global языка программирования Python.

84. Особенности передачи списков в качестве аргументов функции в языке программирования Python.

85. Возврат кортежей как результата работы функции в языке программирования Python.

86. Передача списка аргументов переменной длины в функцию в языке программирования Python (использование *args).

87. Передача именованного списка аргументов переменной длины в функцию в языке программирования Python (использование **kwargs).

88. Функции высшего порядка в языке программирования Python.

89. Лямбда-функции в языке программирования Python.

90. Лямбда map в языке программирования Python.

91. Лямбда filter в языке программирования Python.

92. Лямбда zip в языке программирования Python.
93. Лямбда enumerate в языке программирования Python.
94. Аннотация функций в языке программирования Python.
95. Функции help в языке программирования Python.
96. Функция dir в языке программирования Python.
97. Встраивание help-комментария в функцию в языке программирования Python.
98. Импорт функций из внешних модулей в языке программирования Python.
99. Библиотека math функций в языке программирования Python.
100. Библиотека random функций в языке программирования Python.
101. Библиотека datetime функций в языке программирования Python.
102. Модуль PyQt в языке программирования Python.
103. Модуль sqlite в языке программирования Python.
104. Модуль request в языке программирования Python.
105. Модуль flask в языке программирования Python.
106. Что такое объектно-ориентированная технология программирования?
107. Что такое класс в ООП?
108. Что такое объект в ООП?
109. Как создать класс в языке программирования Python?
110. Как создать объект в языке программирования Python?
111. Что такое инкапсуляция в ООП?
112. Что такое наследование в ООП?
113. Что такое полиморфизм в ООП?
114. Как создать наследника класса в языке программирования Python?

Примеры тестовых практических заданий

Примерный внешний вид результата работы компьютерной программы практического тестового задания.

Вариант N<номер варианта>

Выполнил учащийся <ФИО> группы <группа>

Расчет формулы <формула> или задание

a= <значение a>

b= <значение b>

Ответ: c= <вычисленное значение>

Для выхода из программы нажмите Enter

<задержка выхода из программы>

1. Найти количество натуральных делителей числа A.
2. Вычислить НОД (наибольший общий делитель) двух натуральных чисел A и B.
3. Вычислить НОК (наименьшее общее кратное) двух натуральных чисел A и B.
4. Написать программу генерации n (число n вводится с клавиатуры) псевдослучайных чисел.

5. Заданное натуральное число n , не превосходящее 1000, записать прописью, т.е. вывести соответствующее количественное числительное.
6. Написать программу разложения натурального числа $n < 1000$ на простые сомножители.
7. Написать программу, определяющую является ли данное натуральное число n простым или нет.
8. Дано натуральное число $n < 10000$. Переставить его десятичные цифры так, чтобы получить максимально возможное число, записанное теми же цифрами.
9. Найти сумму всех делителей натурального числа n .
10. Найти произведение всех простых делителей числа n , взятых по одному разу.
11. Найти все натуральные числа, не превосходящие заданного натурального числа n и делящиеся на каждую из своих цифр.
12. Найти все простые несократимые дроби, знаменатель которых не превышает натурального числа n .
13. Даны натуральные числа a и b . Найти все натуральные числа x , меньшие a и взаимно простые с b .
14. Дано натуральное число n . Найти произведение его цифр.
15. Дано натуральное число $99 < n < 1000$. Найти все натуральные числа, образованные перестановками цифр числа n .
16. Дано натуральное число n . Найти наибольшую и наименьшую его цифры.
17. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм пузырьковой сортировки).
18. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм пузырьковой сортировки).
19. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм сортировки выбором).
20. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм сортировки выбором).
21. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм быстрой сортировки).
22. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм быстрой сортировки).
23. Упорядочить элементы массива $A[n]$ по возрастанию (алгоритм пирамидальной сортировки).
24. Упорядочить элементы массива $A[n]$ по убыванию (алгоритм пирамидальной сортировки).
25. Подсчитать в массиве $A[n]$ количество различных элементов.
26. Найти в массиве $A[n]$ k -й по величине элемент.
27. Найти сумму первых по величине k элементов массива $A[n]$.
28. Найти произведение первых по величине k элементов массива $A[n]$.
29. Упорядочить элементы массива $A[n]$ на четных индексах по возрастанию, а на нечетных по убыванию.

