



Урок 10. C++: Построение и проверка графика

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 10 из 34 · 45 минут · Модуль «Графики»

Уровень: начинающий программист; C++17.

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

Программа может завершиться без ошибки, а график оказаться неверным. Проверим не только существование SVG, но и оси, число маркеров, порядок и координаты.

Результат урока: Создать открываемый браузером SVG с подписями, проверить его по данным и обработать отсутствие наблюдений.

01. Проблемная ситуация

0–5 минут. Серия (0,3),(2,7),(4,6). Запишите число маркеров и экранные координаты из формул $X=60+100*t$, $Y=240-20*h$. Что должно быть в файле для пустой серии?

02. Понятия и разобранный пример

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

Рисунок как структурированный файл

В стандартной библиотеке C++ нет встроенной функции построения графика. Здесь используем доступный путь без установки графического пакета: функция `chart` формирует SVG через стандартный файловый поток. SVG — текстовое описание изображения, которое открывает обычный браузер. Это не вызов сторонней библиотеки графиков.

Элемент `polyline` задаёт ломаную, `circle` — маркер, `text` — подпись. В нашем фиксированном шаблоне нет пользовательского текста и внешних ссылок. Время и высота переводятся по тем же формулам, что в уроке 9. Оси содержат отметки 0,2,4 мин и 0,5,10 см.

Контракт: до 20 пар целых, t в $0...4$, h в $0...10$, строгое возрастание t . До открытия файла проверяем данные. Пустота создаёт подписанные оси и No data; один маркер не нуждается в отрезке. После записи закрываем поток и проверяем его состояние. Работайте с новым именем в учебной папке: открытие для записи заменяет прежний файл.

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 (0,3),(2,7),(4,6) экранные точки (60,180),(260,100),(460,120). Ломаная сначала поднимается на 80 px, затем опускается на 20 px. Программа пишет `chart.svg`; открывайте его браузером после успешного вывода `saved`.

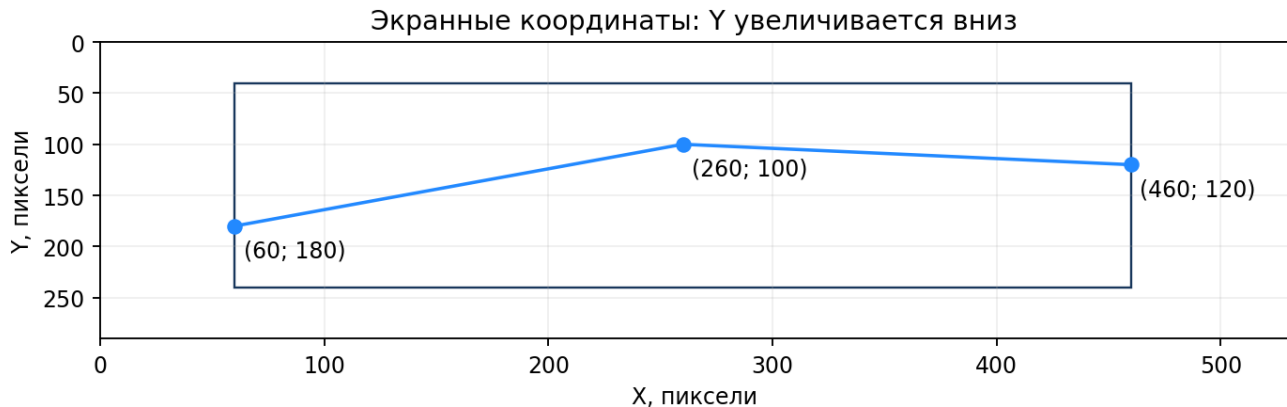
```
#include <fstream>
#include <iostream>
#include <vector>
#include <utility>
using Rows=std::vector<std::pair<int,int>>;
bool chart(const Rows& a, const char* path, bool line=true) {
    if (a.size()>20) return false;
    for (std::size_t i=0;i<a.size();++i) {
        auto [t,h]=a[i];
        if(t<0 || t>4 || h<0 || h>10) return false;
        if(i && t<=a[i-1].first) return false;
    }
    std::ofstream f(path);
    if(!f) return false;
    f<<R"(<svg xmlns="http://www.w3.org/2000/svg"
width="520" height="300" viewBox="0 0 520 300">
<rect width="520" height="300" fill="white"/>
<path d="M60 40V240H460" fill="none" stroke="black"/>
<g font-size="14" font-family="sans-serif">
<text x="55" y="260">0</text><text x="255" y="260">2</text>
<text x="455" y="260">4</text><text x="470" y="260">t,min</text>
<text x="35" y="245">0</text><text x="35" y="145">5</text>
<text x="30" y="45">10</text><text x="10" y="20">h,cm</text>
</g>";
    if(a.empty()) f<<R"(<text x="220" y="140">No data</text>);";
    if(line && a.size()>1) {
        f<<R"(<polyline fill="none" stroke="blue" points=");";
```



```

        for(auto [t,h]:a) f<<60+100*t<<" "<<240-20*h<<" ";
        f<<R("</>");
    }
    for(auto [t,h]:a) {
        f<<"<circle cx=' "<<60+100*t<<"' cy=' "<<240-20*h;
        f<<" r='4' fill='blue'/>";
    }
    f<<"</svg>"; f.close(); return bool(f);
}
int main() {
    bool ok=chart({{0,3},{2,7},{4,6}},"chart.svg");
    std::cout<<(ok?"saved":"failed")<<"\n";
    return ok?0:1;
}

