

Муниципальное бюджетное общеобразовательное учреждение
Тёпловская средняя общеобразовательная школа
Центр образования естественно-научной и технологической направленности «Точка Роста»

Принято решением
Педагогического совета
№ 5
от «30» августа 2024года



«Утверждено»
директор МБОУ Тёпловская СОШ
/Бойдарико В.Г./
приказ №40
«30» августа 2024 года



**Общеобразовательная общеразвивающая
дополнительного образования
программа технической направленности
«Юный программист»**

**Целевая аудитория: 8-9 классы
Срок реализации: 1 год**

Составитель:
учитель информатики
Федоров Александр Николаевич

п. Тёплое
2024-2025 учебный год

Оглавление

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА.....	3
Актуальность.....	3
Аспект новизны.....	3
Цель.....	3
Задачи.....	3
Образовательные.....	3
Воспитательные	3
Развивающие	3
Режим занятий.....	4
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ КУРСА ДЕЯТЕЛЬНОСТИ	4
Метапредметные	4
Личностные	5
Предметные	5
3. СОДЕРЖАНИЕ КУРСА.....	5
Раздел 1. «Создание приложений на языке программирования Python»	5
Раздел 2. «Обработка структур данных на языке программирования Python»	6
Раздел 3. «Разработка алгоритмов и программ на языке программирования Python»	6
4. УЧЕБНЫЙ ПЛАН.....	6
5. КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ	8
6. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ	9
Формы аттестации	9
Методическое обеспечение программы.....	9
Материально-техническое обеспечение.....	12
Информационное обеспечение	12
7. ОЦЕНОЧНЫЕ ПРОЦЕДУРЫ	13
Критерии оценивания	13
Единая шкала критериев оценки проектов.....	13
Единая шкала критериев оценки практической работы.....	15
8. СПИСОК ЛИТЕРАТУРЫ	17
Литература для обучающихся и родителей	17
Литература для педагогов.....	17
Интернет-ресурсы	17
Приложение 1. Входной контроль в форме теста.....	18
Приложение 2. Примерные темы проектов для итоговой аттестации.....	19
Приложение 3. Календарный учебный график на 2024-2025 учебный год	20

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Общеобразовательная общеразвивающая программа дополнительного образования «Юный программист» (далее – Программа) относится к **технической направленности (базовый уровень)**.

Актуальность

В школьном курсе информатики вопросы программирования рассматриваются лишь в ознакомительном плане и на это выделяется недостаточное количество часов, как следствие – формальное восприятие учащимися основ современного программирования и низкие показатели успешности по заданиям из разделов «Элементы теории алгоритмов» и «Программирование» по результатам Государственной итоговой аттестации в виде ОГЭ и ЕГЭ. Образовательная программа «Юный программист» направлена на устранение данного пробела. Это особенно актуально в новом информационном обществе, где пользователей очень часто не устраивают возможности программ и им хочется адаптировать приложения для своих конкретных потребностей.

В процессе обучения у учащихся формируются навыки программирования, представление о профессии программиста, механизм работы и устройство операционной системы Windows. Знания и умения, приобретённые в результате освоения программы, являются фундаментом для дальнейшего совершенствования мастерства в области программирования, а также помогут учащимся в дальнейшем обучении в вузах и в профессиональной деятельности.

Аспект новизны

Аспект новизны заключается в том, что:

1. При организации обучения по программе используется современный комплекс программного обеспечения, состоящий из актуальных версий сред программирования (Pascal ABC, Python);
2. Обучающиеся получают опыт разработки программного обеспечения, которое могут использовать в повседневной жизни.
3. Обучающиеся получают опыт разработки программного обеспечения, которое могут использовать при разработке и программировании сайтов, а также в процессе программирования робототехнических устройств.

Отличительной особенностью Программы является практико-ориентированный характер (теоретическая часть составляет 1/3 от общего учебного времени), а также то, что итоговыми результатами деятельности являются приложения на языке Python.

Цель

развитие навыков алгоритмического мышления учащихся, познавательных, интеллектуальных и творческих способностей учащихся, формирование базовых понятий структурного программирования и способствовать развитию интереса обучающихся к программированию посредством разработки приложений.

