



Урок 33. Защита мини-проекта

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 33 из 33 · 45 минут · Модуль «Марафон по строкам и массивам»

Уровень: начинающий программист; Python 3 и C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Программа уверенно печатает `sum=6` для `[1,5]`, но проверяющий ожидал `sum=5`. Как защитить решение по контракту и найти ошибку одним коротким тестом?

Результат урока: Защитить мини-проект: сформулировать контракт, предъявить тесты, обнаружить ошибку на границе и подтвердить исправление в двух языках.

01. Проблемная ситуация

0–5 минут. Правило: выбрать $x \geq 2$, сохранить порядок и выдать выбранные элементы, их сумму и максимум. Для `[1,5]`, `[2]` и `[]` вручную составьте ожидаемые результаты. Почему одного удачного теста мало?

02. Понятия и разобранный пример

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

Защита начинается с проверяемого обещания

Проект `filter_report(a)` обрабатывает не более 100 целых от -1000 до 1000. Выбирает значения $x \geq 2$, сохраняет порядок и повторы; возвращает выбранный набор, сумму и максимум. При отсутствии выбранных сумма 0, максимума нет.

Вычисление суммы должно относиться к выбранному набору. Если прибавлять все исходные значения, тест только из подходящих чисел не выявит ошибку. Если заменить ≥ 2 на > 2 , различающий тест — `[2]`. Минимальный контрпример делает причину ошибки видимой.

В защите показывают исходный контракт, таблицу ожидаемых и фактических ответов, изменение конкретной строки и повторный прогон всех тестов. Исправление одного теста не отменяет проверки пустоты, повторений и отрицательных входов. Реальные тесты запускают; бумажный разбор честно называют трассировкой.

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 `[1,5,2,5,-1]` выбранный набор `[5,2,5]`, сумма 12, максимум 5. Если суммировать весь вход, получится 12 тоже: 1 и -1 случайно взаимно уничтожатся. Поэтому добавляем `[1,5]`, где неправильная сумма 6 отличается от 5.

```
def filter_report(a):
    selected = [x for x in a if x >= 2]
    total = sum(selected)
    best = max(selected) if selected else None
    return selected, total, best

cases = [[1,5,2,5,-1],[1,5],[2],[],[-3,0],[3,3]]
for a in cases:
    print(a, filter_report(a))
```

Контракт	$x \geq 2$; порядок и повторы сохранены
Слепое пятно	<code>[1,5,2,5,-1]</code> : обе суммы равны 12
Различающий тест	<code>[1,5]</code> : выбранная сумма 5, общая 6
Повторный прогон	граница <code>[2]</code> , пустота <code>[]</code> , повторы <code>[3,3]</code>

Удачный ответ может скрывать ошибку: тест должен различать верную и неверную версии.



03. Практика 1. Проверяем материал

Минимальный тест отличает версии

14–25 минут. Проведите защиту в парах: автор показывает контракт и шесть тестов, проверяющий выбирает строку для изменения.

Вход	Выбранные; сумма; максимум
[1,5,2,5,-1]	[5,2,5]; 12; 5
[1,5]; [2]	[5]; 5; 5 / [2]; 2; 2
[]; [-3,0]	[]; 0; None для обоих
[3,3]	[3,3]; 6; 3

2 минуты — ожидаемые ответы; 4 — запуски; 3 — замените `sum(selected)` на `sum(a)`, найдите контрпример и исправьте; 2 — повторно прогоните все случаи. Продукт: таблица вход/ожидание/факт до/факт после. Критерии: контрпример, причина, исправление, полный повторный прогон.

04. Практика 2. Перенос способа

Перенос и повторный прогон

25–33 минуты. На C++ выдайте тот же смысл в формате: значения через пробел, затем `|sum=S|max=M`, для отсутствия максимума `none`. За 2 минуты прогноз для [2,1,2], за 4 выполните перенос, за 2 проверьте [], [2], [1,5] и [3,3]. Критерии: граница, сумма только выбранных, отсутствие выдуманного максимума.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты оцените заявление «все мои тесты прошли — ошибок нет», за 3 предложите неудобный вход и ограничение проекта, за 2 сформулируйте честный итог защиты.

06. Индивидуальная проверка

40–45 минут. Для [1,2,2,-5] запишите результат, тест против замены `>=` на `>`, тест против `sum(a)` и одну фразу честного отчёта о проверке. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Основной тест не замечает неправильную сумму из-за $1+(-1)=0$. [1,5] даёт ошибочно 6, правильно 5; [-3,0] ошибочно -3, правильно 0. Исправление — суммировать selected. Удалять неудобные тесты нельзя. Шесть окончательных ответов заданы таблицей.

Практика 2: разбор и критерии

[2,1,2] даёт 2 2 |sum=4|max=2; [] даёт |sum=0|max=none; [2] — 2 |sum=2|max=2; [1,5] — 5 |sum=5|max=5; [3,3] — 3 3 |sum=6|max=3. Пробел после каждого выбранного числа предусмотрен форматом.

```
#include <iostream>
#include <vector>
void show(const std::vector<int>& a) {
    std::vector<int> b;
    long long sum=0;
    for (int x : a) {
        if (x>=2) { b.push_back(x); sum+=x; }
    }
    for (int x : b) { std::cout << x << " "; }
    std::cout << "|sum=" << sum << "|max=";
    if (b.empty()) { std::cout << "none"; }
    else {
        int best=b[0];
        for (int x : b) { if (x>best) { best=x; } }
        std::cout << best;
    }
    std::cout << "\n";
}
int main() {
    for (auto a : std::vector<std::vector<int>>
        {{2,1,2},{},{2},{1,5},{3,3}}) { show(a); }
}
```

Обсуждение: возможный вывод

Тесты подтверждают поведение на проверенных входах, но не доказывают отсутствие всех ошибок. Качественный вывод: контракт и ограничения названы, проверены шесть категорий случаев, исправление показано на различающем тесте, повторный прогон успешен. Файлы и произвольный текст пока не поддерживаются.

Индивидуальная проверка: ответ и критерии

[(2,2),4,2); [2] различает границу; [1,5] различает общую и выбранную суммы; пример фразы: «На заданных тестах выход соответствует контракту; произвольные входы вне диапазона не проверялись». По баллу за пункт. Этим занятием завершается 33-часовая траектория 8 класса.

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. Для второй практики также подготовьте G++: `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`; запуск `./lesson`. 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, 8 класс. Предпосылки: фильтрация, поиск максимума, текстовый отчёт, тестирование. 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
Марафон по строкам и массивам	Защитить мини-проект: сформулировать контракт, предъявить тесты, обнаружить ошибку на границе и подтвердить исправление в двух языках.	Исходник, журнал тестов и объяснение результата
Повторение и обобщение; строки, условия и коллекции	Перенос и повторный прогон	Самостоятельная доработка, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Python: структуры данных

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

C++: обработка vector

<https://eel.is/c++draft/vector.overview>

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.