КОМИТЕТ ПО ОБРАЗОВАНИЮ САНКТ-ПЕТЕРБУРГА
АДМИНИСТРАЦИЯ ЦЕНТРАЛЬНОГО РАЙОНА САНКТ-ПЕТЕРБУРГА
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГИМНАЗИЯ № 155 ЦЕНТРАЛЬНОГО РАЙОНА САНКТ-ПЕТЕРБУРГА

ПРИНЯТО
Педагогическим советом
ГБОУ гимназии № 155
Центрального района Санкт-Петербурга
Протокол от «31» 08. 2026 г. № 24

УТВЕРЖДЕНО
Директор ГБОУ гимназии № 155
Центрального района Санкт-Петербурга
_____ О. Е. Шуйская
Приказ от «01 »
09 2026 г. № 156/1-к

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГИМНАЗИЯ № 155 ЦЕНТРАЛЬНОГО РАЙОНА САНКТ-ПЕТЕРБУРГА
Подписано цифровой подписью: ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГИМНАЗИЯ № 155 ЦЕНТРАЛЬНОГО РАЙОНА САНКТ-ПЕТЕРБУРГА
Дата: 2026.11.19 20:10:22 +03'00'

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
(ПЛАТНЫЕ ОБРАЗОВАТЕЛЬНЫЕ УСЛУГИ)

«ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON»

Возраст обучающихся: 12–16 лет (5–8 классы)
Срок реализации программы: 1 год (34 академических часа)
Режим занятий: 1 раз в неделю по 1 академическому часу
Уровень освоения: базовый
Педагог (автор-составитель): Безушко Кирилл Дмитриевич,
педагог дополнительного образования

Санкт-Петербург
2026

РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

1.1. Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на языке Python» имеет техническую направленность и разработана для реализации в рамках платных образовательных услуг ГБОУ гимназии № 155 Центрального района Санкт-Петербурга в 2026–2027 учебном году.

Нормативно-правовая база разработки программы:

- **Федеральный закон «Об образовании в РФ»:** от 29.12.2012 № 273-ФЗ (с изменениями и дополнениями).
- **Приказ Минпросвещения России:** от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
- **Концепция развития дополнительного образования детей:** распоряжение Правительства РФ от 31.03.2022 № 678-р (до 2030 года).
- **Федеральный государственный образовательный стандарт основного общего образования (ФГОС ООО):** в части формирования информационной грамотности, алгоритмической культуры и логического мышления.
- **Санитарно-эпидемиологические требования:** СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».
- **Локальные нормативные акты ГБОУ гимназии № 155:** Устав гимназии, Положение об организации деятельности по оказанию платных образовательных услуг, учебный план гимназии на 2026–2027 уч. год.

Актуальность программы:

Язык программирования Python на сегодняшний день является одним из наиболее востребованных, динамично развивающихся и универсальных языков в мировой IT-индустрии. Python активно используется в промышленной разработке веб-приложений, научных вычислениях, машинном обучении и технологиях искусственного интеллекта (Data Science & AI), автоматизации инженерных задач и системном администрировании. Благодаря чистому, лаконичному и интуитивно понятному синтаксису, Python признан идеальной стартовой платформой для перехода школьников от визуального блочного конструирования к промышленному текстовому коду. Освоение языка Python открывает перед подростками перспективы успешного участия в профильных олимпиадах (ВсОШ по информатике, Национальная технологическая олимпиада (НТО)), предпрофессиональных экзаменах и будущей осознанной профориентации в инженерно-технической сфере.

Педагогическая целесообразность:

Программа ориентирована на развитие системного алгоритмического мышления, математической культуры и навыков решения прикладных задач. В процессе написания программ подростки учатся формализовать текстовые условия, проектировать структуры данных, разбивать объемную задачу на функции (декомпозиция) и вырабатывать культуру чистого кода в соответствии со стандартом PEP 8. Деятельностный подход в обучении обеспечивает не механическое заучивание синтаксических конструкций, а их немедленное применение при создании практических консольных утилит, текстовых квестов, математических тренажеров и интерактивных приложений.

