



Урок 21. C++: линии, прямоугольники и окружности

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 21 из 34 · 45 минут · Модуль «Компьютерная графика на C++»

Уровень: начинающий программист; C++17.

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

Вместо круга вышел овал, а поверх линии оказалась синяя заливка. Какие свойства фигуры и порядок записи определяют результат?

Результат урока: Создать SVG из трёх примитивов, объяснить радиус круга, размеры прямоугольника и порядок наложения.

01. Проблемная ситуация

Для значка нужны корпус 10...60 по x, 30...60 по y; круг с центром(90,20), радиусом 10; красная линия от(10,25)до(60,25). На сетке 120×80 найдите рамку круга и назовите, что изменится при радиусе 15. Отличайте радиус от диаметра.

02. Понятия и разобранный пример

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

Размеры примитивов в SVG

rect задаётся левым верхним углом и размерами, а не двумя противоположными углами: $x=10$, $width=50$ означает правую границу 60. circle задаётся центром s_x , s_y и радиусом r ; при $r=10$ диаметр 20. ellipse имеет разные радиусы r_x , r_y и при их неравенстве не является кругом.

line соединяет x_1 , y_1 и x_2 , y_2 ; stroke задаёт цвет линии, stroke-width — толщину. У заполненного rect или circle цвет задаёт fill. Фигуры рисуются в порядке записи: более поздняя заливка может закрыть предыдущую. SVG в данном примере — векторный формат; его границы непрерывны, поэтому не переносите без оговорки правило включённых пиксельных концов Pillow.

Файл badge.svg записывается полностью стандартным C++ через fstream. Соберите исходник с -std=c++17 и запустите в папке с правом записи, затем откройте SVG в браузере. Одна внутренняя единица здесь показывается как 3 экранных единицы. Исходный шаблон содержит все атрибуты; внешние библиотеки и интернет не нужны.

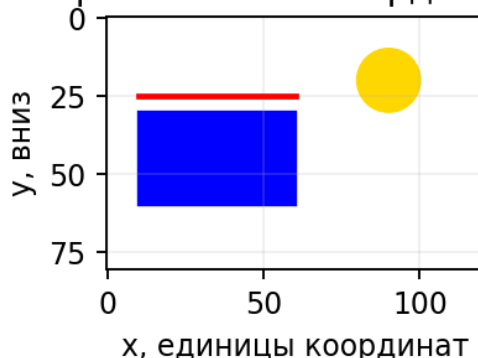
Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Корпус имеет ширину 50 и высоту 30. Рамка круга: $x=80...100$, $y=10...30$. Красная линия $y=25$ не пересекает корпус, верх которого 30. Точки(20,40),(90,20),(20,25) соответственно синие, золотистые и красные.

```
#include <fstream>
int main() {
    std::ofstream out("badge.svg");
    out << R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
        width="360" height="240" viewBox="0 0 120 80">
<rect width="120" height="80" fill="white"/>
<rect x="10" y="30" width="50" height="30" fill="blue"/>
<circle cx="90" cy="20" r="10" fill="gold"/>
<line x1="10" y1="25" x2="60" y2="25" stroke="red" stroke-width="2"/>
</svg>
)SVG";
    return 0;
}
```



Геометрия значка в координатах SVG



Значок 120×80: синий корпус(10,30),50×30; круг(90,20), r=10; линия y=25. SVG описывает размеры, а не пиксельные концы.

03. Практика 1. Проверяем материал

Увеличиваем круг и проверяем рамку

14–25 минут. Новое условие увеличивает круг и смещает его центр. Перед запуском вычислите рамку и диаметр.

Параметр	Новый вариант
Центр круга	(90,25)
Радиус	15
Корпус и линия	Без изменений
Файл	badge-big.svg

2 минуты — расчёт, 4 — изменение исходника и сборка, 3 — просмотр, 2 — объяснение партнёру. Продукт — полный C++-файл, badge-big.svg и рамка с единицами SVG. Критерии: центр, радиус, вычисленная рамка, отсутствие обрезания в 120×80.

