**Министерство науки и высшего образования Российской Федерации**

**Муниципальное бюджетное учреждение дополнительного образования «Центр детского научного и инженерно-технического творчества» города Невинномысска**

СОГЛАСОВАНО
Педагогическим советом
Протокол № 1
от «27» августа 2026 г.

УТВЕРЖДАЮ
Директор
_____Т.В. Чилхачоян
«27» августа 2026 г.

# Дополнительная общеобразовательная программа технической направленности

# РАБОЧАЯ ПРОГРАММА по курсу «Информатика и основы алгоритмизации»

7 класс
Срок реализации программы – 1 год

**ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

**СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП**

**Сертификат:** 008830DD740B303171238478D0E3864D90
**Владелец:** ЧИЛХАЧОЯН ТАТЬЯНА ВАЛЕРЬЕВНА
**Действителен:** с 04.05.2026 по 28.07.2027
**Время подписи:** 28.08.2026 в 14:47:40
**Организация:** МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ "ЦЕНТР ДЕТСКОГО НАУЧНОГО И ИНЖЕНЕРНО-ТЕХНИЧЕСКОГО ТВОРЧЕСТВА" ГОРОДА НЕВИННОМЫССКА
**Должность:** Директор

Подписано с помощью цифрового сервиса «На-Подписи»

Авторы-составители:
Бенескул А.В., педагог
Фоменко О.Н., педагог
Завялик О.П., педагог

Невинномысск, 2026

# СОДЕРЖАНИЕ

1. Пояснительная записка
2. Учебно-тематический план и содержание
3. Организационно-педагогические условия реализации программы
4. Список литературы

# 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа технической направленности «Информатика и основы алгоритмизации» предназначена для обучающихся 7 класса и реализуется в рамках дополнительного образования. Актуальность программы обусловлена необходимостью системного выравнивания знаний учащихся, поступающих по разным траекториям 5–6 классов (1 час и 2 часа в неделю), и создания единого стартового уровня для дальнейшего углублённого изучения Python. Программа разработана с опорой на модульно-концентрический принцип, где фундаментальные алгоритмические конструкции и базовый синтаксис интегрированы с практико-ориентированными задачами, что обеспечивает плавный переход от визуального программирования к текстовому коду и формирование навыков самостоятельной разработки консольных и модульных приложений.

**Актуальность программы** обусловлена требованиями современного общества к формированию системы работы с заинтересованными инженерным и научно-техническим творчеством детьми в условиях дополнительного образования.

Программа разработана на основе следующих документов:

– закон Российской Федерации «Об образовании» (Федеральный закон от 29 декабря 2012 г. № 273-ФЗ);

– приказ Министерства просвещения РФ от 27 июля 2022 г. №629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

– концепция развития дополнительного образования детей до 2030 года (Распоряжение Правительства РФ от 31 марта 2022 г. №678-р);

– постановление Главного государственного санитарного врача от 28.09.2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодёжи»;

– постановление Главного государственного санитарного врача от 28.01.2021 г. № 2 «Об утверждении санитарных правил СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

**Педагогическая целесообразность программы** обуславливается стимулированием интересов учащихся к дисциплинам технического направления, экспериментальным исследованиям, проектной деятельности, в обеспечении адаптации школьников к жизни в обществе, профессиональной ориентации, а также выявлении и поддержке учащихся, проявивших выдающиеся способности.

Программа может быть реализована с помощью дистанционных технологий, технологий смешанного и модульного обучения.

**Цель программы** – формирование у обучающихся единого базового уровня алгоритмического мышления и практических компетенций в написании структурированного кода на Python, достаточных для перехода к объектно-

ориентированному проектированию и разработке прикладных программных продуктов.

**Задачи программы:**

− Консолидация основ синтаксиса, типов данных и алгоритмических конструкций Python.