Задачи

Образовательные

- Овладение базовыми понятиями объектно-ориентированного программирования и применение их при создании проектов в среде программирования Python;
- Овладение базовыми понятиями объектно-ориентированного программирования и применение их при создании приложений с использованием языка программирования Python;
- Приобщение обучающихся к новым технологиям, способным помочь им в реализации собственного творческого потенциала;
- Развитие познавательной деятельности учащихся в области новых информационных технологий;
- Совершенствование навыков работы на компьютере и повышение интереса к программированию.

Воспитательные

- Формирование культуры и навыки сетевого взаимодействия;
- Способствование развитию творческих способностей и эстетического вкуса подростков;
- Способствование развитию коммуникативных умений и навыков обучающихся.

Развивающие

- Способствование развитию логического мышления, памяти и умению анализировать;
- Создание условия для повышения самооценки обучающегося, реализации его как личности;
- Формирование потребности в саморазвитии;
- Способствование развитию познавательной самостоятельности.
- Программа адресована обучающимся 13-15 лет образовательных организаций. Образовательный процесс организуется в соответствии с учебным планом в творческих объединениях, сформированных из учащихся одной возрастной категории (состав группы: постоянный).

Режим занятий

Программа рассчитана на 1 год обучения и предполагает изучение языка программирования Python.

Режим занятий: одно занятие в неделю, продолжительностью два академических часа; общее количество часов: 35 часов.

2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ КУРСА ДЕЯТЕЛЬНОСТИ

Метапредметные

- Умение самостоятельно определять цели обучения, ставить и формулировать новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности.
- Самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера.
 - Умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач.
 - Умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией.
 - Умение оценивать правильность выполнения учебной задачи, собственные возможности её решения.
 - Строить схему, алгоритм действия, исправлять или восстанавливать неизвестный ранее алгоритм на основе имеющегося знания об объекте, к которому применяется алгоритм;
 - Смысловое чтение. Обучающийся сможет: находить в тексте требуемую информацию (в соответствии с целями своей деятельности); ориентироваться в содержании текста, понимать целостный смысл текста, структурировать текст; устанавливать взаимосвязь описанных в тексте событий, явлений, процессов.
 - Работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение.
 - Формирование и развитие компетентности в области использования информационно-коммуникационных технологий.
 - Выбирать, строить и использовать адекватную информационную модель для передачи своих мыслей средствами естественных и формальных языков в соответствии с условиями коммуникации.
 - Выделять информационный аспект задачи, оперировать данными, использовать модель решения задачи.
 - Самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера.
 - Определять действия в соответствии с учебной и познавательной задачей, планировать свои действия в соответствии с поставленной задачей и условиями её решения, осуществлять пошаговый контроль своей познавательной деятельности, определять потенциальные затруднения при решении практической задачи и находить средства для их устранения, осознавать качество и уровень усвоения материала по модулям;
 - Строить рассуждение от общих закономерностей к частным явлениям и от частных явлений к общим закономерностям; строить рассуждение на основе сравнения предметов и явлений, выделяя при

этом общие признаки;

- Полученные знания помогут социализации, вовлекут в проектную и научно-исследовательскую деятельность.

Личностные

- формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых творческих учебных проектов;
- 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, участия 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;
- способность увязать учебное содержание с собственным жизненным опытом, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества.
- Формирование коммуникативной компетентности в процессе образовательной, учебно-исследовательской, творческой и других видов деятельности;

Предметные

- знать значение понятий «программирование», «компоненты приложения», «дизайн приложения», «блоки программирования»;
- уметь читать и выполнять трассировку программы на языке программирования;
- знать особенности написания программ на языках программирования Python;
- знать как описываются структуры цикла и ветвления на языках программирования Python;
- уметь написать программу на языке программирования Python, содержащую линейную или циклическую структуру и содержащую графические операторы.
- знать особенности обработки последовательностей (строки, массивы, списки) на языке программирования Python
- знать как подключаются модули в языке программирования Python
- уметь написать программу на языке программирования Python, содержащую структуры цикла и ветвления и обрабатывающую последовательности.
- уметь читать и выполнять трассировку программы на языке программирования;

3. СОДЕРЖАНИЕ КУРСА

Раздел 1. «Создание приложений на языке программирования Python»

Особенности Питона. Дзен Питона. Сфера применения Питона. Структура программы на Python. Стил программирования для Python.

