



Урок 5. Python: изменение координат

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 5 из 34 · 45 минут · Модуль «Графика»

Уровень: начинающий программист; Python 3.

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

Табличку сдвинули вправо, но она растянулась. Как переместить фигуру, сохранив её размеры и не выйдя за холст?

Результат урока: Перенести прямоугольник и линию по формулам $x' = x + dx$, $y' = y + dy$; вычислить допустимый сдвиг и проверить границы.

01. Проблемная ситуация

Исходный прямоугольник (20,30)–(60,50) нужно сдвинуть вправо на 30. Ученик изменил только правый угол: (20,30)–(90,50). Нарисуйте оба варианта на сетке и решите, какой сохраняет разности координат 40 и 20.

02. Понятия и разобранный пример

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

Перенос изменяет обе вершины

Перенос на dx по горизонтали и dy по вертикали одинаково изменяет все точки: $x' = x + dx$, $y' = y + dy$. На экранном холсте положительное dy перемещает вниз. Перенос не меняет разности координат, поэтому размеры сохраняются.

Для холста 120×90 допустимы $x = 0 \dots 119$ и $y = 0 \dots 89$. Чтобы целиком перенести рамку (20,30,60,50), нужно $-20 \leq dx \leq 59$ и $-30 \leq dy \leq 39$. Проверяют и минимум, и максимум, а не только один угол.

Pillow может обрезать часть рисуемой фигуры у границы. Отсутствие сообщения об ошибке не означает, что вся фигура поместилась. Проверка условий до рисования — способ найти ошибку задачи, а не ждать исключения.

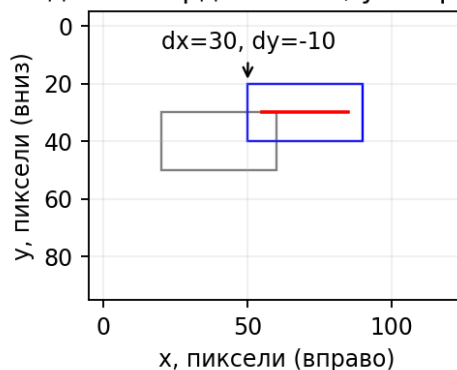
Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Для $dx=30$, $dy=-10$ получаем (50,20)–(90,40). Разности остаются 40 и 20. Линия внутри таблички (25,40)–(55,40) становится (55,30)–(85,30). В коде сохраняем исходный серый контур, затем рисуем перенесённую синюю табличку.

```
from PIL import Image, ImageDraw
dx, dy = 30, -10
im = Image.new("RGB", (120, 90), "white")
d = ImageDraw.Draw(im)
d.rectangle((20, 30, 60, 50), outline="gray")
d.rectangle((20+dx, 30+dy, 60+dx, 50+dy),
            outline="blue")
d.line((25+dx, 40+dy, 55+dx, 40+dy), fill="red")
im.save("move.png")
```



Рамки заданы координатами; у направлена вниз



Перенос таблички: исходная рамка (20,30)–(60,50), новая (50,20)–(90,40); $dx=30$, $dy=-10$.

03. Практика 1. Проверяем материал

Переносим табличку без искажения

Исходная рамка и размеры холста заданы в теории. Три строки таблицы — три независимых запуска, а не последовательные переносы.

Сдвиг (dx,dy)	Новая рамка: заполните до запуска
(30,−10)	Обе вершины и проверка границ
(−20,−30)	Обе вершины и проверка границ
(59,39)	Обе вершины и проверка границ

Для каждой строки вычислите новую рамку. Запустите полный пример, меняя только dx , dy и имя выходного файла `move1.png`, `move2.png`, `move3.png`. Сравните с бумажным расчётом. Результат — три PNG и заполненная таблица. Критерии: обе вершины изменены, размеры сохранены, граничные координаты проверены, три файла различимы.