− Сформировать устойчивые навыки работы со стандартными и составными структурами данных, файловой системой и механизмами обработки ошибок;

− Ввести базовые принципы объектно-ориентированного программирования (ООП) и модульной архитектуры приложений;

− Развить алгоритмическую культуру через решение практических задач, отладку кода и версионирование в Git;

− Закрепить теоретические знания через разработку, тестирование и публичную защиту промежуточного и итогового проектов.

**Объем, содержание и планируемые результаты программы** определены, исходя из особенностей интересов и потребностей учащихся в области технических наук. Вместе с тем при определении объема и содержания программы учитывалась сложность конкретной темы по отношению к другим темам раздела, возможность приобретения учащимися практического опыта и осуществления межпредметных связей. Порядок тем и блоков, представленный в учебно-тематическом плане, допускается корректировать в пределах программы класса.

**К отличительным особенностям программы** концентрический подход построения программы и ее пропедевтическая направленность. Программа обеспечивает готовность к применению информатики в других дисциплинах и является основой для успешного усвоения других дисциплин технической и естественнонаучной направленности.

**Срок реализации** программы – 1 год.

**Продолжительность учебного года** – 34 недели.

**Общий объем** – 68 часов.

**Формы и режим занятий:**

Занятия проводятся по 2 часа в неделю в постоянных группах учащихся 7 классов (оптимальное количество участников в группе: 8-15 человек), в форме теоретических, практических и индивидуальных занятий, а также консультаций по выполнению проектов.

**Ожидаемые результаты освоения программы курса:**

1. Уверенное написание читаемого и структурированного кода на Python с использованием функций, циклов, условий и базовых структур данных;
2. Способность проектировать и реализовывать обработку текстовых, табличных (CSV/JSON) и файловых данных с учётом валидации и обработки исключений;
3. Понимание и применение базовых принципов ООП (классы, объекты, инкапсуляция, наследование) для моделирования предметных областей;

4. Совершенствование умения формализации и структурирования информации;

5. Приобретение умения выбирать способ представления данных в соответствии с поставленной задачей с использованием соответствующих программных средств обработки данных;

6. Владение инструментами версионного контроля (Git), виртуальными окружениями и базовыми приёмами модульного тестирования;

7. Готовность самостоятельно планировать архитектуру небольшого приложения, реализовывать мини-проект в середине года и итоговый проект в конце года с последующей защитой результатов.

Результаты освоения программы определяются с использованием пятибалльной шкалы.

**Контроль освоения программы** – текущий, промежуточный и итоговый.

***Текущий контроль*** осуществляется на занятиях (ответы у доски, письменные работы, практические работы и устные ответы, домашние задания); после изучения блока или набора взаимосвязанных блоков (выделенных педагогом) защита практической работы или устный опрос.

***Промежуточный контроль*** проводится в соответствии с учебно-тематическим планом после освоения темы или набора взаимосвязанных тем в интерактивной форме, содержащей устную и практическую часть или защиты практической работы.

***Итоговый контроль*** – осуществляется в форме итогового проекта. Ученик может выбрать 3 направления для разработки итогового проекта, общественно полезные проекты, проекты для внутреннего использования, и проекты, направленные на коммерческие компании.

Программой не предусмотрено использование тестов для итогового контроля.

