



Урок 9. C++: Подготовка данных и осей

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 9 из 34 · 45 минут · Модуль «Графики»

Уровень: начинающий программист; C++17.

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

На экране координата y растёт вниз. Чтобы высокий столбик данных не оказался внизу рисунка, нужно явно перевести координаты в пиксели.

Результат урока: Проверить серию и перевести физические координаты в экранные по заданным границам, сохранив дробный результат.

01. Проблемная ситуация

0–5 минут. Область рисунка: x от 60 до 460 пикселей, y от 40 до 240. Время 0...4 мин, высота 0...10 см. Где окажутся (0,0) и (4,10)? Запишите две экранные пары.

02. Понятия и разобранный пример

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

Два направления вертикальной оси

SVG использует начало в левом верхнем углу: x вправо, y вниз. Для области шириной 400 и высотой 200 px преобразование $X=60+100*t$, $Y=240-20*h$. Значение $h=0$ попадает на низ, $h=10$ — на верх. Оси здесь являются границами рабочей области.

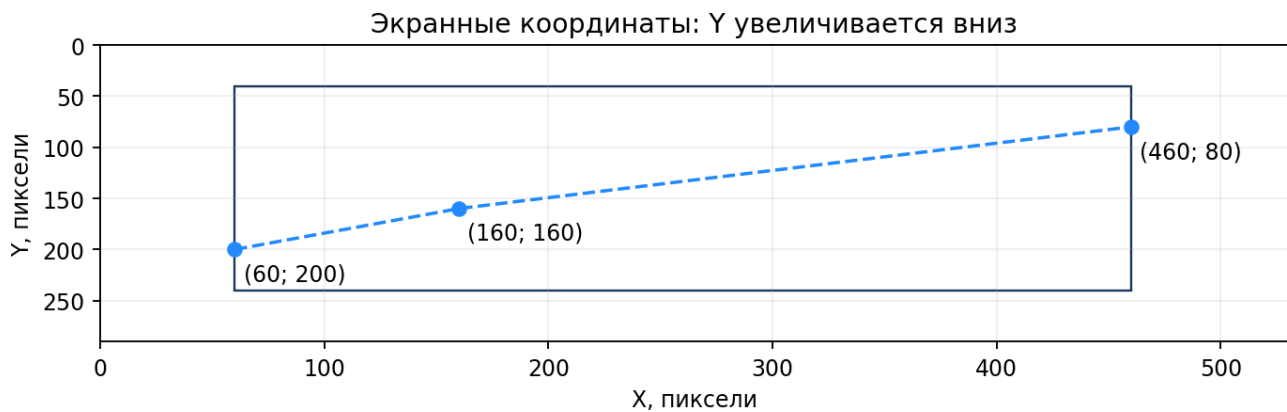
В C++ деление целых может отбросить дробную часть. Поэтому масштаб вычисляется как $400.0/4.0$ и $200.0/10.0$. Хотя текущие данные целые, другой диапазон может дать дробные экранные координаты. Размеры не должны зависеть от числа наблюдений.

Контракт: до 20 пар целых; $0 \leq t \leq 4$, $0 \leq h \leq 10$; времена строго возрастают. Пустая серия допустима: оси остаются, но точек нет. Урок готовит координаты и выводит их текстом; сохранение готового SVG будет на следующем занятии.

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 (0,2),(1,4),(4,8) получаем (60,200),(160,160),(460,80). Первый горизонтальный интервал 100 px , второй 300 px . На вертикали рост h на 2 см уменьшает Y на 40 px .

```
#include <iostream>
#include <vector>
#include <utility>
bool valid(const std::vector<std::pair<int,int>>& a) {
    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;
    }
    return true;
}
std::pair<double,double> screen(int t,int h) {
    return {60+400.0*t/4.0, 240-200.0*h/10.0};
}
int main() {
    std::vector<std::pair<int,int>> a{{0,2},{1,4},{4,8}};
    if (!valid(a)) { std::cout<<"invalid\n"; return 1; }
    std::cout<<"plot: X=60..460 Y=40..240\n";
    if (a.empty()) std::cout<<"No data\n";
    for (auto [t,h]:a) {
        auto [x,y]=screen(t,h);
        std::cout<<x<<" "<<y<<"\n";
    }
}
```



Перевод координат: одинаковая физическая высота всегда даёт одинаковую экранную строку.

03. Практика 1. Проверяем материал

Считаем контрольные точки

14–25 минут. Изменяйте только инициализацию `a` в `main`. Фигурные скобки задают `vector` пар.

Серия	Результат
<code>{{0,0},{4,10}}</code>	60 240; 460 40
<code>{}; {{2,5}}</code>	No data; 260 140
<code>{{2,1},{1,2}}</code>	invalid
<code>{{0,11}};</code> <code>{{1,1},{1,2}}</code>	invalid; invalid

2 минуты — расчёт; 4 — компиляция и запуски; 3 — проверка направлений осей и ошибок; 2 — журнал вход/код возврата/точки. Критерии: шесть серий таблицы, отсутствие целочисленного масштаба, сохранение порядка.

04. Практика 2. Перенос способа

Меняем область рисунка

25–33 минуты. Новый прямоугольник: слева 20, справа 320, сверху 30, снизу 180 px; физические диапазоны те же. За 2 минуты получите масштаб, за 4 измените `screen` и строку описания, за 2 проверьте (1,4) и (4,10). Критерии: ширина 300, высота 150, корректный знак и два ответа.

05. Обсуждение и границы вывода

33–40 минут. Почему нельзя заменить `t` индексом элемента? За 2 минуты сравните интервалы, за 3 объясните примером, за 2 обсудите случай, когда индексы всё же допустимы.

06. Индивидуальная проверка

40–45 минут. В исходной области вычислите экранную точку (3,7), масштаб по обеим осям и положение `h=0`. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Для допустимых серий код возврата 0, для недопустимых 1. Пустота печатает описание области и No data. При $h=10$ $Y=40$, при $h=0$ $Y=240$. Ошибка знака плюс вместо минуса помещает верхние точки ниже области.

Практика 2: разбор и критерии

screen возвращает $\{20+300.0*t/4.0, 180-150.0*h/10.0\}$. Описание plot: $X=20..320$ $Y=30..180$. Для (1,4) получаем (95,120), для (4,10) — (320,30). Одна минута теперь равна 75 px, один сантиметр — 15 px.

Обсуждение: возможный вывод

Индексы описывают порядок записей, а t — время. Для 0,1,4 индексы 0,1,2 искажают интервал. Индексы подходят, если горизонтальная величина действительно номер наблюдения и так подписана; равные интервалы времени нельзя молча предполагать.

Индивидуальная проверка: ответ и критерии

(360,100); 100 px/мин; 20 px/см; $h=0$ соответствует $Y=240$. Каждый пункт — один балл.

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`). 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: vector, пары, циклы, функции и вывод чисел. 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)	Считаем контрольные точки	Проверить серию и перевести физические координаты в экранные по заданным границам, сохранив дробный результат.
Структурирование решения и отладка	Меняем область рисунка	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: система координат SVG

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