



Урок 17. Python: графический мини-проект

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 17 из 34 · 45 минут · Модуль «Марафон по программированию»

Уровень: начинающий программист; Python 3.

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

Индикатор показывает три выполненных шага даже тогда, когда введён ноль. Как защитить мини-проект не словами «работает», а набором проверок?

Результат урока: Объединить ввод, условия, цикл и процедуру в графическом индикаторе и защитить результат по проверяемым требованиям.

01. Проблемная ситуация

Требуется показать число готовых этапов n от 0 до 3: первые n ячеек зелёные, остальные серые. При $n=0$ зелёных быть не должно, при $n=3$ — ровно три. На бумаге нарисуйте четыре ожидаемых варианта и сформулируйте ошибку в условии $i \leq n$.

02. Понятия и разобранный пример

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

Требования к мини-проекту и разделение действий

Мини-проект имеет вход, ограничения и наблюдаемый продукт. Вход — одна целая строка n ; допустимы $0 \dots 3$. Выход — `panel.png` и строка `saved`, либо только `error` для числа вне диапазона. Каждый запуск выполняйте в отдельной пустой папке либо учитывайте, что старый PNG после ошибочного ввода остаётся старым: это не новый результат.

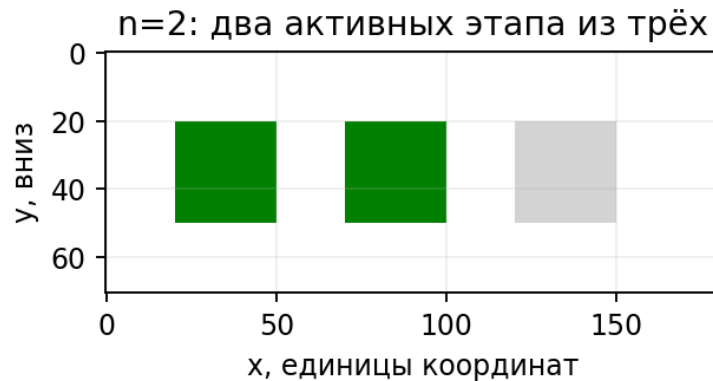
Процедура `slot` отвечает за одну ячейку, цикл — за размещение трёх ячеек, условие $i < n$ — за цвет. Индексы 0,1,2; поэтому при $n=2$ активны 0 и 1. Параметр `active` содержит логическое значение, выбранное сравнением. Условное выражение выбирает цвет и не является отдельным игровым решением.

Холст 180×70 : ячейки начинаются при $x=20, 70, 120$, имеют рамку 30×30 по разностям и $y=20 \dots 50$. Край последней 150 меньше 180. Сначала проверяйте число ячеек и границы, потом оформление. Индикатор учебный, не оценка способностей ребёнка и не запись реальных учебных результатов.

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 $n=2$ первые две ячейки зелёные, третья светло-серая. Контрольные точки (30,30), (80,30), (130,30) лежат внутри ячеек. Полный пример объединяет уже изученные конструкции; ничего не требуется скачивать ученику.

```
from PIL import Image, ImageDraw
n = int(input())
if not 0 <= n <= 3:
    print("error")
else:
    im = Image.new("RGB", (180, 70), "white")
    d = ImageDraw.Draw(im)
    def slot(x, active):
        color = "green" if active else "lightgray"
        d.rectangle((x, 20, x + 30, 50), fill=color)
    for i in range(3):
        slot(20 + 50 * i, i < n)
    im.save("panel.png")
    print("saved")
```



При $n=2$: ячейки с индексами 0 и 1 зелёные, 2 серая. Проверяются три внутренние точки и правый край 150.

03. Практика 1. Проверяем материал

Приёмочные тесты индикатора из трёх ячеек

14–25 минут. Выполните приёмку готового примера по таблице, затем исправьте в копии одну ошибку: замените $i < n$ на $i \leq n$ и найдите контрпример.