Отличительные особенности программы:

- **Практико-ориентированный интенсив:** более 70% учебного времени отводится непосредственной разработке кода за компьютером и разбору практических кейсов;
- **Современный стек инструментов:** обучение ведется на актуальной версии Python 3.12+ с использованием профессиональных сред разработки (IDLE, VS Code, Thonny);
- **Олимпиадный и проектный потенциал:** программа готовит фундамент для участия в олимпиадном программировании и создания индивидуальных исследовательских проектов.

Адресат программы:

Программа адресована обучающимся 12–16 лет (учащиеся 5–8 классов общеобразовательной школы). В подростковом возрасте происходит качественный переход к абстрактно-логическому и теоретическому мышлению. Подростки испытывают естественный интерес к взрослым профессиональным инструментам, стремятся создавать полезные утилиты, игры и автоматизированные решения, проявляют интерес к командной разработке и проектному сотворчеству.

Объем и сроки освоения программы:

Программа рассчитана на 1 учебный год (34 учебные недели в 2026–2027 учебном году). Общий объем программы составляет 34 академических часа (10 часов теории и 24 часа практических занятий).

Форма обучения и режим занятий:

Форма обучения — очная. Занятия проводятся в группах численностью 10–14 человек (с обеспечением каждого обучающегося персональным рабочим местом с ПК). Режим занятий: 1 раз в неделю по 1 академическому часу (45 минут). В структуру занятия обязательно включаются упражнения для профилактики зрительного утомления и разминка согласно нормам СанПиН.

1.2. Цель и задачи программы

Цель программы — формирование у обучающихся фундаментальных знаний и практических навыков текстового программирования на языке высокого уровня Python, развитие алгоритмической культуры, логического и системного мышления, а также компетенций самостоятельного проектирования и отладки программных продуктов.

Обучающие (предметные) задачи:

- **Синтаксис и базовые конструкции:** освоить фундаментальный синтаксис языка Python, правила оформления кода по стандарту PEP 8;
- **Типизация и переменные:** сформировать понимание строгой динамической типизации, базовых типов данных (int, float, str, bool), правил преобразования типов;
- **Управляющие конструкции:** научить уверенно применять условные конструкции (if, elif, else) и логические операторы (and, or, not);
- **Циклические алгоритмы:** освоить применение циклов while и for, генератора числовых последовательностей range(), операторов управления циклом break и continue;
- **Структуры данных:** сформировать устойчивые навыки работы с индексируемыми последовательностями — строками (str), списками (list) и кортежами (tuple);
- **Модульность и функции:** научить декомпозировать программы с помощью пользовательских функций (def, return) и подключать стандартные библиотеки (math, random, time);
- **Отладка и тестирование:** обучить поиску и устранению синтаксических и логических ошибок, основам перехвата исключений (try... except).

Развивающие (метапредметные) задачи:

- **Алгоритмическое мышление:** развивать способность структурировать сложные задачи, выделять подзадачи и выстраивать оптимальные логические цепочки;
- **Критическое мышление и анализ:** формировать привычку анализировать эффективность алгоритмов, оценивать чистоту и читаемость написанного кода;
- **Информационная грамотность:** развивать умение работать со справочной технической документацией и онлайн-ресурсами разработчиков;
- **Самоорганизация:** совершенствовать навыки планирования учебного времени, самопроверки и рефлексии.

Воспитательные (личностные) задачи:

- **Профессиональное самоопределение:** пробудить устойчивый интерес к инженерно-техническим и IT-профессиям;
- **Интеллектуальная культура:** воспитывать культуру научного поиска, аккуратность, терпение и настойчивость при отладке программ;
- **Этика разработчика:** формировать понимание ответственности программиста за качество создаваемого ПО и уважение к чужой интеллектуальной собственности.

1.3. Содержание программы

Учебный (тематический) план на 34 часа (1 час в неделю)

№ п/п	Наименование раздела / темы	Всего (ч)	Теория	Практика	Формы контроля
1	Раздел 1. Введение в Python и базовый синтаксис	4	1.5	2.5	Опрос, практическое задание
2	Раздел 2. Ветвление и логические выражения	5	1.5	3.5	Практикум: текстовый квест
3	Раздел 3. Циклические конструкции (while, for)	6	1.5	4.5	Промежуточный контроль (тест+код)
4	Раздел 4. Строки и методы обработки текста	4	1	3	Мини-проект: валидатор данных
5	Раздел 5. Списки (list) и кортежи (tuple)	5	1.5	3.5	Проект: консольный To-Do лист
6	Раздел 6. Функции и стандартные модули	5	1.5	3.5	Практическая работа: модуль утилит
7	Раздел 7. Итоговая проектная деятельность	5	1.5	3.5	Итоговая аттестация (защита)
	ИТОГО ПО ПРОГРАММЕ:	**34**	**10**	**24**	**Зачет / защита проектов**