## 2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН И СОДЕРЖАНИЕ

**[60 часов, 2 часа в неделю]**

| № | Наименование тем и блоков | Общее количество учебных часов | Теор. | Практ. |
|---|---|---|---|---|
| **Тема 1** | **Python - основы** | **18** | **6** | **12** |
| Блок 1 | Переменные, типы данных, операторы ввода/вывода, основы Git | 6 | 2 | 4 |
| Блок 2 | Условные конструкции, логические выражения, циклы (for, while) | 6 | 2 | 4 |
| Блок 3 | Функции, области видимости, возврат значений, документирование кода | 6 | 2 | 4 |
| **Тема 2** | **Структуры данных и работа с файлами** | **14** | **6** | **8** |
| Блок 1 | Списки, кортежи, множества | 6 | 2 | 4 |
| Блок 2 | Словари, чтение/запись файлов (txt, csv, json), обработка исключений | 8 | 4 | 4 |
| | **Мини проект** | **2** | | |
| **Тема 3** | **Введение в ООП и архитектура приложений** | **24** | **4** | **12** |
| Блок 1 | Классы, объекты, атрибуты, методы | 10 | 2 | 8 |
| Блок 2 | Инкапсуляция, наследование, полиморфизм | 6 | 2 | 4 |
| Блок 3 | Модульность, виртуальные окружения, структура проекта, CLI/Flask основы | 8 | 4 | 4 |
| | **Итоговый проект** | **2** | | |
| | **Резерв** | **8** | | |
| | Всего | 68 | | |

## СОДЕРЖАНИЕ ПРОГРАММЫ

**Тема 1. Python - основы**

**Блок 1. Переменные, типы данных, операторы ввода/вывода, основы Git.**

Повторение и углубление работы с базовыми типами данных, арифметическими операциями, вводом/выводом. Первичная настройка репозиториев в Git, создание первого коммита и работа с ветками.

**Блок 2. Условные конструкции, логические выражения, циклы (for, while).**

Закрепление навыков ветвления, построения сложных логических условий и оптимизации итерационных алгоритмов. Сравнение производительности циклических конструкций на реальных задачах.

**Блок 3. Функции, области видимости, возврат значений, документирование.**

Декларация пользовательских функций, управление глобальными/локальными переменными, передача аргументов. Оформление технической документации (docstrings) к подпрограммам.

**Практическая работа №1-2:** Настройка рабочего окружения, отработка операций с базовыми типами данных, арифметикой и консольным вводом/выводом. Инициализация Git-репозитория, создание структуры папок проекта, выполнение базовых операций версионирования.

**Практическая работа №3-5:** Реализация алгоритмов ветвления с использованием сложных логических выражений. Написание и оптимизация итерационных алгоритмов с помощью циклов for и while, отладка вложенных конструкций.

**Практическая работа №6-7:** Декларация и вызов пользовательских функций, отработка работы с областями видимости и возвратом значений. Оформление документации к модулю кода, финальная отладка и публикация в удалённый репозиторий.

**Тема 2. Структуры данных и работа с файлами**

**Блок 1. Списки, кортежи, множества, списковые включения.**

Применение изменяемых и неизменяемых коллекций, фильтрация элементов и освоение эффективного синтаксиса списковых включений для компактной генерации данных.

**Блок 2. Словари, чтение/запись файлов (txt, csv, json), обработка исключений.**

Хранение структурированных данных в словарях, парсинг внешних файловых источников. Реализация безопасной обработки потенциальных ошибок выполнения с помощью try/except/finally.

**Практическая работа №1–2:** Работа со списками, кортежами и множествами. Реализация задач на фильтрацию, объединение и пересечение коллекций, отработка синтаксиса списковых включений.

**Практическая работа №3–4:** Разработка консольного приложения для анализа, очистки и визуализации небольшого набора данных.

***Защита мини проекта с презентацией***

**Тема 3. Введение в ООП и архитектура приложений**

**Блок 1. Классы, объекты, атрибуты, методы.**

Создание классов, инициализация объектов, определение атрибутов и методов для моделирования реальных сущностей. Понимание разницы между классом и экземпляром.

**Блок 2. Инкапсуляция, наследование, полиморфизм.**

Организация доступа к данным, построение иерархий наследования, переопределение поведения объектов. Использование магических методов для кастомизации вывода и операторов.

**Блок 3. Модульность, виртуальные окружения, структура проекта, CLI/Flask основы.**

Разбиение кода на логические модули, настройка изолированных окружений (venv). Подготовка структуры проекта для запуска консольных утилит или базовых веб-скриптов.

