



Урок 33. Сравнение Python и C++

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 33 из 34 · 45 минут · Модуль «Итоговый марафон»

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

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

Две реализации могут дать один и тот же отчёт и разное измеренное время. Сначала проверим равенство результатов, затем решим, что именно было измерено.

Результат урока: Сопоставить реализации по одному контракту и ограничить вывод о скорости условиями измерения.

01. Проблемная ситуация

0–5 минут. Получены времена 0,20 мс и 20 мс. В первом случае измеряли только функцию, во втором — запуск программы вместе с чтением файла. Можно ли заключить, что первый язык быстрее в 100 раз? Назовите отсутствующее условие сравнения.

02. Понятия и разобранный пример

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

Одинаковый результат и границы замера

Сравниваем вычислительное ядро, не разбор токенов: вход — уже проверенный список 0...100 целых 0...1000, порог 20 минут. Нужны n , сумма, количество ≥ 20 и неубывающий порядок. Python использует `sorted` и прямой подсчёт; C++ сортирует `vector` и использует `lower_bound`. Это разные реализации одного результата.

Коды ниже — два отдельных полностью запускаемых файла. Python печатает те же две строки без лишнего пробела в конце; C++ оставляет конечный пробел во второй строке. Для этой проверки сравниваем числовые токены строк, а не байты файлов. Если требуются побайтно одинаковые файлы, формат вывода надо унифицировать отдельно.

Измерять надо одинаковую работу: одинаковый вход, включение копирования, сортировки, поиска и вывода, одинаковое число повторов. Отдельно фиксируют версии, настройки компиляции и устройство. Медиана повторов устойчивее единичного запуска к случайным задержкам, но не превращает один опыт в универсальное свойство языка. В обоих примерах секундомер охватывает создание входного контейнера, сортировку, сумму и поиск/подсчёт; печать вынесена за замер. `kernel_ms` выводится отдельно в поток ошибок. Для пяти чисел время слишком мало для устойчивого ранжирования; сохраните показания как демонстрацию границ замера, не как доказательство скорости языка.

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 обе программы дают 5 90 3 и значения 8 12 20 25 25. После сортировки C++ `lower_bound(20)` указывает на позицию 2; $n-2=3$. Для пустого вектора `begin()==end()`, разность равна 0.

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <chrono>
using namespace std;
int main() {
    auto start=chrono::steady_clock::now();
    vector<int> a={12,25,20,8,25};
    sort(a.begin(),a.end());
    long long total=0;
    for (int x:a) total+=x;
    auto first=lower_bound(a.begin(),a.end(),20);
    auto stop=chrono::steady_clock::now();
    double ms=chrono::duration<double,milli>(stop-start).count();
    cerr << "kernel_ms=" << ms << "\n";
    cout << a.size() << ' ' << total << ' '
         << a.end()-first << '\n';
    for (int x:a) cout << x << ' ';
    cout << '\n';
}
```



Python	sorted + прямой подсчёт
C++	sort + lower_bound
Общее	n=5; сумма=90; количество=3
Сравнение	Сначала результат, затем одинаковый замер

Результат сравнивают по контракту, а скорость — по одинаковой измеряемой работе.

03. Практика 1. Проверяем материал

Сопоставляем реализации

14–25 минут. Python-код ниже сохраните отдельно; в обоих файлах меняйте только исходный набор.

Набор	Ожидаемые n, сумма, >=20
12,25,20,8,25	5,90,3
пустой	0,0,0
20,20,0	3,40,2
19,21	2,40,1

2 минуты — прогноз; 5 — четыре пары запусков; 2 — сравнение токенов двух строк; 2 — журнал. Критерии: одинаковые числовые результаты, сохранённые повторы, пустой случай, объяснение различия реализации. В журнал внесите kernel_ms и версии Python/G++, пометив единичные замеры как демонстрационные.

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

from time import perf_counter
import sys
start=perf_counter()
n,total,count,b=report([12,25,20,8,25],20)
ms=(perf_counter()-start)*1000
print("kernel_ms=",ms,file=sys.stderr)
print(n,total,count)
print(*b)
```

04. Практика 2. Перенос способа

Проверяем сравнимость времени

25–33 минуты. Даны учебные, не реально измеренные здесь времена одного и того же ядра в мс: P=[2.1,2.0,2.2,2.0,8.0], C=[0.8,0.9,0.8,0.9,3.0]. Копирование включено, печать и запуск процесса исключены. За 2 минуты упорядочьте серии, за 3 найдите медианы и их отношение, за 3 напишите вывод с двумя ограничениями. Отдельно оцените сравнение P с C, если в C включили компиляцию.

05. Обсуждение и границы вывода

33–40 минут. Предлагаются решения: переписать проект целиком ради языка либо сначала найти медленный участок. За 2 минуты назовите цену переписывания, за 3 предложите воспроизводимый замер, за 2 запишите решение.

06. Индивидуальная проверка

40–45 минут. Времена [4,1,2,2,9] мс: найдите медиану, назовите два условия честного сравнения и один вывод, который эти числа не разрешают. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Первая строка указана в таблице; вторые: 8 12 20 25 25; пустая; 0 20 20; 19 21. Пробел в конце строки C++ не меняет токены, но меняет байты. У C++ исходный vector изменён sort; Python report создаёт b и сохраняет исходный a.

Практика 2: разбор и критерии

Р отсортировано: 2.0,2.0,2.1,2.2,8.0; медиана 2.1 мс. С: 0.8,0.8,0.9,0.9,3.0; медиана 0.9 мс. Отношение 2.1/0.9≈2.33. В этой учебной серии С меньше по медиане; не доказано для других входов и устройств. Включение компиляции только в С делает границы замера несопоставимыми. Критерии: обе медианы, отношение, оговорки.

Обсуждение: возможный вывод

Сначала проверяют правильность и измеряют чтение, расчёт, запись отдельно. Медленный диск или вывод могут доминировать над поиском. Оптимизируют подтверждённое узкое место; выбор языка зависит также от сопровождения и библиотек. Учебные числа не являются реальным сравнительным тестом Python и C++.

Индивидуальная проверка: ответ и критерии

Медиана 2 мс; одинаковый вход; одинаковые границы измерения (допустимы также устройство/повторы/настройки); нельзя заключить, что язык всегда быстрее другого. 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. Компилятор G++ с поддержкой C++17 и редактор. Сборка: `g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson`. Запуск `./lesson` в Linux/macOS или `.lesson.exe` в Windows (при сборке задайте `-o lesson.exe`). Также установите Python 3.11+; второй исходник запускается `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: проект уроков 31–32, стандартная сортировка, измерение времени. 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	Сопоставляем реализации	Сопоставить реализации по одному контракту и ограничить вывод о скорости условиями измерения.
Самостоятельная проверка результата	Проверяем сравнимость времени	Обоснованный ответ и исправление ошибки

Источники и права на изображения

Исходная программа курса

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

Python: счётчик времени `perf_counter`

https://docs.python.org/3/library/time.html#time.perf_counter

C++: стандартный поиск границы

<https://eel.is/c++draft/alg.binary.search>

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.