Содержание разделов и тем учебного плана

Раздел 1. Введение в Python и базовый синтаксис (4 ч)

Тема 1.1. Техника безопасности. Среда разработки. Первая программа
Теория (0.5 ч): ТБ в кабинете информатики. История Python, философия языка. Интерпретатор и среды разработки (IDLE, VS Code). Структура программы.
Практика (0.5 ч): Запуск среды, написание программы «Hello, World!», сохранение файлов .py.

Тема 1.2. Функция print() и ее параметры

Теория (0.5 ч): Форматированный вывод текста и чисел. Именованные параметры sep (разделитель) и end (окончание строки). Управляющие символы (\n, \t).

Практика (0.5 ч): Решение задач на вывод текстовых узоров и структурированных чеков/визиток.

Тема 1.3. Переменные, базовые типы данных и правила PEP 8

Теория (0.5 ч): Понятие переменной как ссылки на объект в памяти. Типы данных: int (целые числа), float (вещественные), str (строки), bool (логический). Стандарт PEP 8.

Практика (0.5 ч): Объявление переменных, определение типов с помощью type(), арифметические вычисления.

Тема 1.4. Ввод данных input() и преобразование типов

Теория (0.5 ч): Функция input(), особенности строкового ввода. Явное приведение типов: int(), float(). Арифметические операторы (+, -, *, /, //, %, **).

Практика (0.5 ч): Разработка интерактивного консольного калькулятора (расчет площади, периметра, конвертер валют).

Раздел 2. Ветвление и логические выражения (5 ч)

Тема 2.1. Логический тип bool и операторы сравнения

Теория (0.5 ч): Значения True и False. Операторы сравнения: ==, !=, >, <, >=, <=. Таблицы истинности.

Практика (0.5 ч): Составление и вычисление логических выражений.

Тема 2.2. Условная конструкция if... else

Теория (0.5 ч): Синтаксис условия. Принцип отступов (4 пробела) в Python как основа блочной структуры.

Практика (0.5 ч): Программа проверки четности числа и проверки возраста для доступа к аттракционам.

Тема 2.3. Каскадное ветвление if... elif... else

Теория (0.5 ч): Множественный выбор. Порядок вычисления условий в цепочке elif.

Практика (0.5 ч): Программа перевода школьных баллов в словесные оценки, классификатор треугольников по сторонам.

Тема 2.4. Логические связки and, or, not. Приоритет операций

Теория (0.5 ч): Логическое умножение (and), сложение (or), отрицание (not). Скобки и приоритет.

Практика (0.5 ч): Проверка попадания точки в заданный координатный диапазон, проверка сложности пароля.

Тема 2.5. Практикум: Интерактивный текстовый квест

Теория (0.5 ч): Архитектура разветвленного нелинейного сюжета.

Практика (0.5 ч): Разработка текстовой ролевой мини-игры с выбором действий героя и различными финалами.

Раздел 3. Циклические конструкции (while, for) (6 ч)

Тема 3.1. Цикл с предусловием while

Теория (0.5 ч): Назначение цикла while. Условие продолжения цикла. Изменение переменной-счетчика внутри тела цикла.

Практика (0.5 ч): Вывод последовательностей чисел, таблиц умножения, накопление суммы.

Тема 3.2. Бесконечные циклы, операторы break и continue

Теория (0.5 ч): Опасность зацикливания. Принудительный выход из цикла (break) и пропуск итерации (continue).

Практика (0.5 ч): Организация постоянного меню программы с выходом по команде «exit».

Тема 3.3. Цикл for и генератор последовательностей range()
Теория (0.5 ч): Итерируемые объекты. Параметры функции range(start, stop, step). Прямой и обратный порядок.
Практика (0.5 ч): Подсчет суммы чисел в диапазоне, вычисление факториала, отбор четных чисел.

Тема 3.4. Вложенные циклы
Теория (0.5 ч): Внешний и внутренний цикл. Механика работы итераторов.
Практика (0.5 ч): Построение геометрических фигур из символов (прямоугольники, треугольники, ромбы), таблица Пифагора.

