



Урок 29. Частоты символов

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 29 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>

У текстов "abc" и "cba" одинаковые частоты букв, но разный порядок. Как составить точный отчёт, который даст одинаковые числа в Python и C++?

Результат урока: Построить частотный отчёт по общему контракту, сравнить два языка и выбрать лидера с явным правилом ничьей.

01. Проблемная ситуация

0–5 минут. В строке "cab ab!" вручную сосчитайте a,b,c и остальные символы, включая пробел. Проверьте, что сумма четырёх счётчиков равна длине строки.

02. Понятия и разобранный пример

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

Одинаковый контракт важнее одинаковой записи

Алфавит отчёта — ровно "abc", регистр учитывается. Счётчики a,b,c хранятся в списке из трёх нулей; всё остальное попадает в other. Входы урока — ASCII длиной до 100, поэтому Python и C++ обрабатывают одинаковые единицы текста.

Python `alphabet.find(ch)` возвращает 0,1,2 или -1. В C++ `find` возвращает позицию либо `pros`; нельзя механически переносить проверку -1 в условие без понимания типа. Каждый входной символ увеличивает ровно один счётчик, поэтому сумма частот плюс other равна длине.

Частоты сохраняют количество, но теряют порядок. Для выбора наиболее частой буквы нужно правило ничьей: среди a,b,c выбираем первую по порядку алфавита. Если ни одной из этих букв нет, лидер обозначается знаком -. Ноль наблюдений не должен случайно объявлять а победителем.

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

"cab ab!" содержит семь символов: c,a,b,пробел,a,b,!. Частоты [2,2,1], other=2, сумма 2+2+1+2=7. a и b делят максимум 2; по принятому правилу лидер a.

```
def frequencies(text):
    alphabet = "abc"
    counts = [0, 0, 0]
    other = 0
    for ch in text:
        i = alphabet.find(ch)
        if i == -1:
            other += 1
        else:
            counts[i] += 1
    return counts, other

for text in ["cab ab!", "", "aaa", "ABC", "abc", "cba"]:
    print(repr(text), frequencies(text))
```



Текст	c a b пробел a b !
Категория	c a b other a b other
Счётчики	a: 2; b: 2; c: 1; other: 2
Проверка	2 + 2 + 1 + 2 = 7

Каждый символ ASCII входит ровно в одну категорию; пробел и восклицательный знак учитываются как other.

03. Практика 1. Проверяем материал

Контроль суммы счётчиков

14–25 минут. Проверьте шесть строк Python. Для каждой допишите длину и сумму счётчиков. До запуска обведите все символы, которые попадают в other.

Текст	[a,b,c]; other
"cab ab!"	[2,2,1]; 2
""; "aaa"	[0,0,0]; 0 и [3,0,0]; 0
"ABC"	[0,0,0]; 3
"abc"; "cba"	[1,1,1]; 0 для обоих

2 минуты — подсчёт, 4 — запуск, 3 — ошибочно замените if i== -1 на if i и сравните результаты для "a", "!" и "b", 2 — восстановите программу. Критерии: частоты, регистр, other, контроль общей суммы.

04. Практика 2. Перенос способа

Второй язык и правило ничьей

25–33 минуты. Запустите второй полный исходник на C++ для "aabcc!", "bbbcca", "CAB", "" и сравните первые четыре числа с Python frequencies на тех же входах. Последний знак — лидер; при ничьей выбирайте более раннюю букву, при нулевых частотах — -. За 2 минуты составьте прогноз, за 4 выполните сравнение, за 2 объясните условие строгого > при выборе лидера. Критерии: совпадение счётчиков, other, ничья, пустой результат.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сопоставьте abc и cba. За 3 объясните, почему контроль общей суммы не доказывает правильность распределения по буквам. За 2 решите, какие дополнительные проверки нужны для переноса на кириллицу.

06. Индивидуальная проверка

40–45 минут. Для "ccba?" запишите [a,b,c], other, лидера и проверку суммы. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Для `a` индекс 0 ложен, поэтому ошибочная ветка совпадёт с `else` и даст верный результат лишь случайно; для `b` и `c` ненулевые индексы ошибочно отправят буквы в `other`. Для `!` индекс -1 истинен и попадёт в `other`. Поэтому добавьте `"b"` как различающий тест: условие должно сравнивать `i` именно с -1. Сумма может сохраняться даже у ошибочного распределения.

Практика 2: разбор и критерии

Ключи C++: `"2 1 2 1 a"; "1 3 2 0 b"; "0 0 0 3 -"; "0 0 0 0 -"`. Строгое `>` сохраняет первого лидера при равенстве; `>=` выбрало бы последнюю букву с тем же максимумом. Для первого теста Python возвращает `([2,1,2],1)`.

```
#include <iostream>
#include <string>
#include <array>
int main() {
    std::string text; std::getline(std::cin, text);
    const std::string alphabet = "abc";
    std::array<int,3> counts{0,0,0};
    int other = 0;
    for (char ch : text) {
        auto i = alphabet.find(ch);
        if (i == std::string::npos) { ++other; }
        else { ++counts[i]; }
    }
    int best = 0; char winner = '-';
    for (std::size_t i = 0; i < counts.size(); ++i) {
        if (counts[i] > best) {
            best = counts[i]; winner = alphabet[i];
        }
        std::cout << counts[i] << " ";
    }
    std::cout << other << " " << winner << "\n";
}
```

Обсуждение: возможный вывод

Одинаковые частоты не означают одинаковый текст. Общая сумма проверяет полноту распределения, но не правильные категории: ошибка может просто переложить букву в `other`. Для кириллицы надо согласовать кодировку, единицу подсчёта, регистр и алфавит; побайтовый `char`-перебор UTF-8 не равен перебору букв.

Индивидуальная проверка: ответ и критерии

`[1,1,2]; other=1`; лидер `c`; `1+1+2+1=5`. Каждый символ учтён один раз, и частота с строго больше остальных.

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: строки ASCII, списки и `vector`, циклы, условия. 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	Построить частотный отчёт по общему контракту, сравнить два языка и выбрать лидера с явным правилом ничьей.	Исходник, трассировка и таблица проверок
Повторение и обобщение: строки и массивы	Второй язык и правило ничьей	Самостоятельный алгоритм, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

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

Python: поиск позиции символа

<https://docs.python.org/3/library/stdtypes.html#str.find>

C++: поиск и npos

<https://eel.is/c++draft/string.find>

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.