04. Практика 2. Перенос способа

Переставляем слои пересекающихся фигур

25–33 минуты. Новая линия проходит по y=45 и пересекает корпус. Нужны два файла: front.svg с красной линией поверх синего корпуса и back.svg с линией под корпусом. Круг исходного примера сохраните. Самостоятельно измените координату и порядок записи. Предскажите цвет(20,45) в обоих случаях. Критерии: одна геометрия, разный порядок, правильные прогнозы.

05. Обсуждение и границы вывода

33–40 минут. Достаточно ли смотреть только на число фигур? Приведите ошибку радиуса и ошибку порядка, при которых фигур по-прежнему три. Предложите совместную геометрическую и цветовую проверку.

06. Индивидуальная проверка

40–45 минут. Для circle с центром(40,30)и r=7 назовите диаметр и рамку. Для rect x=5, y=8, width=20, height=12 назовите правую и нижнюю границы. Четыре критерия: диаметр, рамка круга, правый и нижний края.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Рамка $x=75\dots105$, $y=10\dots40$, диаметр 30. Все границы внутри 120×80 . Если заменить только sx/cy , радиус останется 10; если использовать $r=30$, диаметр будет 60. Полный новый вариант:

```
#include <fstream>
int main() {
    std::ofstream out("badge-big.svg");
    out << R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
        width="360" height="240" viewBox="0 0 120 80">
<rect width="120" height="80" fill="white"/>
<rect x="10" y="30" width="50" height="30" fill="blue"/>
<circle cx="90" cy="25" r="15" fill="gold"/>
<line x1="10" y1="25" x2="60" y2="25" stroke="red" stroke-width="2"/>
</svg>
)SVG";
    return 0;
}
```

Практика 2: разбор и критерии

В `front.svg` линия записана после корпуса и точка красная; в `back.svg` — перед корпусом и точка синяя. Две полные программы получаются из следующего кода: для `back.svg` перенесите строку `rect` корпуса после строки `line` и измените только имя файла. Полные версии ниже даны обе.

```
#include <fstream>
int main() {
    std::ofstream out("front.svg");
    out << R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
        width="360" height="240" viewBox="0 0 120 80">
<rect width="120" height="80" fill="white"/>
<rect x="10" y="30" width="50" height="30" fill="blue"/>
<circle cx="90" cy="20" r="10" fill="gold"/>
<line x1="10" y1="45" x2="60" y2="45" stroke="red" stroke-width="2"/>
</svg>
)SVG";
    return 0;
}
```

Второй отдельный исходник для `back.svg`:

```
#include <fstream>
int main() {
    std::ofstream out("back.svg");
    out << R"SVG(<svg xmlns="http://www.w3.org/2000/svg"
        width="360" height="240" viewBox="0 0 120 80">
<rect width="120" height="80" fill="white"/>
<circle cx="90" cy="20" r="10" fill="gold"/>
<line x1="10" y1="45" x2="60" y2="45" stroke="red" stroke-width="2"/>
<rect x="10" y="30" width="50" height="30" fill="blue"/>
</svg>
)SVG";
    return 0;
}
```

Обсуждение: возможный вывод

Три фигуры могут иметь неверные размеры или перекрывать друг друга. Проверяют параметры, рамки и внутренние контрольные точки. Точки на самой сглаженной границе для сравнения цветов неудобны; выбирайте внутренние области и середину достаточно толстой линии.

Индивидуальная проверка: ответ и критерии

Диаметр 14; $x=33\dots47$, $y=23\dots37$; правый край прямоугольника 25, нижний 20. По 1 баллу за каждый критерий.



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: запись SVG из C++, координаты и прямоугольная сетка. 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
Линии, прямоугольники, окружности	Записывает основные фигуры SVG из C++	Три примитива с проверенными размерами
Линии и замкнутые фигуры	Меняет порядок наложения	Два файла и прогноз контрольной точки

Источники и права на изображения

Исходная программа курса

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

W3C SVG2: основные геометрические фигуры

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