Ввод	Ожидаемый продукт
0	3 серые ячейки; saved
1	1 зелёная и 2 серые; saved
2	2 зелёные и 1 серая; saved
3	3 зелёные; saved
4	error; новый PNG не создаётся

2 минуты — прогноз, 4 — запуск и сохранение вариантов в отдельных папках 0...4, 3 — проверка цветов, 2 — защита исправления партнёру. Продукт — программа, четыре корректных PNG, журнал ошибки для 0. Критерии: точное число активных ячеек, правильный вывод, контроль границ, объяснение контрпримера.

04. Практика 2. Перенос способа

Проект из четырёх ячеек на новом холсте

25–33 минуты. Новый заказ: четыре ячейки, холст 230×70 , допустимый ввод 0...4, файл panel4.png. Формула $x=20+50*i$ прежняя. Самостоятельно согласуйте проверку, цикл и имя файла. Проверьте 0,4,5; найдите правый край последней ячейки и её индекс. Критерии: обновлены все три требования, не изменена форма ячейки, границы доказаны.

05. Обсуждение и границы вывода

33–40 минут. На защите автор говорит: «Я добавил красивую рамку, значит проект готов». Партнёр требует тесты 0 и 3. Обоснуйте, какие требования обязательны, какие являются оформлением, и что проверить после нового требования о четырёх ячейках.

06. Индивидуальная проверка

40–45 минут. Без запуска предскажите цвет точки (130,30) при $n=1$ и при $n=3$; назовите контрпример для $\leq$ и причину, по которой старый PNG нельзя считать результатом ошибочного запуска. Четыре критерия по 1 баллу.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Ошибка $\leq$ при $n=0$ красит ячейку с индексом 0 в зелёный; при $n=2$ красит все три. Возврат к $<$ исправляет оба случая. Успешный тест 3 один не обнаруживает ошибку. Полный исправленный код совпадает с примером.

```
from PIL import Image, ImageDraw
n = int(input())
if not 0 <= n <= 3:
    print("error")
else:
    im = Image.new("RGB", (180, 70), "white")
    d = ImageDraw.Draw(im)
    def slot(x, active):
        color = "green" if active else "lightgray"
        d.rectangle((x, 20, x + 30, 50), fill=color)
    for i in range(3):
        slot(20 + 50 * i, i < n)
    im.save("panel.png")
    print("saved")
```

Практика 2: разбор и критерии

Индексы 0...3, последний $x=170$, правый край $200<230$. Для 0 все серые, для 4 все зелёные, для 5 только error. Меняются верхняя граница, `range(4)`, размер холста и имя файла.

```
from PIL import Image, ImageDraw
n = int(input())
if not 0 <= n <= 4:
    print("error")
else:
    im = Image.new("RGB", (230, 70), "white")
    d = ImageDraw.Draw(im)
    def slot(x, active):
        color = "green" if active else "lightgray"
        d.rectangle((x, 20, x + 30, 50), fill=color)
    for i in range(4):
        slot(20 + 50 * i, i < n)
    im.save("panel4.png")
    print("saved")
```

Обсуждение: возможный вывод

Оформление полезно после проверки контракта. Приёмка включает допустимые края, неверный ввод, число объектов, цвета и границы. После расширения до 4 прежний тест 3 уже не проверяет правый край новой схемы. Удобное разбиение на процедуру и цикл сокращает число мест изменения, но не отменяет повторные тесты.

Индивидуальная проверка: ответ и критерии

При 1 точка серая, при 3 зелёная. Контрпример $n=0$. В ветке error файл не сохраняется, поэтому оставшийся файл относится к предыдущему запуску.

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: ввод, условия, `for`, `range`, процедуры и 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
Марафон: повторение и обобщение	Объединяет ввод, условие, цикл и процедуру	Готовый PNG и журнал приёмочных тестов
Рефакторинг и параметры	Расширяет индикатор с 3 до 4 ячеек	Все ограничения согласованы, нет обрезания

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.