04. Практика 2. Перенос способа

Проверяем предельный сдвиг

Новая рамка (70,10)–(100,40) на том же холсте. Требуется максимальный перенос вправо без изменения y . Самостоятельно вычислите dx и напишите полный код для `border.png`: белый фон, зелёный контур перенесённой рамки. Проверьте альтернативу $dx+1$: объясните, какая координата окажется недопустимой. Используйте готовый образец структуры программы, но новые числа.

05. Обсуждение и границы вывода

Ученик говорит: «Файл сохранился, значит границы правильные». Возразите по примеру $dx=20$. Альтернатива — считать обрезку допустимым художественным приёмом. При каком условии это приемлемо?

06. Индивидуальная проверка

На холсте 80×60 рамка (5,10)–(25,30). Найдите максимальные dx вправо и dy вниз, затем рамку при одновременном таком сдвиге. Критерии: пределы, обе вершины, сохранение разностей.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Рамки: (50,20)–(90,40); (0,0)–(40,20); (79,69)–(119,89). Все помещаются. Последний вариант касается правой и нижней границ. Код ниже — полный третий запуск; первые два получаются теми же изменениями dx , dy и имени, которые перечислены в задании.

```
from PIL import Image, ImageDraw
dx, dy = 59, 39
im = Image.new("RGB", (120, 90), "white")
d = ImageDraw.Draw(im)
d.rectangle((20, 30, 60, 50), outline="gray")
d.rectangle((20+dx, 30+dy, 60+dx, 50+dy),
            outline="blue")
d.line((25+dx, 40+dy, 55+dx, 40+dy), fill="red")
im.save("move3.png")
```

Практика 2: разбор и критерии

Максимум $dx=119-100=19$; новая рамка (89,10)–(119,40). При $dx=20$ правая координата 120 вне диапазона. Критерии: верный предел, обе вершины, объяснение выхода.

```
from PIL import Image, ImageDraw
dx = 19
im = Image.new("RGB", (120, 90), "white")
d = ImageDraw.Draw(im)
d.rectangle((70+dx, 10, 100+dx, 40), outline="green")
im.save("border.png")
```

Обсуждение: возможный вывод

Успешное сохранение не проверяет требование «вся фигура видна». Обрезка допустима, только если это явно задумано. Здесь условие требует полного переноса, поэтому расчёт границ обязателен.

Индивидуальная проверка: ответ и критерии

$dx=79-25=54$, $dy=59-30=29$. Рамка (59,39)–(79,59), разности 20 и 20 сохраняются. 3 балла.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, локальную папку с правом записи, Python/IDLE, распечатку ученической части PDF, бумагу и ручку. Только для уроков 4–5 заранее установите Pillow 10+ командой `python -m pip install Pillow`; если интерпретатор называется `python3`, используйте его. Установка учителем до занятия. PNG открываются обычным просмотрщиком, графическое окно Python не требуется. Все программы, входные данные, таблицы и ключи находятся в статье и PDF. Детские аккаунты, сеть и регистрация не нужны.

Уровень — начинающий, 7 класс. Предпосылки: координаты и графические примитивы уроков 3–4. Работа в паре: автор прогнозирует и вводит код, проверяющий ведёт журнал; затем роли меняются. На бумаге трассируйте готовый код и входные данные; для графики используйте координатную сетку и схему. Учитель может показать заранее полученный вывод/PNG. Бумажный вариант проверяет понимание, но не выдаётся за реальный запуск учеником. Чтение, ввод и проверка включены в план. Матрица программы — материал учителя.



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	Применяет перенос ко всем точкам	Размеры сохранены
Система координат	Сравнивает рамку с $0 \dots W-1$ и $0 \dots H-1$	Граничный тест без обрезки

Источники и права на изображения

Исходная программа курса

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

Pillow: координаты ImageDraw

<https://pillow.readthedocs.io/en/stable/reference/ImageDraw.html>

Python. Подключение модулей

<https://docs.python.org/3/tutorial/modules.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: 17.10.2026.