



Урок 16. Защита программы с ветвлениями

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 16 из 33 · 45 минут · Модуль «Повторение и обобщение: марафон по ветвлениям»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Две версии программы робота выдают одинаковые команды на обычном маршруте. Защита проекта должна показать, что они верны и для нулевого пути, нехватки энергии и ошибочного ввода.

Результат урока: Представить две реализации одного контракта, обосновать полноту ветвей и защитить тесты независимым расчётом.

01. Проблемная ситуация

0–5 минут. В учебной модели энергия e — целое $0 \dots 100$, расстояние d — целое $0 \dots 20$. Один шаг стоит 3 единицы энергии. При $d=0$ ответ stay, иначе при достаточной энергии go, иначе charge. Для $(0,0), (5,2), (6,2)$ сначала ответьте без кода.

02. Понятия и разобранный пример

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

Один контракт, две структуры

Контракт задаёт приоритеты: недопустимый вход → error; допустимый нулевой путь → stay; ненулевой путь при $e \geq 3d$ → go; иначе charge. Не начинайте с проверки достаточности энергии, иначе $(0,0)$ ошибочно получит go.

В первой реализации последовательно исключаем случаи ранним return. Во второй после проверки диапазона используются вложенные if. Обе должны возвращать ровно одну строку. Эквивалентность доказываем разбиением контракта на четыре случая, а затем подтверждаем тестами.

Защита — не чтение исходника вслух. Нужно предъявить контракт, таблицу ожидаемых результатов, реальный вывод и объяснение одного контрпримера для ошибочного кода. Совпадение двух версий само по себе недостаточно: обе могут забыть нулевой путь. Энергия и расстояние — учебные величины, не параметры реального устройства.

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 $e=5, d=2$ требуется 6, значит charge. Для $6,2$ равенства достаточно: go. Для $0,0$ действует отдельное правило stay независимо от запаса. Недопустимое $e=101$ даёт error даже при $d=0$.

```
def route_a(e, d):
    if e < 0 or e > 100 or d < 0 or d > 20:
        return "error"
    if d == 0:
        return "stay"
    if e >= 3*d:
        return "go"
    return "charge"

def route_b(e, d):
    if 0 <= e <= 100 and 0 <= d <= 20:
        if d == 0:
            return "stay"
        else:
            if e >= 3*d:
                return "go"
            else:
                return "charge"
    else:
        return "error"

for e,d in [(0,0),(5,2),(6,2),(100,20),(101,0)]:
    print(e,d,route_a(e,d),route_b(e,d))
```



Ошибка диапазона	error; последующие проверки не нужны
Допустимо, $d=0$	stay
$d>0$, $e \geq 3d$	go
$d>0$, $e < 3d$	charge

Четыре взаимоисключающих случая учебного маршрута; ошибка проверяется до остановки и движения.

03. Практика 1. Проверяем материал

Протокол защиты и поиск отличий

14–25 минут. Подготовьте защиту обеих функций по 11 входам таблицы. Сверьте не только равенство версий, но и расчёт $3d$.

e,d	Ожидаемая команда
0,0;100,0	stay;stay
0,1;2,1;3,1	charge;charge;go
59,20;60,20;100,20	charge;go;go
−1,1;101,0;3,21	error;error;error

2 минуты — прогноз;4 — запуски обеих версий;3 — замените $\geq$ на $>$ только в `route_a` и найдите расхождение;2 — восстановите и проведите минутную защиту по четырём случаям. Критерии: полный контракт,11 ответов, контрпример, аргумент покрытия.

04. Практика 2. Перенос способа

Резерв энергии после маршрута

25–33 минуты. Новое правило: при $d>0$ после пути должно остаться хотя бы 5 единиц энергии. При $d=0$ по-прежнему stay, диапазоны прежние. За 2 минуты вычислите(10,2),(11,2),(4,0),за 4 напишите `route_reserve`,за 2 добавьте(65,20),(64,20),(101,0). Критерии: резерв, равенство, особый ноль, защита диапазона.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты подготовьте защиту утверждения «два совпавших результата доказывают готовность»,за 3 приведите возражения на нулевом пути и границе энергии,за 2 согласуйте минимальный комплект доказательств.

06. Индивидуальная проверка

40–45 минут. Для(14,3) найдите `route_a` и `route_reserve`; укажите остаток после пути и объясните, изменился ли контракт во второй практике. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Смену $\geq$ на $>$ выявляет (3,1) или (60,20): правильный go, ошибочный charge. Нулевой путь проверяет порядок приоритетов. Четыре случая разделяют все целые входы: сначала ошибка, потом три допустимых варианта.

Практика 2: разбор и критерии

charge, go, stay, go, charge, error. Для двух шагов требуется $3 \cdot 2 + 5 = 11$. Условие нового правила равно $e - 3d \geq 5$. Нулевая дистанция имеет отдельный ответ, даже если $e < 5$.

```
def route_reserve(e, d):
    if e < 0 or e > 100 or d < 0 or d > 20:
        return "error"
    if d == 0:
        return "stay"
    if e >= 3*d + 5:
        return "go"
    return "charge"

for e,d in [(10,2), (11,2), (4,0), (65,20), (64,20), (101,0)]:
    print(e,d,route_reserve(e,d))
```

Обсуждение: возможный вывод

Нужны контракт, независимые ожидаемые ответы, покрытие всех четырёх случаев, граничные тесты и объяснение ветвей. Проверка эквивалентности двух реализаций дополняет, но не заменяет проверку правильности каждой.

Индивидуальная проверка: ответ и критерии

go;go;остаток $14 - 9 = 5$;контракт изменён добавлением обязательного резерва для ненулевого пути. Это расширение требований, а не просто рефакторинг.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск `python lesson.py` (либо `python3 lesson.py`). Внешние библиотеки не нужны. Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 8 класс. Предпосылки: Python-функции, if/elif, таблицы тестов, целочисленные вычисления. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. 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
Повторение и обобщение: марафон по ветвлениям	Представить две реализации одного контракта, обосновать полноту ветвей и защитить тесты независимым расчётом.	Исходник, таблица прогнозов и фактических результатов
Составлять и отлаживать программы; использовать условные операторы	Резерв энергии после маршрута	Самостоятельное решение и индивидуальная проверка по четырём критериям

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Python: управление выполнением

<https://docs.python.org/3/tutorial/controlflow.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: 18.10.2026.