Тема 3.5. Модуль random и игра «Угадай число»
Теория (0.5 ч): Псевдослучайные числа. Функции randint(), choice().
Практика (0.5 ч): Разработка популярной игры «Угадай число» с подсказками «Больше/Меньше» и подсчетом попыток.

Тема 3.6. Промежуточный контроль: алгоритмический практикум
Теория (0.5 ч): Анализ типичных ошибок при работе с циклами и условиями.
Практика (0.5 ч): Контрольная практическая работа (решение 3 разноуровневых алгоритмических задач на время).

Раздел 4. Строки и методы обработки текста (4 ч)

Тема 4.1. Строки как неизменяемые последовательности. Индексация и срезы
Теория (0.5 ч): Положительная и отрицательная индексация символов. Синтаксис срезов: [start:stop:step]. Длина строки len().
Практика (0.5 ч): Извлечение подстрок, реверс строки, проверка на палиндром.

Тема 4.2. Базовые методы строк
Теория (0.5 ч): Изменение регистра (lower(), upper(), title()), поиск подстроки (find(), count()), замена (replace()).
Практика (0.5 ч): Написание антиспам-фильтра и цензора нежелательных слов в тексте.

Тема 4.3. Очистка и проверка строк: strip(), isdigit(), isalpha()
Теория (0.5 ч): Удаление пробельных символов, проверка состава строки (только цифры, только буквы).
Практика (0.5 ч): Создание модуля валидации номера телефона и адреса электронной почты.

Тема 4.4. Форматирование строк: f-строки (f-strings)
Теория (0.5 ч): Преимущества интерполяции строк через f-строки. Форматирование дробных чисел ({val:.2f}).
Практика (0.5 ч): Генератор красивых персонализированных отчетов и квитанций.

Раздел 5. Списки (list) и кортежи (tuple) (5 ч)

Тема 5.1. Понятие списка (массива). Создание и индексация
Теория (0.5 ч): Список как изменяемая упорядоченная коллекция объектов. Создание списков, изменение элементов по индексу.
Практика (0.5 ч): Заполнение списка оценками учащихся, расчет среднего балла.

Тема 5.2. Основные методы списков: добавление и удаление
Теория (0.5 ч): Методы append(), insert(), extend(), remove(), pop(), clear().
Практика (0.5 ч): Моделирование очереди клиентов и корзины интернет-магазина.

Тема 5.3. Перебор списков в цикле for. Функции min(), max(), sum()
Теория (0.5 ч): Итерация по элементам и по индексам (через range(len())). Поиск экстремумов и

суммирование.
Практика (0.5 ч): Решение задач на поиск максимального балла, подсчет положительных элементов.

Тема 5.4. Кортежи (tuple): неизменяемые последовательности
Теория (0.5 ч): Различия между list и tuple. Защита данных от модификации, экономия памяти. Распаковка кортежей.
Практика (0.5 ч): Хранение координат точек и фиксированных констант в кортежах.

Тема 5.5. Практикум: Консольный менеджер задач (To-Do List)
Теория (0.5 ч): Проектирование архитектуры консольного приложения со списком задач.
Практика (0.5 ч): Написание интерактивной программы с функциями добавления, просмотра, отметки и удаления задач.

Раздел 6. Функции и стандартные модули (5 ч)

Тема 6.1. Пользовательские функции: синтаксис def
Теория (0.5 ч): Понятие подпрограммы. Принцип DRY (Don't Repeat Yourself). Объявление функции и вызов.
Практика (0.5 ч): Написание функций вычисления математических формул и печати декоративных рамок.

Тема 6.2. Аргументы функции и возврат значения (return)
Теория (0.5 ч): Параметры и аргументы. Оператор return. Разница между печатью (print) и возвратом значения.
Практика (0.5 ч): Создание функций проверки простоты числа, нахождения корней квадратного уравнения.

Тема 6.3. Область видимости переменных: локальные и глобальные
Теория (0.5 ч): Локальное пространство имен функции и глобальная область. Опасность злоупотребления global.
Практика (0.5 ч): Практические упражнения на локализацию переменных и передачу данных через параметры.