Вывод информации на экран. Оператор вывода данных. Примеры использования оператора вывода. Разноуровневые задания по теме «Вывод информации на экран»

Переменные величины. Ввод данных в программу. Инструкция присваивания. Переменная величина, характеристики переменной. Понятие объекта и типа. Преобразование типов. Арифметические операции.

Целочисленная арифметика. Действительные числа. Библиотека math. Разноуровневые задания по теме «Переменные величины. Ввод данных в программу. Инструкция присваивания»

Условный оператор. Инструкция if. Синтаксис условной инструкции. Вложенные условные инструкции. Операторы сравнения. Тип данных bool. Логические операторы. Каскадные условные инструкции. Разноуровневые задания по теме «Условный оператор. Инструкция if»

Цикл for в языке программирования Python. Функция range. Настройка функции print().

Разноуровневые задания по теме «Цикл for в языке программирования Python»

Цикл While. Инструкции управления циклом. Множественное присваивание. Разноуровневые задания по теме «Цикл While»

«Обрабатываем» натуральное число. Выделение цифр числа. Определение n-ой цифры числа справа или слева. Определение суммы цифр числа. Определение максимальной или минимальной цифры числа. Определение номера максимальной или минимальной цифры числа. Разноуровневые задания по теме «Обрабатываем натуральное число»

Последовательности: списки. Задание и вывод на экран элементов списка. Обращение к элементам списка. Методы split и join. Генераторы списков. Разноуровневые задания по теме «Последовательности: списки».

Создание консольных приложений. Разбор алгоритмов игр «Чет или нечет?», «Кубик», «Отгадай число», «Карты», «Викторина», «Предметы на столе». Создание игр «Чет или нечет?», «Кубик», «Отгадай число», «Карты», «Викторина», «Предметы на столе».

Раздел 2. «Обработка структур данных на языке программирования Python»

Типовые задачи обработки списков. Расчёты. Поиск и отбор нужных элементов. Работа с максимальными, минимальными элементами. Проверка соответствия списка в целом некоторому условию. Разноуровневые задания по теме «Типовые задачи обработки списков»

Типовые задачи обработки набора чисел. Определение максимального значения в наборе. Определение максимального значения в наборе чисел, удовлетворяющих определённому условию. Нахождение второго по величине максимального числа набора. Нахождение количества максимальных элементов. Разноуровневые задания по теме «Типовые задачи обработки набора чисел»

Сортировка массивов. Встроенные возможности сортировки языка программирования. Сортировка по нескольким критериям (уровням). Двоичный поиск в упорядоченном массиве.

Двумерные массивы (матрицы). Основные алгоритмы обработки двумерных массивов (матриц): заполнение двумерного массива случайными числами и с использованием формул; вычисление суммы элементов, минимума и максимума строки, столбца, диапазона; поиск заданного значения.

Последовательности: строки. Понятие строки. Функции работы со строками. Срезы. Методы find, rfind, replace, count. Практика: Разноуровневые задания по теме «Последовательности: строки»

Раздел 3. «Разработка алгоритмов и программ на языке программирования Python»

Алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел. Разбиение записи натурального числа в позиционной системе с основанием, меньшим или равным 10, на отдельные цифры. Разложение натурального числа на простые сомножители. Алгоритм проверки натурального числа на простоту.

Разбиение задачи на подзадачи. Вспомогательные алгоритмы (подпрограммы, процедуры, функции). Параметры как средство изменения результатов работы подпрограммы. Результат функции. Логические функции.

Рекурсия. Рекурсивные подпрограммы (процедуры, функции). Условие окончания рекурсии (базовые случаи). Применение рекурсии для перебора вариантов.

Динамическое программирование. Задачи, решаемые с помощью динамического программирования: вычисление функций, заданных рекуррентной формулой; подсчёт количества вариантов, выбор оптимального решения.