30. Упорядочить элементы массива $A[n]$ на четных индексах по убыванию, а на нечетных по возрастанию.
31. Найти в массиве $A[n]$ наиболее длинную цепочку одинаковых подряд стоящих элементов.
32. Написать программу сложения «сверхдлинных» неотрицательных целых чисел, каждое из которых представляется в виде массива цифр.
Замечание: ограничить максимальную длину числа 100 цифрами.
33. Написать программу вычитания «сверхдлинных» неотрицательных целых чисел, каждое из которых представляется в виде массива цифр.
Замечание: ограничить максимальную длину числа 100 цифрами.
34. Написать программу деления нацело «сверхдлинных» неотрицательных целых чисел, каждое из которых представляется в виде массива цифр.
Замечание: ограничить максимальную длину числа 100 цифрами.
35. Написать программу вычисления остатка от деления «сверхдлинных» неотрицательных целых чисел, каждое из которых представляется в виде массива цифр.
Замечание: ограничить максимальную длину числа 100 цифрами.
36. В целочисленной прямоугольной матрице числа, кратные заданному числу m , вводимому с клавиатуры, заменить частным от деления на m , а остальные числа заменить их остатками от деления на m .
37. В квадратной матрице порядка n найти сумму всех тех элементов, сумма индексов которых равна n .
38. Определить, является ли заданная квадратная матрица симметричной относительно главной диагонали.
39. Найти число элементов в каждой из нечетных строк матрицы, равных заданному числу, вводимому с клавиатуры.
40. Преобразовать данную прямоугольную матрицу так, чтобы все элементы в ее строках были записаны в обратном порядке.
41. Вычислить количество и сумму положительных элементов в нижнем левом треугольнике данной квадратной матрицы, включая диагональные элементы.
42. Дана вещественная квадратная матрица порядка n . Преобразовать эту матрицу по правилу: строку с номером k сделать столбцом с номером k , а столбец с номером k сделать строкой с тем же номером.
43. Положительные элементы первой строки прямоугольной матрицы умножить на первый элемент этой же строки, а отрицательные - на последний ее элемент; то же самое проделать с остальными строками.
44. Заменить все элементы строки с номером s и столбца с номером k прямоугольной матрицы на противоположные по знаку (элемент, стоящий на пересечении выбранной строки и выбранного столбца, не изменять).
45. Из квадратной матрицы получить другую матрицу путем исключения диагональных элементов и вычитания из каждого элемента следа исходной матрицы.