Тема 6.4. Стандартные модули Python: math, time
Теория (0.5 ч): Инструкция import. Модуль math (sqrt, pi, sin, ceil, floor), модуль time (sleep, time).
Практика (0.5 ч): Написание секундомера, таймера с обратным отсчетом и математического калькулятора.

Тема 6.5. Практикум: Разработка библиотеки утилит и игр
Теория (0.5 ч): Структурирование проекта из нескольких функций.
Практика (0.5 ч): Сборка консольной игры «Камень, ножницы, бумага» или «Виселица» на основе функций.

Раздел 7. Итоговая проектная деятельность (5 ч)

Тема 7.1. Выбор темы и архитектурное проектирование проекта
Теория (0.5 ч): Жизненный цикл ПО. Техническое задание (ТЗ), функциональные требования, декомпозиция на модули.
Практика (0.5 ч): Составление паспорта проекта, проектирование структур данных и прототипа меню.

Тема 7.2. Разработка программного ядра проекта
Теория (0.5 ч): Реализация ключевой бизнес-логики и математических моделей проекта.
Практика (0.5 ч): Написание основного кода функций и главного цикла программы.

Тема 7.3. Разработка интерфейса и обработка исключений (try... except)
Теория (0.5 ч): Понятие исключений (ValueError, ZeroDivisionError). Конструкция try... except для надежности.
Практика (0.5 ч): Защита программы от некорректного пользовательского ввода.

Тема 7.4. Комплексное тестирование и оформление документации
Теория (0.5 ч): Тест-кейсы (штатные, граничные, ошибочные). Оформление комментариев (docstrings) и README.
Практика (0.5 ч): Взаимное тестирование программ одноклассниками, исправление выявленных дефектов.

Тема 7.5. Итоговая аттестация: публичная защита проектов
Теория (0.5 ч): Регламент выступления, правила презентации кода и демонстрации работы ПО.
Практика (0.5 ч): Публичная защита авторских программных проектов обучающимися, подведение итогов курса.

1.4. Планируемые результаты освоения программы

В соответствии с ФГОС основного общего образования и целями дополнительного образования, освоение программы обеспечивает достижение личностных, метапредметных и предметных результатов.

Личностные результаты:

- **Профориентационная готовность:** осознание возможностей применения программирования в современных профессиях, формирование устойчивой мотивации к инженерно-технической деятельности;
- **Критичность мышления:** способность критически оценивать корректность и оптимальность полученных программных результатов;
- **Настойчивость и целеустремленность:** умение преодолевать трудности в процессе поиска ошибок и багов, стремление довести решение задачи до идеального состояния;
- **Информационная этика:** понимание норм сетевого взаимодействия, соблюдение этических правил использования чужого кода (open-source лицензии).

Метапредметные результаты:

- **Регулятивные УУД:** умение формулировать цель деятельности, составлять план действий, прогнозировать результаты работы программы, осуществлять самоконтроль и рефлексию;
- **Познавательные УУД:** владение методами формализации и моделирования реальных процессов, способность переводить условие задачи на язык формальных алгоритмов, анализировать и преобразовывать информацию;
- **Коммуникативные УУД:** способность аргументированно отстаивать свою точку зрения при выборе архитектуры программы, презентовать программные решения, продуктивно взаимодействовать в проектной команде.

Предметные результаты:

- **Синтаксис и конструкции Python:** знание правил синтаксиса Python 3, стандарта PEP 8, типов данных (int, float, str, bool), операторов ветвления (if-elif-else) и циклов (while, for);

- **Работа со структурами данных:** умение применять индексацию, срезы и встроенные методы при обработке строк, списков и кортежей;
- **Модульное программирование:** умение создавать собственные функции с параметрами и возвращаемыми значениями, импортировать и использовать стандартные модули (math, random, time);
- **Отладка и обработка исключений:** навыки выявления и исправления синтаксических и логических ошибок, защита программ от сбоев с помощью try... except;
- **Проектная реализация:** опыт полного цикла разработки прикладного консольного приложения: от постановки задачи до презентации работающего продукта.

РАЗДЕЛ 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ РЕАЛИЗАЦИИ ПРОГРАММЫ

2.1. Календарный учебный график

Календарный учебный график регламентирует организацию образовательного процесса по программе на 2026–2027 учебный год в ГБОУ гимназии № 155 Центрального района Санкт-Петербурга.