4. УЧЕБНЫЙ ПЛАН

№	Тема (раздел)	Количество часов		
		Всего	Теория	Практика
1	Создание приложений на языке программирования Python	11	4	7
1.1	Введение в язык программирования Python.	1	1	
1.2	Вывод информации на экран.	1		1
1.3	Переменные величины. Ввод данных в программу. Инструкция присваивания.	1		1
1.4	Целочисленная арифметика	2		2
1.5	Условный оператор. Инструкция if	2	1	1

1.6	Цикл for в языке программирования Python	2	1	1
1.7	Цикл While	2	1	1
2	Обработка структур данных на языке программирования Python	12	4	8
2.1	«Обрабатываем» натуральное число	1		1
2.2	«Обрабатываем» натуральное число	1		1
2.3	Последовательности: списки.	1	1	
2.4	Типовые задачи обработки списков	2	1	1
2.5	Типовые задачи обработки набора чисел	2	1	1
2.6	Сортировка в массиве	1		1
2.7	Двумерные массивы	1	0,5	0,5
2.8	Последовательности: строки.	3	0,5	2,5
3	Разработка алгоритмов и программ на языке программирования Python	12	3	9
3.1	Алгоритмы обработки чисел	1	1	
3.2	Разбиение задачи на подзадачи.	2	0,5	1,5
3.3	Рекурсия.	1	0,5	0,5
3.4	Динамическое программирование.	3	1	2
3.5	Решение комплексных задач	2		2
3.6	Создание консольных приложений	2		2
3.7	Презентация проектов, выполненных в рамках курса.	1		

5. КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

№	Дата проведения занятия	Тема занятия	Кол-во часов	Форма контроля
1.	04.09	Введение в язык программирования Python.	1	Практическая работа
2.	11.09	Вывод информации на экран.	1	Практическая работа
3.	18.09	Переменные величины. Ввод данных в программу. Инструкция присваивания.	1	Практическая работа
4.	25.09	Целочисленная арифметика	1	Практическая работа
5.	02.10	Целочисленная арифметика	1	Практическая работа
6.	09.10	Условный оператор. Инструкция if	1	Практическая работа
7.	16.10	Условный оператор. Инструкция if	1	Практическая работа
8.	23.10	Цикл for в языке программирования Python	1	Практическая работа
9.	06.11	Цикл for в языке программирования Python	1	Практическая работа
10.	13.11	Цикл While	1	Практическая работа
11.	20.11	Цикл While	1	Практическая работа
12.	27.11	«Обрабатываем» натуральное число	1	Практическая работа
13.	04.12	«Обрабатываем» натуральное число	1	Практическая работа
14.	11.12	Последовательности: списки.	1	Практическая работа
15.	18.12	Типовые задачи обработки списков	1	Практическая работа
16.	25.12	Типовые задачи обработки списков	1	Практическая работа
17.	15.01	Типовые задачи обработки набора чисел	1	Практическая работа
18.	22.01	Типовые задачи обработки набора чисел	1	Практическая работа
19.	29.01	Сортировка в массиве	1	Практическая работа
20.	05.02	Двумерные массивы	1	Практическая работа
21.	12.02	Последовательности: строки.	1	Практическая работа
22.	19.02	Последовательности: строки.	1	Практическая работа
23.	26.02	Последовательности: строки.	1	Практическая работа
24.	05.03	Алгоритмы обработки чисел	1	Практическая работа
25.	12.03	Разбиение задачи на подзадачи.	1	Практическая работа
26.	19.03	Разбиение задачи на подзадачи.	1	Практическая работа
27.	02.04	Рекурсия.	1	Практическая работа
28.	09.04	Динамическое программирование.	1	Практическая работа
29.	16.04	Динамическое программирование	1	Практическая работа
30.	23.04	Динамическое программирование	1	Практическая работа
31.	30.04	Решение комплексных задач	1	Практическая работа
32.	07.05	Создание консольных приложений	1	Практическая работа
33.	14.05	Создание консольных приложений	1	Практическая работа
34.	21.05	Презентация проектов, выполненных в рамках курса.	1	Практическая работа

6. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Формы аттестации

- Входной контроль осуществляется в начале учебного года в форме текста и определяет начальный уровень знаний ученика о программировании
- Итоговая аттестация проходит в форме защиты проектов в конце года.

Кроме того, дополнительными формами аттестации могут служить:

- зачетное занятие;
- выступление на конференции;
- участие в конкурсах различного уровня;
- участие в олимпиадах различного уровня.

Формы отслеживания и фиксации образовательных результатов: журнал посещаемости, архив готовых работ обучающихся, размещённых на сайте объединения, портфолио работ участников творческого объединения (в электронном виде).

Формы предъявления и демонстрации образовательных результатов: защита творческих работ, итоговый отчёт, портфолио (в электронном виде).

Методы обучения:

словесный, наглядный практический, объяснительно-иллюстративный, частично-поисковый, исследовательский проблемный; игровой, проектный.

