



Урок 15. Python: случайные числа и ветвление

Часть 1. Материалы ученика

Программирование на Python и C++ · 7 класс

Урок 15 из 34 · 45 минут · Модуль «Марафон по программированию»

Уровень: начинающий программист; Python 3.

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

В игре каждый раз другое секретное число. Как убедиться, что ветка «вы угадали» вообще работает?

Результат урока: Создать учебную игру с одной попыткой и отделить случайный запуск от фиксированных тестов всех ветвей.

01. Проблемная ситуация

Ученик пять раз получил higher и решил, что программа правильная. Но ветки win и lower он не видел. Для секрета 4 запишите по одному вводу, который должен привести в каждую ветку. Не запускайте игру, пока не записали ожидаемые ответы.

02. Понятия и разобранный пример

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

Случайный выбор и предсказуемая проверка

Псевдослучайный генератор вычисляет последовательность по внутреннему состоянию. randint(1,6) выбирает целое с включёнными концами 1 и 6. Такой генератор подходит нашей учебной игре, но не является средством защиты паролей. На уроке нет ставок, покупок или соревнования по случайному результату.

Полная цепочка if/elif/else выбирает ровно один ответ. Равенство проверяем ==, а = присваивает значение. Если догадка меньше секрета, ответ higher предлагает увеличить её; если больше, lower — уменьшить. Вход всегда две отдельные строки с целыми: секрет или 0, затем догадка.

Для проверки разрешаем задать секрет явно от 1 до 6; 0 включает случайный выбор. Это учебный тестовый режим, а не секретная защита. Перед игровыми ветками проверяем диапазоны. Нецелые строки не входят в контракт этого урока: обработка ошибок int здесь не изучается. Случайный запуск проверяет возможные ответы, но не заменяет тесты каждой ветви.

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 вводе 4 и 2 вывод higher, при 4 и 4 — win, при 4 и 6 — lower. Для 0 и 3 заранее известен только набор возможных ответов. Программа ничего не печатает до двух вводов; подсказки к вводу находятся в условии, поэтому машинный вывод можно сравнить точно.

```
from random import randint
secret = int(input())
guess = int(input())
if not (0 <= secret <= 6 and 1 <= guess <= 6):
    print("error")
else:
    if secret == 0:
        secret = randint(1, 6)
    if guess == secret:
        print("win")
    elif guess < secret:
        print("higher")
    else:
        print("lower")
```



Python: случайные числа и ветвление

Догадка 2	$2 < 4 \rightarrow \text{higher}$
Догадка 4	$4 = 4 \rightarrow \text{win}$
Догадка 6	$6 > 4 \rightarrow \text{lower}$
Секрет 0	<code>randint(1,6)</code> , оба конца включены

Фиксированный секрет 4: догадки 2,4,6 попадают в разные ветви; режим 0 выбирает секрет из 1...6.

03. Практика 1. Проверяем материал

Три ветви и проверка допустимого ввода

14–25 минут. Таблица — готовый тестовый набор. В каждом тесте запускайте программу заново и вводите два числа на отдельных строках.

Ввод: секрет / догадка	Ожидаемый вывод
4 / 2	higher
4 / 4	win
4 / 6	lower
1 / 0	error
7 / 3	error

2 минуты — прогноз, 4 — ввод кода, 3 — пять запусков, 2 — исправление и журнал. Дополнительно проверьте границы 1/1 и 6/6. Продукт: код и семь записей «ввод → ожидание → факт». Критерии: все три игровые ветви, проверка диапазона, включённые границы, вывод одной строки.

04. Практика 2. Перенос способа

Как меняется контракт для другого диапазона

25–33 минуты. Перенос: игра работает с секретом и догадкой от 1 до 10; 0 по-прежнему включает случайный выбор. Напишите полный вариант и замените все относящиеся к диапазону числа, не меняя смысл ответов. Проверьте 10/10, 10/9, 1/2, 0/11 и 11/5. Критерии: согласованы генератор и обе проверки, тесты границ и ошибок, верные направления подсказок.

05. Обсуждение и границы вывода

33–40 минут. Что лучше для отладки: сто случайных запусков или пять специально выбранных? Сравните охват ветвей и повторяемость, назовите полезную роль каждого подхода.

06. Индивидуальная проверка

40–45 минут. Для секрета 5 укажите вывод при догадках 1,5,6. Назовите две строки программы, которые обязательно менять при переходе к диапазону 1...12, и объясните роль 0. По одному баллу: три ответа вместе, проверки диапазонов, генератор, роль тестового режима.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Пять первых ответов даны в таблице; два граничных дают win. Проверка неверного диапазона должна идти раньше случайного выбора и сравнения. Одинаковая случайная серия не доказывает отсутствие ошибки в ветвлении. Полная программа практики совпадает с разобранным примером.

```
from random import randint
secret = int(input())
guess = int(input())
if not (0 <= secret <= 6 and 1 <= guess <= 6):
    print("error")
else:
    if secret == 0:
        secret = randint(1, 6)
    if guess == secret:
        print("win")
    elif guess < secret:
        print("higher")
    else:
        print("lower")
```

Практика 2: разбор и критерии

Выводы: win, higher, lower, error, error. Если заменить только randint, секрет 10 в тестовом режиме останется запрещён; если заменить только проверку, случайная игра не достигнет 7...10. Ниже согласованный вариант.

```
from random import randint
secret = int(input())
guess = int(input())
if not (0 <= secret <= 10 and 1 <= guess <= 10):
    print("error")
else:
    if secret == 0:
        secret = randint(1, 10)
    if guess == secret:
        print("win")
    elif guess < secret:
        print("higher")
    else:
        print("lower")
```

Обсуждение: возможный вывод

Фиксированные тесты позволяют повторить ошибку и гарантированно проверить нужную ветвь. Случайные запуски дополняют их, но могут много раз выбирать похожие случаи. Проверка должна учитывать контракт входа и обе границы, а не только число успешных запусков.

Индивидуальная проверка: ответ и критерии

higher, win, lower. Меняют обе верхние границы в условии и верхнюю границу randint. Ноль выбирает случайный секрет, а не является допустимой догадкой. Четыре содержательных пункта — 4 балла.

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+; внешние библиотеки не нужны. 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: целые числа, сравнения, ввод и отступы. 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
Случайные и псевдослучайные числа	Разделяет случайный режим и фиксированный секрет	Повторяемые тесты всех ветвей
Ветвления, марафон	Строит и испытывает игру по контракту	Верные ответы на нормальных и граничных вводах

Источники и права на изображения

Исходная программа курса

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

Python: генератор псевдослучайных чисел

<https://docs.python.org/3.11/library/random.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.