Параметр	Значение / Сроки
Период реализации программы	01 сентября 2026 г. – 26 мая 2027 г.
Продолжительность учебного года	34 учебные недели
Общее количество часов	34 академических часа (1 час в неделю)
Режим занятий	1 занятие в неделю по 45 минут
Осенние каникулы	25 октября 2026 г. – 03 ноября 2026 г. (по графику КО СПб)
Зимние каникулы	31 декабря 2026 г. – 10 января 2027 г.
Весенние каникулы	27 марта 2027 г. – 04 апреля 2027 г.
Промежуточная аттестация	Декабрь 2026 г. (15–16 занятие, алгоритмический практикум)
Итоговая аттестация	Май 2027 г. (34 занятие, публичная защита проектов)

2.2. Условия реализации программы

Материально-техническое обеспечение:

- **Учебный компьютерный класс:** кабинет информатики, оснащенный регулируемой эргономичной мебелью, соответствующий санитарным нормам СП 2.4.3648-20;
- **Рабочие места учащихся:** персональные компьютеры (процессор от Intel Core i3 / AMD Ryzen 3, ОЗУ от 8 ГБ, SSD-накопитель, монитор от 21.5" с Full HD разрешением, клавиатура, мышь) — 15 шт.;
- **Рабочее место преподавателя:** ПК преподавателя, мультимедийный проектор с проекционным экраном / интерактивная панель от 65" для пошаговой трансляции написания кода;
- **Сетевая инфраструктура:** локальная сеть, высокоскоростной интернет (от 100 Мбит/с) с контент-фильтрацией.

Информационно-программное обеспечение:

- **Операционная система:** Windows 10/11 Pro или отечественные дистрибутивы Linux (Astra Linux / Alt Linux);
- **Среда разработки и интерпретатор:** дистрибутив Python 3.12+, встроенная среда IDLE, а также специализированные среды для обучения программированию (Thonny, VS Code с расширением Python);
- **Сопутствующее ПО:** веб-браузеры для доступа к документации docs.python.org, текстовые редакторы.

Кадровые условия:

Программу ведет педагог дополнительного образования Безушко Кирилл Дмитриевич. Квалификация педагога полностью соответствует профессиональному стандарту «Педагог дополнительного образования детей и взрослых», подтверждена профильным образованием в сфере информационных технологий и регулярным повышением квалификации по современным языкам программирования и методике преподавания IT-дисциплин.

2.3. Формы аттестации и оценочные материалы

Контроль освоения программы осуществляется в форме безотметочной рейтинговой и качественной оценки, направленной на поддержку мотивации и выявление зон роста каждого обучающегося.

Виды и этапы контроля:

- **Входной контроль (1–2 урок):** опрос и вводные практические задания на оценку базовой компьютерной грамотности и логического мышления;
- **Текущий контроль (на каждом уроке):** проверка правильности выполнения практических задач, разбор кода (code review), тестирование программ на контрольных данных;
- **Промежуточный контроль (декабрь, 15–16 урок):** практическая зачетная работа по разделам «Базовый синтаксис», «Ветвления» и «Циклы»;

- **Итоговый контроль (май, 34 урок):** публичная защита авторского выпускного проекта (консольной утилиты, игры или информационной системы).

Критерии оценивания программных проектов на Python:

Критерий оценки	Высокий уровень (3 балла)	Средний уровень (2 балла)	Низкий уровень (1 балл)
Работоспособность и алгоритмы	Программа стабильно работает, решает поставленную задачу, использует ветвления, циклы, структуры данных и функции	Программа работает в штатном режиме, но возможны сбои при нетипичном вводе, логика упрощена	Программа содержит синтаксические или критические ошибки во время выполнения
Качество кода (PEP 8)	Соблюдены отступы, осмысленные имена переменных и функций, код снабжен комментариями (docstrings)	Имеются небольшие отклонения от PEP 8 (неинформативные имена переменных), мало комментариев	Хаотичное форматирование, грубые нарушения отступов, отсутствие комментариев
Обработка исключений	Реализована надежная защита от ошибок ввода через try... except, валидация данных	Присутствует частичная проверка условий ввода через if, нет обработки исключений	Защита от некорректного ввода отсутствует, программа падает с ошибкой
Самостоятельность и защита	Автор свободно ориентируется в собственном коде, подробно отвечает на технические вопросы, уверенно выступает	Ориентируется в коде с наводящими вопросами педагога, отвечает на базовые вопросы	Затрудняется объяснить работу ключевых функций, не может модифицировать код на ходу

2.4. Методические материалы

В основу методики положен метод проблемного обучения и парного/индивидуального программирования. Каждая новая тема начинается с демонстрации практической проблемы или интересного кейса, после чего совместно с учащимися вырабатывается алгоритмическое решение.

