



Урок 14. Python: рефакторинг рисунка

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 14 из 34 · 45 минут · Модуль «Процедуры»

Уровень: начинающий программист; Python 3.

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

В двух нарисованных домиках заменили цвет стен, но исправили только один фрагмент. Как сделать одно изменение правилом для всех домиков?

Результат урока: Вынести рисунок домика в процедуру, собрать три экземпляра и проверить, что перенос не меняет их форму.

01. Проблемная ситуация

Два домика состоят из одинаковых стен и крыши. Для первого вершина крыши (25,10), стены (10,25)–(40,50); для второго вершина (85,10), стены (70,25)–(100,50). Найдите общий сдвиг и запишите, какие числа приходится менять при копировании. За 5 минут предложите название одного повторяемого действия.

02. Понятия и разобранный пример

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

Параметры задают положение, тело задаёт форму

Рефакторинг меняет устройство программы при сохранении требуемого результата. Сначала фиксируем контрольные признаки рисунка, затем выделяем повтор. Процедура `house` получает `x` и `y` вершины левого края условной рамки и цвет стен; тело содержит координаты относительно этой опоры.

В примере `y+15` — верх стен, `y+40` — низ; вершина крыши имеет координаты (`x+15`, `y`). При переносе меняется опора, а разности координат остаются прежними. Определение `def` ещё не рисует дом: рисование происходит при вызове. `d` — заранее созданный инструмент рисования, доступный процедуре; параметры `x`, `y`, `color` получают значения каждого вызова.

Перед рефакторингом сохраните эталонный рисунок или координатную карточку. Сравнивать нужно форму, число домов, порядок и контрольные точки. Меньшее число строк само по себе не доказывает правильность. Код полностью дан; библиотека устанавливается учителем заранее.

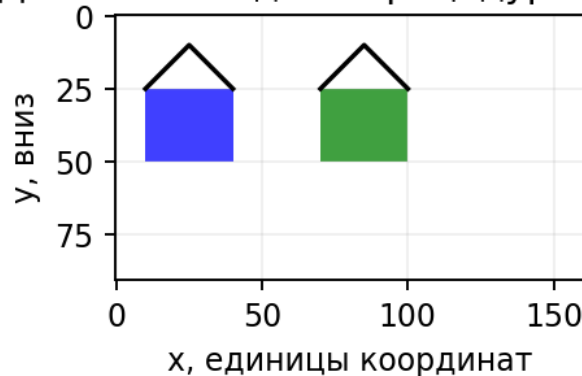
Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

Первый вызов задаёт синие стены от (10,25) до (40,50), второй — зелёные от (70,25) до (100,50). Точка (20,35) синяя, (80,35) зелёная; крыши имеют одинаковые смещения. Файл `houses.png` создаётся в рабочей папке.

```
from PIL import Image, ImageDraw
im = Image.new("RGB", (160, 90), "white")
d = ImageDraw.Draw(im)
def house(x, y, color):
    d.rectangle((x, y + 15, x + 30, y + 40), fill=color)
    d.line((x, y + 15, x + 15, y, x + 30, y + 15), fill="black", width=2)
house(10, 10, "blue")
house(70, 10, "green")
im.save("houses.png")
```



Два вызова одной процедуры house



Опора домика (x, y): стены ниже на 15; ширина 30; высота рамки 40. Перенос второго домика относительно первого: dx=60, dy=0.

03. Практика 1. Проверяем материал

Собираем деревню из трёх вызовов

14–25 минут. Используйте готовую процедуру и таблицу. Третий дом должен быть добавлен вызовом, а не копированием команд тела.

Дом	Опора и цвет
А	(10,10), blue
Б	(70,10), green
В	(115,35), red

За 2 минуты вычислите рамки всех домов, за 4 измените программу, за 3 откройте village.png, за 2 заполните журнал «прогноз → наблюдение». Проверьте точки (20,35), (80,35), (125,60). Продукт — полный файл программы, PNG и три строки проверки. Критерии: одно определение house, три вызова, верные рамки и цвета, отсутствие обрезания.

