



Урок 22. C++: изменение координат

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 22 из 34 · 45 минут · Модуль «Компьютерная графика на C++»

Уровень: начинающий программист; C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

Корпус перенесли, но круг остался на прежнем месте. Как сдвинуть составную фигуру целиком и заранее обнаружить обрезание?

Результат урока: Перенести группу SVG по введённым dx , dy , вычислить новые координаты и проверить её границы.

01. Проблемная ситуация

Группа содержит корпус $x=10\dots40$, $y=20\dots40$, круг с центром(30,10), $r=5$ и линию $y=15$ от $x=10$ до 40. При $dx=20$, $dy=10$ найдите новые координаты центра и рамки корпуса. За 5 минут объясните, почему одному элементу нельзя применить другой сдвиг.

02. Понятия и разобранный пример

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

Локальные координаты и перенос группы

Перенос задаётся $x'=x+dx$, $y'=y+dy$. У SVG y растёт вниз: положительный dy двигает вниз. Внутри g фигуры сохраняют локальные координаты, `transform="translate(dx dy)"` переносит всю группу. Ширины, высоты и радиус не изменяются; это не масштабирование.

Рамка всех видимых элементов исходной группы укладывается в $x=10\dots40$, $y=5\dots40$. Толщина линии 2 добавляет по 1 к её вертикальной рамке 14...16, что не выходит за общую рамку. После сдвига прибавляем dx к обеим горизонтальным границам, dy — к обеим вертикальным. Холст 100×70; в практических тестах оставляем запас от краёв, чтобы не спорить о сглаживании.

Вход — две целые строки dx , dy . Основная практика ограничена $0\leq dx\leq 50, 0\leq dy\leq 20$, что гарантирует помещение группы; программа намеренно не проверяет диапазон, это предусловие ввода. Во второй практике исследуем выход за него как проверку геометрии, не как корректный заказ. Файл `shift.svg` всегда записывается после запуска; наличие файла само по себе не гарантирует видимость всех элементов.

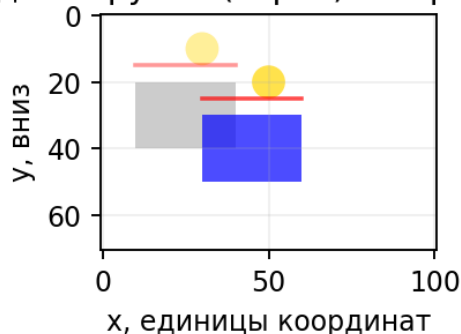
Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

При 20 10 корпус станет $x=30\dots60$, $y=30\dots50$; центр круга(50,20), радиус 5. Общая рамка $x=30\dots60$, $y=15\dots50$ помещается. Числа внутри `rect` и `circle` в исходнике остаются прежними: координаты меняет групповое преобразование.

```
#include <fstream>
#include <iostream>
int main() {
    int dx, dy;
    std::cin >> dx >> dy;
    std::ofstream out("shift.svg");
    out << "<svg xmlns='http://www.w3.org/2000/svg' "
        "width='400' height='280' viewBox='0 0 100 70'>\n";
    out << "<rect width='100' height='70' fill='white'>\n";
    out << "<g transform='translate(" << dx << " " << dy << ")'>\n";
    out << "<rect x='10' y='20' width='30' height='20' fill='blue'>\n";
    out << "<circle cx='30' cy='10' r='5' fill='gold'>\n";
    out << "<line x1='10' y1='15' x2='40' y2='15' stroke='red' stroke-width='2'>\n";
    out << "</g></svg>\n";
    return 0;
}
```



Исходная группа (серая) и перенос 20,10



Перенос 20 вправо и 10 вниз: общая рамка(10...40;5...40) переходит в(30...60;15...50); размеры не меняются.

03. Практика 1. Проверяем материал

Контроль новых координат без изменения формы

14–25 минут. Для каждого допустимого ввода сначала заполните в тетради координаты корпуса и центра круга, затем соберите программу и откройте сохранённый SVG.

Ввод dx dy	Общая новая рамка
0 0	x10...40, y5...40
20 10	x30...60, y15...50
50 20	x60...90, y25...60

2 минуты — прогноз, 4 — сборка и запуск, 3 — три файла в отдельных папках, 2 — сверка. Продукт — исходник и три пары «ввод → рамка». Критерии: переносится вся группа, размеры не меняются, направления осей объяснены, оба края рамки проверены.

04. Практика 2. Перенос способа

Находим перенос, который обрезает фигуру

25–33 минуты. Исследовательский ввод 70 35 выходит за предусловие основного заказа. Вычислите рамку, предскажите обрезание справа и снизу, затем проверьте в копии папки. Новый допустимый заказ требует сдвига 40 20. Укажите, какие границы изменились, и сохраните его как fixed.svg. Критерии: диагноз по числам, отличие файла от видимого результата, правильное исправление.