Структура типового 45-минутного занятия:

- **1. Организационный этап (2–3 мин):** подготовка ПК, открытие среды разработки;
- **2. Разминка / Повторение (5 мин):** решение короткой разминочной алгоритмической задачи (код-блиц);
- **3. Теоретический блок (10 мин):** разбор новой языковой конструкции, синтаксиса и типичных ошибок на проекторе;

- **4. Оздоровительная пауза (2 мин):** гимнастика для глаз и разминка плечевого пояса;
- **5. Практическая работа за ПК (20 мин):** написание, отладка и тестирование программного кода;
- **6. Code Review и подведение итогов (5 мин):** коллективный разбор удачных решений, рефлексия, фиксация результатов.

2.5. Список литературы и информационных ресурсов

Нормативные правовые акты:

- **1.** Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации».
- **2.** Приказ Министерства просвещения РФ от 27.07.2022 № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
- **3.** СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

Литература для педагога:

- **1.** Поляков К. Ю. Программирование. Python. В 4 частях. — М.: Лаборатория знаний, 2023. — 256 с.
- **2.** Лутц М. Изучаем Python. Том 1, 2. — СПб.: Диалектика, 2022. — 832 с.
- **3.** МакГрат М. Программирование на Python для начинающих. — М.: Эксмо, 2022. — 192 с.

Литература для обучающихся:

- **1.** Бриггс Дж. Python для детей. Самоучитель по программированию. — М.: Манн, Иванов и Фербер, 2022. — 320 с.
- **2.** Пэйн Б. Легкий Python для детей и родителей. — СПб.: Питер, 2021. — 400 с.

Интернет-ресурсы и образовательные платформы:

- **1.** Официальная документация языка Python: https://docs.python.org/3/
- **2.** Образовательная платформа «Питонтьютор» (интерактивный онлайн-учебник): https://pythontutor.ru
- **3.** Платформа Stepik (курсы «Поколение Python»): https://stepik.org

ПРИЛОЖЕНИЕ 1. КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ НА 2026–2027 УЧЕБНЫЙ ГОД

по дополнительной общеобразовательной общеразвивающей программе «Основы программирования на языке Python» (34 часа, 1 час в неделю)

№	Тема занятия	Часы	Теория	Практ.	Форма контроля / Продукт деятельности
1	Вводный инструктаж по ТБ. Знакомство со средой разработки Python (IDLE / VS Code)	1	0.5	0.5	Входная диагностика, тест ТБ
2	Функция print(): форматированный вывод, параметры sep и end, escape-последовательности	1	0.5	0.5	Практическое задание на вывод
3	Переменные, правила именования (PEP 8). Базовые типы данных: int, float, str, bool	1	0.5	0.5	Упражнение: типизация
4	Ввод данных с клавиатуры input(). Явное преобразование типов. Арифметика в Python	1	0.5	0.5	Консольный мини-калькулятор
5	Логический тип bool. Операторы сравнения. Таблицы истинности	1	0.5	0.5	Практическая работа
6	Условный оператор if... else. Правило синтаксических отступов (indents)	1	0.5	0.5	Проверка условий, код-чекер
7	Каскадные условия: конструкция if... elif... else	1	0.5	0.5	Программа-классификатор
8	Сложные логические выражения: операторы and, or, not	1	0.5	0.5	Упражнение на логику
9	Практикум: Разработка интерактивного разветвленного текстового квеста	1	0	1.0	Текстовый мини-квест
10	Понятие цикла. Цикл с предусловием while. Условие выхода и переменные-счетчики	1	0.5	0.5	Циклические алгоритмы
11	Управление циклом: операторы принудительного выхода break и continue	1	0.5	0.5	Интерактивное консольное меню
12	Цикл со счетчиком for и встроенный генератор range(start, stop, step)	1	0.5	0.5	Вычисление сумм и факториала
13	Вложенные циклы for. Построение геометрических таблиц и паттернов	1	0.5	0.5	Генератор таблиц символов
14	Модуль random: псевдослучайные числа. Разработка игры «Угадай число»	1	0.5	0.5	Игра «Угадай число»