04. Практика 2. Перенос способа

Добавляем размер без изменения смысла параметров

25–33 минуты. Новое условие: ширина w задаётся параметром; высота полной рамки равна w, крыша занимает верхнюю половину. Холст 180×100, дома (10,10, w=40, blue) и (90,20, w=60, green). Сначала вычислите верх стен и вершины, затем напишите полный код. Ширины чётные, поэтому целочисленная половина точна. Критерии: новый параметр действительно используется во всех координатах, оба размера и границы проверены.

05. Обсуждение и границы вывода

33–40 минут. Можно ли считать рефакторинг удачным, если картинка похожа, но третий дом оказался частично за краем? Защитите критерий «результат сохранён», приведите возражение против проверки только на глаз и предложите минимум три теста.

06. Индивидуальная проверка

40–45 минут. Для исходной house(45,20,"blue") назовите рамку стен, вершину крыши и один внутренний пиксель стен. Запишите вызов для переноса на 10 вправо и 5 вверх. Четыре критерия: рамка, вершина, точка, новый вызов.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Дом В занимает по стенам $x=115\dots145$, $y=50\dots75$; вершина крыши (130,35). Все фигуры помещаются в 160×90 . Контрольные цвета: синий, зелёный, красный. По одному баллу за четыре критерия; исправление координат внутри отдельных копий не считается рефакторингом.

```
from PIL import Image, ImageDraw
im = Image.new("RGB", (160, 90), "white")
d = ImageDraw.Draw(im)
def house(x, y, color):
    d.rectangle((x, y + 15, x + 30, y + 40), fill=color)
    d.line((x, y + 15, x + 15, y, x + 30, y + 15), fill="black", width=2)
house(10, 10, "blue")
house(70, 10, "green")
house(115, 35, "red")
im.save("village.png")
```

Практика 2: разбор и критерии

Верх стен: для $w=40$ это $y+20=30$, для $w=60$ это $y+30=50$. Вершины (30,10) и (120,20). Низ стен 50 и 80; правые края 50 и 150. Типичная ошибка — оставить старое +15: тогда пропорции не масштабируются. Пример ниже реализует новое условие целиком.

```
from PIL import Image, ImageDraw
im = Image.new("RGB", (180, 100), "white")
d = ImageDraw.Draw(im)
def house(x, y, w, color):
    d.rectangle((x, y + w // 2, x + w, y + w), fill=color)
    d.line((x, y + w // 2, x + w // 2, y, x + w, y + w // 2),
          fill="black", width=2)
house(10, 10, 40, "blue")
house(90, 20, 60, "green")
im.save("sizes.png")
```

Обсуждение: возможный вывод

Нужны исходные два дома, новый допустимый вызов и граничное положение. Изменение структуры не разрешает незаметно менять форму или терять часть рисунка. Новая возможность размера — отдельное расширение требований, а не доказательство сохранения старого результата.

Индивидуальная проверка: ответ и критерии

Стены (45,35)–(75,60), вершина (60,20), например (55,45). Новый вызов `house(55,15,"blue")`. По 1 баллу за каждый пункт.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, рабочую папку с правом записи, редактор, распечатку ученической части PDF, бумагу и ручку. Python 3.11+, Pillow10+ заранее установлен учителем; команда `python -m pip install Pillow`. PNG открывают обычным просмотрщиком. 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: процедуры с параметрами, координаты и команды Pillow. 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
Процедуры с параметрами	Выделяет повторяемый рисунок	Одна процедура и три вызова
Рефакторинг	Сопоставляет рамки и контрольные точки	Прежняя форма сохранена при переносе

Источники и права на изображения

Исходная программа курса

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

Python: условия, циклы и функции

<https://docs.python.org/3.11/tutorial/controlflow.html>

Pillow: примитивы ImageDraw

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