46. Составить программу создания арифметического квадрата: в нем первый столбец и первая строка заполнены единицами, а каждый из остальных элементов равен сумме своих соседей сверху и слева.
47. Создать проект, реализующий калькулятор (предусмотреть операции сложения, вычитания, умножения и деления операндов).
48. Создать проект, реализующий модель записной книжки емкостью до 10 записей (предусмотреть операции внесения новых записей, удаление старых, просмотр записей).
49. Создать проект, реализующий календарь (предусмотреть операции вывода календаря на произвольную неделю, месяц и год).
50. Создать проект, реализующий модель часов (предусмотреть операции установки часов на любое заданное время, будильник).
51. Создать проект, моделирующий движение системы из 3 материальных точек под действием гравитационных сил (для введенного момента времени t программа должна распечатать координаты всех тел в пространстве).
52. Создать проект, моделирующий движение 3 шаров на бильярдном столе. Учесть тот факт, что шары обладают конечными размерами и массой и при столкновении друг с другом и со стенками стола испытывают упругое соударение. Для данного момента времени t программа должна распечатать координаты шаров относительно поверхности бильярдного стола.
53. Создать проект, моделирующий магнитофон. (Предусмотреть операции перемотки, воспроизведения, включения и выключения магнитофона).
54. Создать проект, моделирующий лифт в здании из 9 этажей. Лифт может перевозить людей с этажа на этаж и при этом он может останавливаться, не доходя до этажа назначения если по пути следования возник еще один вызов в попутном направлении (в этом случае лифт после забора новых пассажиров продолжает движение), либо по техническим причинам (в этом случае лифт останавливается до проведения «ремонта»). В лифте могут поместиться не более 4 человек.
55. Составить проект программы для игры в «Крестики-нолики» на поле размером 3×3 .
56. Создать проект, моделирующий полет ракеты, с учетом воздействия на нее гравитационных полей трех объектов (Земля, Луна, Солнце). Для данного момента времени t программа должна распечатать координаты ракеты в околоземном пространстве. У ракеты имеются тяговые двигатели, а также рулевая система, позволяющие менять вектор скорости.
57. Создать проект, моделирующий 4-х цилиндровый двигатель внутреннего сгорания. (Предусмотреть операции включения, выключения двигателя, наличие топлива, ускорение и замедление работы двигателя).
58. Создать проект, моделирующий телевизор. (Предусмотреть операции включения, выключения, переключения с канала на канал, настройки телевизора).

59. Кошка может находиться в следующих состояниях - спит, мурлыкает, охотится, балуется.

Состояние кошки меняется по следующим правилам:

- 1) после удачной охоты кошка балуется;
- 2) набаловавшись, кошка или мурлыкает, или спит, или снова охотится;
- 3) помурлыкав, кошка засыпает или балуется;
- 4) проснувшись, кошка охотится или мурлыкает.

Из одного состояния в разрешенное другое состояние кошка переходит случайным образом. Создать проект, моделирующий поведение кошки.

60. Создайте проект для выполнения арифметических действий с комплексными числами: сложение, вычитание, умножение, деление, представление в тригонометрической форме, ввод комплексного числа, печать комплексного числа (использование готовых функций работы с комплексными числами запрещено).

61. Создайте проект для нахождения математического ожидания, дисперсии и среднеквадратического отклонения последовательности чисел.

62. Создайте проект для выполнения арифметических действий с обыкновенными дробями: сложение, вычитание, умножение, деление, сокращение, ввод обыкновенной дроби и печать обыкновенной дроби.

63. Написать программу генерации символами псевдографики изображения лабиринта размером $N \times N$ залов (число N заранее не известно).

64. Написать программу деления натуральных чисел a и b с точностью до n десятичных знаков.

Замечание: использование вещественных чисел запрещено.

65. Написать программу вычисления факториала числа n .

Замечание: использование вещественных чисел запрещено.

66. Треугольник задан координатами своих вершин. Определить, является ли точка лежащей внутри треугольника?

67. Написать программу, вычисляющую число Π при помощи метода Монте-Карло (генератора случайных чисел).

68. Написать программу разложения натурального числа $n < 1000$ на простые сомножители.

69. Написать программу, определяющую является ли данное натуральное число n простым или нет.

70. Найти сумму всех делителей натурального числа n .

71. Составить проект расчета значения определенного интеграла пользовательской функции (запрограммировать в виде функции в программе) с заранее заданной точностью методом трапеций.

72. Создать проект для нахождения целочисленных координат всех точек, находящихся внутри треугольника ABC по заданным координатам его вершин

73. Даны n натуральных чисел ($n > 5$). Найти их наибольший общий делитель.

74. Даны n натуральных чисел ($n > 5$). Найти их наименьшее общее кратное.

75. Дано предложение S, состоящее из слов, разделенных пробелами и знаками препинания. Вывести все уникальные слова предложения в порядке неубывания их длин.
76. Дано предложение S, состоящее из слов, разделенных пробелами и знаками препинания. Вывести все уникальные слова предложения.
77. Дано предложение S, состоящее из слов, разделенных пробелами и знаками препинания. В нем только два слова одинаковые. Найти эти слова.