Методы воспитания: убеждение, поощрение, упражнение, стимулирование, мотивация

Формы организации образовательного процесса: индивидуальная, индивидуально-групповая и групповая.

Формы организации учебного занятия: защита проектов, практическое занятие.

Педагогические технологии: технология индивидуализации обучения, технология группового обучения, технология коллективного взаимообучения, технология программированного обучения, технология модульного обучения, технология блочно-модульного обучения, технология дифференцированного обучения, технология разноуровневого обучения, технология развивающего обучения, технология проблемного обучения, технология дистанционного обучения, технология исследовательской деятельности, технология проектной деятельности, технология игровой деятельности, коммуникативная технология обучения, технология портфолио, здоровьесберегающая технология, кейс-технология.

При изучении возможностей программного обеспечения рекомендуется придерживаться следующей **схемы организации и проведения занятия**.

№	Этап занятия	Форма организации и проведения	Примерное время минут)
1.	Актуализация ранее полученных знаний, постановка проблемы, решаемой на занятии	Эвристическая беседа	2
2.	Изучение теории	Кейс-метод, лекция с элементами проблемного обучения.	10
3.	Разбор задания базового уровня сложности.	Кейс-метод, лекция с элементами проблемного обучения.	5
4.	Самостоятельное решение учащимися разноуровневых заданий, подготовка минипроектов	Разноуровневые задания по теме занятия.	20
5.	Подведение итогов занятия, рефлексия.	Беседа	3

Методическое обеспечение программы

№ п/п	Темы программы	Формы организации и проведения занятий/ Форма организации деятельности	Методы и приемы организации учебно-воспитательного процесса	Дидактический материал, техническое оснащение	Вид и форма контроля, форма предъявления результатов
1.	Введение в язык программирования Python.	практическое занятие/ индивидуально-групповая	наглядный практический, объяснительно-иллюстративный, частично-поисковый	Карточки для выполнения заданий.	Текущий контроль Практическая работа

			поисковый		
35.	Презентация проектов, выполненных в рамках курса.	Защита проектов / индивидуальная	проектный.	Презентации	Итоговый контроль Проект

Дидактические материалы: раздаточные материалы, содержащие задания и упражнения для выполнения практических заданий, а также кейс-материалы с примерами игр.

Материально-техническое обеспечение

1. Столы ученические: 4 шт.
2. Столы компьютерные: 8 шт.
3. Стулья: 8 шт.
4. Кресла компьютерные: 8 шт.
5. Интерактивная доска: 1 шт.
6. Компьютеры: 7 шт.
7. Мультимедийный проектор: 1 шт.
8. Многофункциональное устройство: 1 шт.
9. Аудиосистема: 1 шт.
10. Сетевое оборудование для обеспечения доступа к сети Интернет (коммутатор, роутер, свитч)
11. Программное обеспечение (Windows, MS Office, Google Chrome и т.д.)
12. Среда разработки Python
13. Среда разработки PascalABC
14. Среда разработки VisualStudio Code

Информационное обеспечение

Цифровые ресурсы

2. <https://pythonworld.ru/>
3. <https://devpractice.ru/python/>
4. <https://code.tutsplus.com/ru/tutorials/search/python>
5. <https://habrahabr.ru/hub/python/>
6. <https://geekbrains.ru/python>
7. <https://ru.wikibooks.org/wiki/Python>
8. <https://www.learnpython.org/ru/>
9. <https://dvmn.org/>
10. <https://www.coursera.org/courses?query=python&languages=ru>
11. <https://www.youtube.com/playlist?list=PLDyvV36pndZH2fwjegjYvHQrLrWLZ2wyJ>

Кадровое обеспечение: занятия по курсу проводит педагог дополнительного образования, имеющий высшую или первую квалификационную категорию, а также опыт преподавания программирования в образовательных организациях.

7. ОЦЕНОЧНЫЕ ПРОЦЕДУРЫ

Цель аттестации: выявление уровня развития способностей и личностных качеств детей и их соответствия прогнозируемым результатам дополнительной общеобразовательной программы.

Оценочные материалы.

При определении уровня освоения обучающимся программы используется 10-ти балльная система оценки освоения программы:

- минимальный уровень – 1 балл,
- средний уровень – от 2 до 5 баллов,
- максимальный уровень – от 6 до 10 баллов.