**Практическая работа №1-2:** Проектирование классов, инициализация объектов, реализация конструкторов и методов. Моделирование предметных областей через создание экземпляров классов.

**Практическая работа №3-4:** Реализация инкапсуляции, построение иерархий наследования, переопределение методов. Отработка полиморфизма на примере обработки разнородных объектов в едином цикле.

**Практическая работа №5-6:** Модульная архитектура проекта, настройка venv и requirements.txt. Разбиение кода на пакеты, подготовка точки входа (__main__.py), отладка CLI-скрипта.

**Практическая работа №7:** Проектирование архитектуры приложения на основе ООП, реализация основного функционала итогового проекта.

***Защита итогового проекта с презентацией***

## 3. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Занятия проводятся в постоянных группах учащихся (оптимальное количество участников в группе: 8-15 человек), в форме теоретических, практических и индивидуальных занятий и консультаций (проектная деятельность, подготовка к олимпиадам, конференциям).

### 3.1 ТРЕБОВАНИЯ К ТЕХНИЧЕСКОМУ И ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

**Аппаратное обеспечение:**

**Рабочее место:** Персональный компьютер или ноутбук с процессором уровня Intel Core i3 / AMD Ryzen 3 (или выше), 4–8 ГБ ОЗУ, SSD от 128 ГБ, монитор Full HD.

**Интернет:** Стабильное подключение от 10 Мбит/с для загрузки библиотек, работы с Git/GitHub и доступа к облачным редакторам при необходимости.

**Программное обеспечение:**

**ОС:** Windows 10/11, macOS 12+ или Linux (Ubuntu 22.04+).

**Среда разработки:** VS Code (с расширениями Python, GitLens) или PyCharm Community Edition.

**Язык и базовые модули:** Python 3.10+, стандартная библиотека (os, sys, json, csv, pathlib), менеджер пакетов pip.

**Системы контроля версий:** Git, клиент GitHub Desktop или командная строка.

**Дополнительно:** Браузер Chrome/Firefox, пакет офисных приложений для оформления проектной документации, облачное хранилище для обмена кодом.

## 4. СПИСОК ЛИТЕРАТУРЫ

1. Ахмедханлы, Д. М. Основы алгоритмизации и программирования [Текст] : учеб.метод. пособие / Д.М. Ахмедханлы, Н.В. Ушмаева. – Тольятти: Издв. ТГУ, 2016. –123 с.

2. Златопольский, Д. М. Основы программирования на языке Python : учебное пособие для школьников / Д. М. Златопольский. — Москва : Лаборатория знаний, 2021. — 224 с. — (Программирование для школьников). — ISBN 978-5-93208-527-4.

3. Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы : учебное пособие / Д. М. Златопольский. — 4-е изд., электрон. — Москва : Лаборатория знаний, 2020. — 320 с. — (Программирование). — ISBN 978-5-97060-827-2.

4. Гусева, А. И. Учимся программировать на Python : практическое руководство для начинающих / А. И. Гусева. — Санкт-Петербург : БХВ-Петербург, 2022. — 304 с. — (Программирование для школьников). — ISBN 978-5-9775-6821-3.

5. Воронцов, К. В. Введение в машинное обучение : учебный курс / К. В. Воронцов. — Москва : МФТИ, 2023. — Электронный ресурс. — Режим доступа: https://www.machinelearning.ru/wiki/index.php?title=Введение_в_машинное_обучение (дата обращения: 13.05.2026).

6. «Поколение Python»: курс для начинающих : официальный онлайн-курс / автор Т. Гуев. — Платформа Stepik, 2021–2026. — Режим доступа: https://stepik.org/course/67 (дата обращения: 13.05.2026).

7. «Поколение Python»: курс для продвинутых : официальный онлайн-курс / автор Т. Гуев. — Платформа Stepik, 2022–2026. — Режим доступа: https://stepik.org/course/68343 (дата обращения: 13.05.2026).