



Урок 34. Защита и исправление проекта

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

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

Уровень: начинающий программист; Python 3.

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

На защите важно не только показать удачный экран. Другой человек должен повторить запуск, распознать ошибку и увидеть, что исправление не сломало прежний результат.

Результат урока: Предъявить воспроизводимый отчёт и график, обнаружить дефект масштаба и повторно проверить проект после исправления.

01. Проблемная ситуация

0–5 минут. В отчёте значения 12,25,20,8,25 минут. На графике самый большой столбец подписан 12. Какие два элемента нужно сопоставить, прежде чем принимать визуализацию?

02. Понятия и разобранный пример

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

Отчёт, график и воспроизводимость

Финальная стадия общего проекта не добавляет второй марафон для другого языка. Защищаем уже проверенные вычисления и отдельную визуализацию. Ниже полный построитель SVG: никаких пакетов и детских аккаунтов, файл chart.svg открывает браузер. Вход графика — 1...10 проверенных целых длительностей 0...1000 минут. Для 0 или более 10 значений график по этому контракту не строится, но текстовый отчёт общего проекта остаётся допустим.

График сохраняет исходный порядок, номера запусков на горизонтали начинаются с 1. Высота пропорциональна длительности: $h = 180 \cdot x / \text{top}$, где $\text{top} = \max(1, \max(a))$. Базовая линия $Y = 250$, верх столбца $Y = 250 - h$: экранная Y растёт вниз. Над каждым столбцом — точное значение в минутах. Нулевой столбец имеет высоту 0, а его подпись остаётся.

На защите предъявите исходный набор, версию кода, команды и ожидаемый результат. Для общего data.txt «12 25 20 8 25» текстовый отчёт должен содержать 5 90 3 и 8 12 20 25 25. График ниже использует тот же набор напрямую; автоматическое чтение и отчёт были в уроке 32. В этом занятии проверяется отдельный экспорт изображения, а не выдаётся за новый файловый конвейер.

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 $x = 25$, $\text{top} = 25$ высота 180, верх $Y = 70$. Для $x = 12$ высота 86.4, верх $Y = 163.6$. Столбцы 2 и 5 равны. Если ошибочно использовать $Y = h$, больший столбец начнётся ниже и нарушит общую базовую линию.

```
from pathlib import Path
```

```
def save_chart(a, filename):
    if not a or any(not 0 <= x <= 1000 for x in a):
        raise ValueError("nonempty values 0..1000 required")
    if len(a) > 10:
        raise ValueError("at most 10 bars")
    parts = ['<svg xmlns="http://www.w3.org/2000/svg" '
             'width="520" height="300" viewBox="0 0 520 300">',
             '<rect width="520" height="300" fill="white"/>']
    top = max(1, max(a))
    parts.append('<path d="M40 30 V250 H500" '
                'fill="none" stroke="black"/>')
    for i, x in enumerate(a):
        h = 180 * x / top
        left = 50 + i * 440 / len(a)
        width = 300 / len(a)
        parts.append(f'<rect x="{left}" y="{250-h}" '
                    f'width="{width}" height="{h}" fill="blue"/>')
        parts.append(f'<text x="{left}" y="{240-h}">{x}</text>')
        parts.append(f'<text x="{left}" y="275">{i+1}</text>')
    parts.append('<text x="45" y="20">Duration, min</text></svg>')
    Path(filename).write_text("".join(parts), encoding="utf-8")

save_chart([12, 25, 20, 8, 25], "chart.svg")
```



Запуск	1 2 3 4 5
Минуты	12 25 20 8 25
Высота, px	86.4 180 144 57.6 180
Верх Y, px	163.6 70 106 192.4 70

Проверочные координаты графика: все столбцы заканчиваются на Y=250.

03. Практика 1. Проверяем материал

Готовим мини-защиту

14–25 минут. Учитель выдаёт готовый файл построителя. Каждый тест сохраняйте под отдельным именем в тестовой папке.

Материал защиты	Что должен повторить проверяющий
Данные и отчёт	5 90 3; 8 12 20 25 25
chart.svg	Пять столбцов в исходном порядке
[0,0]	Две нулевые подписи, без деления на ноль
[]	ValueError, график не создаётся

2 минуты — расчёт высот; 3 — запуск основного и нулевого набора; 2 — проверка отказа для []; 2 — двухминутная защита партнёру; 2 — протокол замечаний. Критерии: верные подписи/порядок, общая база, нулевой случай, воспроизводимая команда `python lesson.py` и открытие `chart.svg`.

04. Практика 2. Перенос способа

Исправляем нулевой масштаб

25–33 минуты. В копии кода замените `top=max(1,max(a))` на `top=max(a)`. За 2 минуты назовите сбойный вход, за 3 воспроизведите ошибку, за 3 восстановите исправление и повторите обычный и нулевой тесты. Запишите до/после и объяснение, почему `max(1,...)` не искажает положительные целые данные.

05. Обсуждение и границы вывода

33–40 минут. На защите предлагают скрыть сообщение об ошибке и оставить старый красивый график. За 2 минуты назовите последствие, за 3 предложите прозрачную демонстрацию отказа, за 2 сформулируйте правило приёмки.

06. Индивидуальная проверка

40–45 минут. Для [10,20] вычислите высоты и верхние Y двух столбцов; укажите одно доказательство воспроизводимости и один тест ошибки. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Основные высоты приведены на схеме. Для $[0,0]$ $\text{top}=1$, обе высоты 0, подписи 0 видимы на $Y=240$. Для $[]$ возникает `ValueError` до записи. Не проверяйте ошибочный тест по старому `chart.svg`: используйте новое имя или чистую папку. На защите называются данные, команда, результат и границы контракта.

Практика 2: разбор и критерии

Для $[0,0]$ $\text{top}=0$ и вычисление $180 \cdot x / \text{top}$ вызывает `ZeroDivisionError`. Исправление `max(1, max(a))` даёт $\text{top}=1$, высоты 0. При положительном целом максимуме он не меньше 1 и остаётся неизменным. Критерии: минимальный контрпример, причина, исправление и повтор обоих тестов.

Обсуждение: возможный вывод

Старое изображение может относиться к другому набору. Приёмка требует связи входа, версии кода и нового результата. Отказ показывают в отдельной папке и объясняют контракт. Отсутствие исключения недостаточно: проверяют числа и геометрию. Исправление принимают после повторных обычных и граничных тестов.

Индивидуальная проверка: ответ и критерии

Высоты 90 и 180 px; $Y=160$ и 70. Доказательство: исходный набор, версия исходника и команда дают заново тот же SVG. Тест ошибки: $[]$ либо 11 значений вызывает `ValueError`. По баллу за высоты, Y , воспроизводимость, ошибку.

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. SVG открывайте обычным браузером. Для каждого теста используйте отдельную папку или новое имя изображения. 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: контракт, тесты проекта, координаты SVG и открытие файла браузером. 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	Готовим мини-защиту	Предъявить воспроизводимый отчёт и график, обнаружить дефект масштаба и повторно проверить проект после исправления.
Самостоятельная проверка результата	Исправляем нулевой масштаб	Обоснованный ответ и исправление ошибки

Источники и права на изображения

Исходная программа курса

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

W3C: прямоугольники и геометрия SVG

<https://www.w3.org/TR/SVG2/shapes.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.