05. Обсуждение и границы вывода

33–40 минут. Автор проверил только центр круга и заявил, что вся группа внутри. Приведите контрпример по данным второго задания. Сопоставьте проверку одной точки с проверкой общей рамки.

06. Индивидуальная проверка

40–45 минут. Для $dx=-5$, $dy=5$ назовите общую рамку, центр круга и направление перемещения. Это отдельный геометрический вопрос, не ввод основной практики. Три критерия по 1 баллу.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Все три рамки даны в таблице. При 50 20 корпус x60...90, y40...60; центр(80,30), r5. Ошибка — менять ширину вместе с dx или применять dy как движение вверх. Полная рабочая программа:

```
#include <fstream>
#include <iostream>
int main() {
    int dx, dy;
    std::cin >> dx >> dy;
    std::ofstream out("shift.svg");
    out << "<svg xmlns='http://www.w3.org/2000/svg' "
        "width='400' height='280' viewBox='0 0 100 70'>\n";
    out << "<rect width='100' height='70' fill='white'/>\n";
    out << "<g transform='translate(" << dx << " " << dy << ")'>\n";
    out << "<rect x='10' y='20' width='30' height='20' fill='blue'/>\n";
    out << "<circle cx='30' cy='10' r='5' fill='gold'/>\n";
    out << "<line x1='10' y1='15' x2='40' y2='15' stroke='red' stroke-width='2'/>\n";
    out << "</g></svg>\n";
    return 0;
}
```

Практика 2: разбор и критерии

При 70 35 рамка x80...110, y40...75; справа выходит за 100 на 10, снизу за 70 на 5. При 40 20 рамка x50...80, y25...60 помещается. Для нового имени достаточно изменить строку ofstream; сами фигуры остаются прежними.

```
#include <fstream>
#include <iostream>
int main() {
    int dx, dy;
    std::cin >> dx >> dy;
    std::ofstream out("fixed.svg");
    out << "<svg xmlns='http://www.w3.org/2000/svg' "
        "width='400' height='280' viewBox='0 0 100 70'>\n";
    out << "<rect width='100' height='70' fill='white'/>\n";
    out << "<g transform='translate(" << dx << " " << dy << ")'>\n";
    out << "<rect x='10' y='20' width='30' height='20' fill='blue'/>\n";
    out << "<circle cx='30' cy='10' r='5' fill='gold'/>\n";
    out << "<line x1='10' y1='15' x2='40' y2='15' stroke='red' stroke-width='2'/>\n";
    out << "</g></svg>\n";
    return 0;
}
```

Обсуждение: возможный вывод

При 70 35 центр круга(100,45), а правый край корпуса 110 и низ 75 уже за пределами. Даже внутренний центр не гарантировал бы попадание всех объектов: нужно учитывать размеры и все крайние координаты. Проверка всей рамки даёт условие для всей группы.

Индивидуальная проверка: ответ и критерии

Рамка x5...35, y10...45; центр(25,15). На 5 влево и 5 вниз. Размеры не меняются.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, рабочую папку с правом записи, редактор, распечатку ученической части PDF, бумагу и ручку. Локальный компилятор G++ с C++17 или настроенная учителем IDE; сборка g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson. В Windows задайте lesson.exe. Запуск .\lesson в Linux/macOS, .\lesson.exe в Windows. Для 20–22 дополнительно обычный браузер: открывать локальный SVG, не C++-исходник. Все входы и ожидаемые результаты даны. Установку и проверку запуска учитель выполняет до занятия; детские аккаунты и интернет не нужны.

Уровень — начинающий программист, 7 класс. Предпосылки: целочисленные переменные, cin, запись SVG и параметры фигур. В паре автор вводит и объясняет, проверяющий сопоставляет с прогнозом, затем роли меняются. Без компьютера трассируйте полный код на бумаге, отмечайте координаты по



схеме и проверяйте готовые результаты; учитель демонстрирует заранее собранный пример. Бумажная трассировка не выдаётся за реальную компиляцию или запуск ученика. План 45 минут включает чтение, ввод, сборку и обсуждение. В графических задачах сначала прогноз, затем сравнение. 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
Изменение координат, линии и фигуры	Переносит группу по dx , dy	Новые координаты вычислены по формуле
Компьютерная графика	Проверяет общую рамку и обрезание	Различает созданный файл и корректное изображение

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-7-klass-programma>

W3C SVG2: преобразования координат

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

C++: строковые литералы

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

C++: файловые потоки

<https://eel.is/c++draft/input.output>

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: 17.10.2026.