№	Тема занятия	Часы	Теория	Практ.	Форма контроля / Продукт деятельности
15	Промежуточный контроль: алгоритмический практикум по ветвлениям и циклам	1	0	1.0	Зачетная работа (3 задачи)
16	Строки (str) как неизменяемые последовательности. Индексация символов и срезы [start:stop:step]	1	0.5	0.5	Скрипт проверки палиндрома
17	Базовые методы строк: lower(), upper(), find(), replace(), count()	1	0.5	0.5	Программа фильтрации текста
18	Очистка и проверка данных: strip(), isdigit(), isalpha(). Форматирование f-strings	1	0.5	0.5	Генератор чеков и отчетов
19	Практикум: Разработка консольного валидатора контактных данных (телефон, email)	1	0	1.0	Модуль валидации данных
20	Списки (list): динамические массивы. Создание, индексация, изменение элементов	1	0.5	0.5	Журнал оценок учащихся
21	Методы работы со списками: append(), insert(), remove(), pop(), sort(), reverse()	1	0.5	0.5	Симулятор корзины товаров
22	Итерация по спискам в цикле for. Функции min(), max(), sum(), len()	1	0.5	0.5	Анализ числового массива
23	Кортежи (tuple): неизменяемые коллекции данных. Распаковка кортежей	1	0.5	0.5	База координат объектов
24	Практикум: Консольное приложение «Менеджер личных задач» (To-Do List)	1	0	1.0	Консольный To-Do лист
25	Понятие функции в программировании. Синтаксис объявления def и вызов	1	0.5	0.5	Библиотека простых функций
26	Параметры и аргументы функции. Оператор возврата значения return	1	0.5	0.5	Математические функции
27	Область видимости переменных: локальные и глобальные переменные	1	0.5	0.5	Практикум на скоупы переменных
28	Стандартные модули Python: модуль математики math и системного времени time	1	0.5	0.5	Консольный секундомер/таймер
29	Практикум: Создание многофункциональной игры «Поле чудес» / «Виселица» с функциями	1	0	1.0	Игра со словами на функциях
30	Введение в обработку исключений: конструкция try... except для надежности кода	1	0.5	0.5	Защищенный калькулятор

№	Тема занятия	Часы	Теория	Практ.	Форма контроля / Продукт деятельности
31	Итоговый проект: выбор темы, архитектура программы и декомпозиция задач	1	0.5	0.5	Техническое задание и каркас
32	Практическая разработка ядра и логики итогового программного проекта	1	0	1.0	Рабочий прототип проекта
33	Доработка интерфейса, комплексная отладка, обработка ошибок и code review	1	0	1.0	Финальный релиз программы
34	Итоговая аттестация: публичная презентация и защита авторских проектов на Python	1	0	1.0	Итоговый зачет (защита)

ПРИЛОЖЕНИЕ 2. ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ИТОГОВЫХ ТВОРЧЕСКИХ ПРОЕКТОВ

- **Интерактивная консольная игра «Текстовая RPG: Подземелья гимназии» —** ролевая игра с выбором класса героя, характеристиками (HP, атака, инвентарь), пошаговой боевой системой и сохранением прогресса;
- **Автоматизированная система учета личных финансов «SmartBudget» —** консольное приложение для ввода доходов/расходов по категориям, расчета баланса, вывода статистики и предупреждения о перерасходе;
- **Обучающий тренажер для подготовки к ОГЭ по математике/информатике —** генератор случайных заданий по заданным шаблонам, проверка ответов, подсчет процента успешности и разбор ошибок;
- **Консольная версия классической игры «Морской бой» или «Крестики-нолики» —** игра для двух игроков или против простого искусственного интеллекта (компьютера) с двумерным игровым полем;
- **Информационно-справочная система «Школьный электронный органайзер» —** расписание уроков, учет домашних заданий с дедлайнами, фильтрация по дням недели и поиск заметок;
- **Криптографическая утилита «Шифратор и дешифратор текстов» —** реализация шифра Цезаря, шифра Виженера и алгоритмов хеширования с интерфейсом для кодирования сообщений.