



Урок 31. Постановка проекта обработки данных

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 31 из 34 · 45 минут · Модуль «Итоговый марафон»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

Данные о длительности учебных запусков ещё не являются отчётом. До написания программы договоримся о единицах, пустом файле, ошибках и точном формате результата.

Результат урока: Составить проверяемый контракт файла и отчёта, алгоритм и набор приёмочных примеров для общего проекта.

01. Проблемная ситуация

0–5 минут. Файл содержит 12 25 20 8 25. Один автор считает только значения >20 , другой ≥ 20 . Чьи ответы 2 и 3 верны? Запишите, какого уточнения не хватает в задании.

02. Понятия и разобранный пример

5–14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Контракт до кода

Проект четырёх завершающих уроков: отчёт о длительностях учебных запусков, в минутах. Эти числа вымышлены. Файл `data.txt` в UTF-8 содержит от 0 до 100 неотрицательных целых $0 \dots 1000$, разделённых пробельными символами. Допустимы только ASCII-цифры без знака и точки. Пустой файл допустим. Порог $q=20$ минут, требуется количество значений не меньше порога.

`result.txt`: первая строка содержит n , сумму минут, количество значений ≥ 20 через пробел; вторая — все исходные значения, упорядоченные по неубыванию, с сохранением повторов. Обе строки заканчиваются переводом строки. Для пустоты первая строка `0 0 0`, вторая пустая. Файлы именуются точно, повторный успешный запуск заменяет предыдущий `result.txt`.

На неверном токене, числе вне диапазона, более 100 значениях или ошибке чтения программа сообщает `ERROR` и не записывает новый отчёт. Старый отчёт не означает успех текущего запуска: перед тестом используем отдельную папку. Алгоритм: чтение → разбор и проверка → вычисление → запись. На этом занятии составляем спецификацию и проверяем ядро; полный файловый конвейер собирается на следующем.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Для 12 25 20 8 25: $n=5$, сумма 90 минут, ≥ 20 — три значения. Первая строка отчёта `5 90 3`, вторая `8 12 20 25 25`. Два значения 25 сохраняются. Ядро ниже получает уже проверенный список; оно не читает файл и не проверяет токены.

```
def report(a, q):
    b = sorted(a)
    count = sum(x >= q for x in a)
    return len(a), sum(a), count, b

print(report([12, 25, 20, 8, 25], 20))
```



Вход	data.txt: 12 25 20 8 25
Контроль	0...100 значений, каждое 0...1000
Расчёт	n=5; сумма=90 мин; >=20: 3
Выход	5 90 3 / 8 12 20 25 25

Контракт общего проекта: единицы и смысл каждой части отчёта.

03. Практика 1. Проверяем материал

Составляем спецификацию

14–25 минут. В паре составьте лист спецификации, не меняя заданный контракт.

Файл data.txt	Работа проектировщика
12 25 20 8 25	Выписать обе строки
Пустой файл	Определить две строки
0 20 20	Проверить ноль и повторы
12 2.5	Определить ошибку

2 минуты — перечень входов и единиц; 4 — четыре ожидаемых результата; 3 — алгоритм из четырёх этапов; 2 — взаимная проверка. Форма: вход / статус / первая строка / вторая строка. Критерии: порог включён, повторы сохранены, пустота описана, ошибочный файл не превращён в частичный отчёт.

04. Практика 2. Перенос способа

Разбираем спорный файл

25–33 минуты. Получен файл со строками «020 5» и «1001». За 2 минуты решите судьбу 020 и 1001, за 3 составьте исправленный допустимый пример с тем же числом токенов, за 3 запишите ожидаемый отчёт. Отдельно решите судьбу -0, +5 и 5.0.

05. Обсуждение и границы вывода

33–40 минут. Предлагается пропускать плохие токены ради удобства. За 2 минуты назовите выгоду, за 3 покажите риск на 12 2.5, за 2 предложите явную альтернативу, не меняя контракт незаметно.

06. Индивидуальная проверка

40–45 минут. Для файла 20 0 3 укажите две строки отчёта, единицы суммы и одну причину отказа. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Обычный: 5 90 3 и 8 12 20 25 25. Пустой: 0 0 0 и пустая строка. 0 20 20: 3 40 2 и 0 20 20. 12 2.5: ERROR, нового отчёта нет. Чтение только корректного префикса 12 нарушило бы контракт целого файла.

Практика 2: разбор и критерии

020 допустим и означает 20; 1001 вне диапазона, весь файл отклоняется. Например, заменить 1001 на 1000: первая строка 3 1025 2, вторая 5 20 1000. -0,+5,5.0 не состоят только из ASCII-цифр и отклоняются. Критерии: не путать формат токена и числовой диапазон, не терять данные молча.

Обсуждение: возможный вывод

Пропуск облегчает получение хоть какого-то результата, но скрывает потерю данных и меняет n и сумму. В этом проекте отклоняем файл целиком. В другом проекте возможен отдельный режим пропуска с журналом отвергнутых строк, но он требует нового контракта и тестов.

Индивидуальная проверка: ответ и критерии

3 23 1; 0 3 20; сумма в минутах; например токен 2.5 или значение 1001. По баллу за каждый пункт.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск `python lesson.py` (либо `python3 lesson.py`). Внешние библиотеки не нужны. Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 9 класс. Предпосылки: текстовые файлы, списки, функции, сортировка и поиск. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание.

Матрица программы предназначена учителю.

Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ



07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Повторение и обобщение: файлы, функции, сортировка, поиск	Составляем спецификацию	Составить проверяемый контракт файла и отчёта, алгоритм и набор приёмочных примеров для общего проекта.
Самостоятельная проверка результата	Разбираем спорный файл	Обоснованный ответ и исправление ошибки

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

Python: чтение и запись текстовых файлов

<https://docs.python.org/3/tutorial/inputoutput.html>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.