



Урок 31. Фильтрация данных

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 31 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>

Из [5,-1,5,0,8,3] нужно оставить значения от 0 до 5. Можно ли удалять элементы прямо во время перебора и почему две пятёрки должны остаться?

Результат урока: Создать новый набор по условию, сохранив порядок и повторения, и отличить фильтрацию от изменения исходника.

01. Проблемная ситуация

0–5 минут. Вычеркните неподходящие значения [5,-1,5,0,8,3] на бумаге. Запишите оставшиеся в исходном порядке и их индексы. Обсудите, является ли сортировка частью задачи.

02. Понятия и разобранный пример

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

Результат строится отдельно

Фильтр оставляет элементы, удовлетворяющие предикату. Устойчивый порядок означает: если два выбранных элемента шли в исходнике слева направо, они идут так же и в результате. Повторы являются отдельными наблюдениями и не удаляются.

Для замкнутого диапазона $lo \dots hi$ проверяем $lo \leq x \leq hi$. Контракт задаёт $lo \leq hi$ и не более 100 целых от -1000 до 1000. Начинаем с пустого результата, читаем исходный список, добавляем подходящее через `append`. Пустой исходник и отсутствие подходящих дают [].

Удаление из обходящегося списка сдвигает оставшиеся индексы и легко пропускает соседний элемент. Новый список понятнее: он хранит выбранную подпоследовательность уже просмотренного префикса. В C++ создаём отдельный `vector` и используем `&&` между двумя сравнениями; цепочка $lo \leq x \leq hi$ не означает диапазон.

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

При $lo=0, hi=5$ последовательность результата: [5], [5], [5,5], [5,5,0], [5,5,0], [5,5,0,3]. Исходник остаётся [5,-1,5,0,8,3]; число элементов результата 4, исходника 6.

```
def select_range(a, lo, hi):
    result = []
    for x in a:
        if lo <= x <= hi:
            result.append(x)
    return result
```

```
a = [5, -1, 5, 0, 8, 3]
b = select_range(a, 0, 5)
print(b)
print(a)
```

Вход	5 -1 5 0 8 3
$0 \leq x \leq 5$	да нет да да нет да
Индексы	0 2 3 5
Выход	5 5 0 3

Фильтрация сохраняет порядок и повторения; границы 0 и 5 включены.



03. Практика 1. Проверяем материал

Две границы и устойчивый порядок

14–25 минут. Проверьте пять входов. После вызова отдельно напечатайте исходный список и убедитесь, что он не изменён.

Вход; lo,hi	Результат
[5,-1,5,0,8,3]; 0,5	[5,5,0,3]
[]; 0,5 / [-2,-1]; 0,5	[] / []
[0,5,6]; 0,5	[0,5]
[3,3,2]; 3,3	[3,3]

2 минуты — прогноз; 4 — запуски; 3 — замените $\leq$ на $<$ с обеих сторон и найдите потери; 2 — восстановите код. Критерии: обе границы, порядок, повторы, исходник.

04. Практика 2. Перенос способа

Перенос условия в C++

25–33 минуты. На C++ отберите чётные числа замкнутого диапазона. За 2 минуты получите ответ для [6,2,5,2,-2,0], lo=-2,hi=5; за 4 напишите функцию; за 2 проверьте пустой вход и диапазон 2...2. Критерии: &&, чётность, границы, порядок и повторы.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сравните фильтр с удалением дублей, за 3 разберите, почему [2,2] отличается от [2], за 2 предложите проверку исходника. Можно ли считать правильным ответ с теми же числами в другом порядке?

06. Индивидуальная проверка

40–45 минут. Для [4,-2,4,1,0], диапазона 0...4 запишите обычный фильтр, чётный фильтр, индексы выбранных чётных и исходник после вызова. Четыре ответа — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Строгие сравнения теряют 0 и 5; при $lo=hi=3$ теряются все тройки. Для основного входа ошибочный результат [3], правильный [5,5,0,3]. Пустой результат допустим, а сортировка и удаление повторов не соответствуют условию.

Практика 2: разбор и критерии

Ответ [2,2,-2,0]; для диапазона 2...2 — [2,2]; для пустого вектора — пусто. Остаток $x\%2==0$ работает и для отрицательных чётных. Полная программа выводит результат и размер неизменённого источника:

```
#include <iostream>
#include <vector>
std::vector<int> select_even(const std::vector<int>& a,
                             int lo, int hi) {
    std::vector<int> b;
    for (int x : a) {
        if (lo<=x && x<=hi && x%2==0) { b.push_back(x); }
    }
    return b;
}
int main() {
    std::vector<int> a{6,2,5,2,-2,0};
    auto b=select_even(a,-2,5);
    for (int x : b) { std::cout << x << " "; }
    std::cout << "\nsource-size=" << a.size() << "\n";
}
```

Обсуждение: возможный вывод

По нашему контракту порядок и количество вхождений важны. Повторяющиеся значения могут быть разными записями. Сравниваем весь результат с ожидаемой подпоследовательностью и весь исходник с его копией до вызова; одного размера мало.

Индивидуальная проверка: ответ и критерии

[4,4,1,0]; [4,4,0]; [0,2,4]; [4,-2,4,1,0]. По баллу за каждый результат; ноль чётный и входит в диапазон.

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: перебор списков и `vector`, логические условия, функции. 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	Создать новый набор по условию, сохранив порядок и повторения, и отличить фильтрацию от изменения источника.	Исходник, журнал тестов и объяснение результата
Повторение и обобщение; строки, условия и коллекции	Перенос условия в C++	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++: 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.