Критерии оценивания

Примеры оценивания							
№ п/п	Ф.И. воспитанника	Показатели					Итогов ый балл
		Теоретическая подготовка учащегося	Практическая подготовка учащегося	Умения и навыки учащегося			
				Интеллектуальные умения	Учебно- коммуникативные умения	Учебно- организационные умения	
		а) теоретические знания; б) владение специальной терминологией	а)практические умения и навыки; б) решение задач	а) умение подбирать и анализировать специальную литературу; б) умение осуществлять проектную работу.	а) умение слушать и слышать педагога	а) умение организовать рабочее место; б) навыки соблюдения правил безопасности.	

Единая шкала критериев оценки проектов

Колич ество баллов	Актуальность проекта, самостоятельность	Теоретическое обоснование и практическая значимость	Структура и оформление результатов	Грамотность и методика исследования	Презентация проекта
0	Задание не выполнено				
1	Ученик выполнил задание. С помощью учителя определена проблема и / или плохо обосновал ее актуальность (использована традиционная тематика, низкий уровень	В проекте нет полного теоретического обоснования всех положений, концепций; работа не имеет практической значимости	Учеником не выдержана структура работы и / или плохо упорядочена, оформление работы не соответствует	Ученик допустил значительное количество орфографических и синтаксических ошибок, стилистических погрешностей (не	Ученик при презентации не использовал никаких наглядно-иллюстративных средств, плохо выстроил логику выступления, не

	новизны); сформулирована цель и задачи проекта (цель не диагностична, задачи не взаимосвязаны и плохо обеспечивают достижение цели); оригинальные идеи отсутствуют или принадлежат научному руководителю; низкая доля самостоятельности в реализации работы на всех этапах проекта	или не описана. Новые научные результаты отсутствуют или принадлежат научному руководителю (ученик плохо может объяснить значимость полученных результатов)	формальным требованиям и требуемому объему (слишком велик или мал). Некорректное оформление сносок, ссылок на используемую литературу или их отсутствие. Низкая культура оформления	соблюден научный стиль изложения), наличие опечаток, сокращений. Плохо разработаны критерии и показатели реализации проекта, методы их диагностики; личный вклад автора в разработку средств, методов незначителен (заимствован или разработан учителем); результаты описаны при значительной помощи учителя	смог ответить на дополнительные вопросы (и / или не уложился в регламент выступления)
2-5	Ученик справился с заданием. Самостоятельно или при небольшой помощи учителя Определил проблему, сформулировал цель и задачи проекта (имеются незначительные неточности, замечания), выбрана тематика по актуальным, перспективным направлениям, имеются собственные оригинальные идеи; большая доля самостоятельности в реализации на всех этапах проекта	В проекте не до конца дано теоретическое обоснование всех положений проекта, продукт проекта имеет небольшую значимость для решения отдельных практически х задач (может быть использован а в учебных целях)	Учеником не до конца выдержана структура проекта и его оформление, текст разделен на смысловые части. Объем слегка больше или меньше требуемого. Ссылки и цитаты не все корректно оформлены	Ученик допустил незначительное количество грамматических ошибок и / или стилистических погрешностей. Достаточно хорошо разработаны критерии и показатели реализации проекта, методы их диагностики, есть неточности; личный вклад автора в разработку средств и методов исследования более половины (адаптирована или создана при помощи учителя); результаты описаны при незначительной помощи учителя или самостоятельно	Ученик не адекватно применил наглядно-иллюстративные средства, допустил нарушения в логике выступления, ответил на все дополнительные вопросы, хотя были не точности в ответах, и аргументации (даны неполные ответы), соблюден регламент
6-10	Ученик справился с заданием. Самостоятельно или при небольшой помощи учителя определил проблему, верно	В проекте представлена информация об объекте проектирования, дано теоретическое	Ученик полностью выдержал структуру проекта, прослеживает с я логика	Ученик не допустил грамматических ошибок и стилистических погрешностей (соблюден	Ученик выстроил логику выступления, оптимально использовал наглядно-иллюстративные