```



Один circle соответствует одному наблюдению; полилиния соединяет соседние точки.

03. Практика 1. Проверяем материал

Проверяем SVG и ошибки записи

14–25 минут. Учитель выдаёт готовый исходник. Меняйте вход и имя файла, а не перепечатавайте шаблон SVG.

Вход в chart	Проверка файла
{{0,3},{2,7},{4,6}}	3 круга, 1 полилиния
{}; {{2,5}}	0 кругов; 1 круг (260,140)
{{0,0},{4,10}}	Крайние круги (60,240),(460,40)
{{4,6},{2,7}}	failed, новый файл не создаётся

2 минуты — прогноз; 4 — сборка и запуск пяти серий; 3 — открытие SVG и сверка подписей, числа маркеров и координат; 2 — журнал. Дополнительно задайте путь missing/chart.svg при отсутствующей папке missing. Критерии: верный рисунок и failed при невозможной записи.

04. Практика 2. Перенос способа

Отрезки или отдельные маркеры

25–33 минуты. Сохраните ту же серию в points.svg, передав третьим аргументом false. За 2 минуты предскажите отличие, за 4 запустите, за 2 проверьте количество circle и polyline просмотром текста SVG. Критерии: 3 круга, 0 полилиний, прежние координаты.

05. Обсуждение и границы вывода

33–40 минут. Достаточно ли сообщения saved? За 2 минуты назовите возможную содержательную ошибку, за 3 предложите независимую проверку, за 2 отделите ошибку данных от ошибки записи.

06. Индивидуальная проверка

40–45 минут. Для {{1,2},{3,9}} укажите два центра кругов, число отрезков при line=true и изменение при false. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

У основной серии три circle и одна polyline с тремя вершинами, то есть двумя отрезками. Пустота содержит No data, одиночная точка не содержит polyline. Для обратного времени false возвращается до открытия файла; прежний файл не стирается. При отсутствующей папке файл открыть нельзя: failed и код возврата 1.

Практика 2: разбор и критерии

В main вызовите `chart({{0,3},{2,7},{4,6}}, "points.svg", false)`. Координаты не меняются. В тексте будет три тега circle и ни одного polyline. Точки показывают только наблюдения; соединение не является дополнительными данными.

Обсуждение: возможный вывод

saved означает успешную запись потока, но не математическую корректность формулы. Ошибка $Y=240+20 \cdot h$ тоже может успешно сохраниться. Проверяем контрольные точки вручную, подписи и число элементов. failed объединяет ошибку входа и записи; для большого приложения полезны разные сообщения, но их причины здесь проверяются отдельными тестами.

Индивидуальная проверка: ответ и критерии

(160,200); (360,60); один отрезок; при false линия исчезает, оба круга остаются. По одному баллу за 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`). Для SVG нужен обычный браузер. Сторонний графический пакет не требуется; шаблон строится средствами стандартной библиотеки. 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: координатное преобразование урока 9, строки, файловые потоки. 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
Библиотеки рисования графиков (марафон)	Проверяем SVG и ошибки записи	Создать открываемый браузером SVG с подписями, проверить его по данным и обработать отсутствие наблюдений.
Структурирование решения и отладка	Отрезки или отдельные маркеры	Исходник, журнал обычных и граничных тестов; индивидуальный ответ

Источники и права на изображения

Исходная программа курса

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

W3C: линии и круги SVG

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

C++: файловые потоки

<https://eel.is/c++draft/fstream.syn>

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.