	определил цель (способствующая решению проблемы, диагностична), задачи взаимосвязаны, обеспечивают достижение цели, выбрана тематика по актуальным и перспективным направлениям и имеющая практическое применение, оригинальные идеи значительны. Высокая доля самостоятельности в реализации работы на всех этапах проекта	обоснование всех положений проекта, продукт имеет значимость для решения отдельных практически х задач. Новые научные результаты принадлежат учащемуся и их значимость значительна	рассуждений при переходе от одной части к другой, оформление соответствует формальным требованиям, правильное оформление ссылок и цитат, соблюден необходимый объем работы. Высокая культура оформления	научный стиль изложения); логичность, четкость и последовательность изложения информации. Представлены ожидаемые результаты от реализации проекта, критерии и показатели, методы их диагностики. Методика исследования хорошо прописана, самостоятельно разработана или при небольшой поддержки учителя	средства раскрывающие тему, четко и лаконично ответил на все заданные вопросы, соблюден регламент, речь выступающего соответствует правилам публичного выступления
--	---	--	---	---	--

Единая шкала критериев оценки практической работы

Ко л- во бал лов	Критерии оценивания			
	Полнота	Работа с оборудованием	Отчет о проведенной работе	Срок сдачи работы
0	Задание не выполнено или не справился			
1	Ученик выполнил задание не полностью, но этой части работы хватает, чтобы получить правильные результаты и выводы	Ученик смог собрать установку для проведения опыта с помощью учителя, выполнил часть работы, допустив существенные ошибки и / или нарушив технику безопасности. Опыт проводился в нерациональных условиях, что привело к получению результатов с большой погрешностью	В отчете допущены значительные недочеты (ошибки), измерения проведены с ошибками, вывод по работе отсутствует или неправилен	Работа выполнена и сдана со значительной задержкой (вне рамок занятия)
2	Ученик задание выполнил с соблюдением необходимой последовательности проведения опытов и измерений, но с небольшими недочетами	Ученик смог собрать установку для проведения опыта опираясь на инструкцию и/или при незначительной помощи учителя. Эксперимент проведен не полностью, во время работы допустил ошибки. Опыт проводился в условиях, не обеспечивающих достаточной точности измерения	В отчете допущены незначительные недочеты: не все измерения проведены правильно, не указаны единицы измерения величин, нет пояснений к рисункам, схемам, сделан вывод (с небольшими замечаниями)	Работа выполнена и оформлена, сдана с незначительной задержкой (немного не уложился во времени)

3	Ученик справился с заданием, выполнено полностью, с соблюдением необходимой последовательности и проведения опытов и измерений	Ученик самостоятельно собрал установку для проведения работы, самостоятельно подготовил и выбрал необходимое оборудование. Самостоятельно провел опыт в условиях режима обеспечивающих получение результатов и выводов с наибольшей точностью, соблюдая при этом технику безопасности	Работа выполнена самостоятельно, научно, логично описаны наблюдения, ход работы. Правильно, аккуратно выполнены все записи, таблицы, чертежи, вычисления, сделан правильный вывод, рассчитаны погрешности (при необходимости)	Своевременная сдача работы (уложился во времени)
---	--	---	---	--

8. СПИСОК ЛИТЕРАТУРЫ

Литература для обучающихся и родителей

1. Златопольский Д.М. «Основы программирования на языке Python, М.: ДМК-Пресс, 2017 г., 284 с.

Литература для педагогов

1. Бриггс Д. «Python для детей», М.: «Манн, Иванов и Фербер», 2017 г., 320 с.
2. Лутц М. Изучаем Python. 4-е издание, СПб, БХВ-Санкт-Петербург, 2014 г., 640 с.
3. Мак Грат М. «Python. Программирование для начинающих», М.: Эксмо, 2016 г., 194 с.
4. Прохоренок Н. - "Python 3 и PyQt. Разработка приложений", СПб, БХВ-Санкт-Петербург, 2014 г., 336 с.
5. Прохоренок Н. "Python. Самое необходимое", СПб, БХВ-Санкт-Петербург, 2015 г., 288 с.
6. Саммерфилд М. Программирование на Python 3, СПб, БХВ-Санкт-Петербург, 2015 г., 416 с.

Интернет-ресурсы

1. <https://www.codecademy.com/learn/python> - обучающий курс на английском языке.
2. <http://pythonworld.ru/samouchitel-python> - самоучитель Питон для начинающих и чайников
3. <http://www.pythonchallenge.com> – обучение программированию через игры и головоломки
4. <http://informatics.mccme.ru/course/view.php?id=156> – курс по языку Python Д.П. Кириенко
5. <http://pythontutor.ru/> - интерактивный учебник по программированию.

Приложение 1. Входной контроль в форме теста

1. Язык программирования, команды которого пишутся на человеческих языках, а затем интерпретируются в машинный код, называется:
 - а) высокоуровневым
 - б) иностранным
 - в) низкоуровневым
2. Строгая последовательность команд, ведущая к результату, называется:
 - а) инструкцией
 - б) алгоритмом
 - в) приказом
3. В программировании величина, которая может быть изменена в процессе исполнения программы, называется:
 - а) константой
 - б) переменной
 - в) значением
4. В программировании величина, которая не может быть изменена в процессе исполнения программы, называется:
 - а) константой
 - б) переменной
 - в) значением
5. Высказывание, которое полностью соответствует предмету, называется:
 - а) неполным
 - б) ложным
 - в) истинным
6. Алгоритмическая структура, которая подразумевает неоднократное повторение одних и тех же действий, называется:
 - а) массивом
 - б) циклом
 - в) условием
7. Что из перечисленного не является языком программирования:
 - а) python
 - б) pascal
 - в) Word
8. Какого числового типа данных не существует:
 - а) символьный
 - б) целочисленный
 - в) вещественный
9. Чем язык программирования отличается от обычных человеческих языков:
 - а) он имеет набор слов, составляющих язык
 - б) он очень формализован и не допускает изменения существующих слов
 - в) его понимают только компьютеры
10. Особый программный комплекс, включающий редактор кода, интерпретатор и дополнительные средства работы с программным кодом, называется:
 - а) средой программирования
 - б) текстовым редактором
 - в) операционной системой

Каждый ответ оценивается в 1 балл. Максимальная сумма баллов – 10

Перевод баллов:

0-4 – «неудовлетворительно»

5-6 – «удовлетворительно»

7-8 – «хорошо»

9-10 – «отлично»

1	2	3	4	5	6	7	8	9	10
а	б	б	а	в	б	в	а	б	а

Приложение 2. Примерные темы проектов для итоговой аттестации

1. Создание простого арифметического кнопочного калькулятора в python
2. Создание игры «Угадай число» в python
3. Создание игры «Стаканчики» в python
4. Создание приложений для выполнения математических задач (вычисление корней квадратного уравнения, геометрического калькулятора, процентного калькулятора и т. п.) в python
5. Создание игры «Слова» в python
6. Создание игры «Камень-ножницы-бумага» в python
7. Создание приложения-кассы в python
8. Создание игры-викторины в python

Приложение 3. Календарный учебный график на 2025-2025 учебный год

«УТВЕРЖДНО»
Директор МБОУ Тёпловская СОШ
Бейдарико В.Г.
Приказ от «20» мая 2025 года № 40



Календарный учебный график МБОУ Тёпловская СОШ на 2025-2025 учебный год

Начало учебного года – 1 сентября 2025г.

Завершение учебного года – 31 августа 2025г.

Режим работы: пятидневная рабочая неделя.

Режим занятий: по расписанию дополнительного образования, утвержденного директором школы.

Продолжительность занятий: 40 минут.

Период	Сроки	Продолжительность периода	Продолжительность каникул	В продолжительность учебного года не входят
1 четверть	01.09.2025 – 26.10.2024	8 недель + 1 день (41 день)		
каникулы	27.10.2025– 04.11.2025		9 дней	
2 четверть	05.11.2024 – 27.12.2024	8 недель + 4 дня (44 дней)		
каникулы	28.12.2024– 08.01.2025		12 дней	
3 четверть	09.01.2025 – 22.03.2025	10 недель + 3 дня (53 дней)		23 февраля 8 марта
каникулы	23.03.2025 – 30.03.2025		8 дней	
4 четверть	31.03.2025 – 26.05.2025	7 недель + 1 день (41 дня)		1-2 мая, 9-10 мая
Летние каникулы	05.06.2025– 31.08.2025	34 недели	88 дней	

Итоговая аттестация обучающихся проводится руководителем программы с привлечением администрации школы по окончанию обучения по дополнительной общеобразовательной общеразвивающей программе. Результаты итоговой аттестации фиксируются в «Протоколе» аттестации обучающихся, который является одним из отчетных документов и хранится у администрации учреждения.

Рассмотрено на педагогическом совете МБОУ Тёпловская СОШ
протокол №5 от «19» мая 2024 года