



Урок 12. Тарифы и кусочные правила

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 12 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>

В учебной мастерской первые десять деталей стоят по 5 жетонов, следующие десять по 3, остальные по 2. Почему заказ из 11 деталей не стоит 33 жетона?

Результат урока: Реализовать накопительный тариф, проверить стыки диапазонов и отличить его от цены за весь заказ.

01. Проблемная ситуация

0–5 минут. Все цены вымышлены. Сравните два расчёта для 11 деталей: $11 \cdot 3$ и $10 \cdot 5 + 1 \cdot 3$. Какой соответствует словам «следующие десять»? Предскажите, может ли общая стоимость уменьшиться при добавлении детали.

02. Понятия и разобранный пример

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

Новая ставка действует только на новую часть

Накопительный тариф сохраняет стоимость уже пройденных ступеней. Для $n \leq 10$ это $5n$. Для $10 < n \leq 20$: $50 + 3(n - 10)$. Для $n > 20$: $80 + 2(n - 20)$, потому что первые двадцать стоят $50 + 30$. При $n=0$ стоимость равна нулю.

Контракт: целое число деталей $0 \dots 50$, результат — целое число жетонов; вне диапазона возвращаем -1 как признак ошибки. Такой маркер безопасен здесь, потому что правильная цена неотрицательна. Называйте единицы: детали на входе, жетоны на выходе.

На стыке нужно проверить и сумму, и разность соседних цен. Цена(10)=50, цена(11)=53: прирост равен новой ставке 3. Если умножить весь заказ на 3, накопленная первая ступень пропадёт. Проверка монотонности полезна, но без точных ответов не выявит все ошибки.

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 23 деталей: $10 \cdot 5 + 10 \cdot 3 + 3 \cdot 2 = 86$. Та же формула в третьей ветви: $80 + 2 \cdot (23 - 20) = 86$. Для 20 третья ставка ещё не используется: 80.

```
def cost(n):
    """Целое n; 0..50 деталей, иначе -1."""
    if n < 0 or n > 50:
        return -1
    if n <= 10:
        return 5 * n
    elif n <= 20:
        return 50 + 3 * (n - 10)
    else:
        return 80 + 2 * (n - 20)

for n in [0, 10, 11, 20, 21, 23, 50, 51]:
    print(n, cost(n))
```



Первые 10 деталей	$10 \times 5 = 50$ жетонов
Следующие 10	$10 \times 3 = 30$ жетонов
Ещё 3 детали	$3 \times 2 = 6$ жетонов
Всего 23 детали	$50 + 30 + 6 = 86$ жетонов

Разложение учебного заказа из 23 деталей на три оплачиваемые части; единица цены — жетон.

03. Практика 1. Проверяем материал

Стоимость и приращение на стыке

14–25 минут. Проверьте таблицу независимо: рисуйте группы по десять деталей, затем запускайте код.

n, деталей	Цена, жетонов
0; 9; 10; 11	0; 45; 50; 53
19; 20; 21	77; 80; 82
49; 50; 51	138; 140; –1
–1	–1

2 минуты — расчёт; 4 — измените список запусков под таблицу; 3 — ошибочно замените 50 на 30 во второй ветви и найдите контрпример; 2 — восстановите и объясните стыки. Продукт: журнал цен и приросты 10→11, 20→21. Критерии: единицы, 11 ответов, два прироста и исправленная формула.

04. Практика 2. Перенос способа

Единая ставка на весь заказ

25–33 минуты. У другой учебной мастерской за весь заказ 0...50 деталей берут по 5 жетонов при $n \leq 10$, иначе по 4. Напишите `flat_cost`. За 2 минуты сравните цены 10 и 11, за 4 реализуйте, за 2 проверьте 0, 10, 11, 20, 50, 51 и объясните уменьшение на стыке. Критерии: расчёт всего заказа, граница, защита, объяснение.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты подготовьте аргумент, почему один и тот же порог 10 не задаёт сам тариф. За 3 сравните оба правила на 11 и 20; за 2 запишите, какую фразу нужно уточнить у заказчика до написания кода.

06. Индивидуальная проверка

40–45 минут. Найдите `cost(24)`, `flat_cost(24)`, разность и объясните происхождение постоянного 80 в накопительном тарифе. Четыре пункта — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

Приоритеты 3 и 2. С ошибочной базой цена(11)=33 вместо 53; первый нарушенный вход во второй ветви —11. Все правильные ответы даны в таблице; для нового списка тестов функция cost остаётся без изменений.

Практика 2: разбор и критерии

Ответы 0,50,44,80,200,—1. Здесь 11 деталей дешевле 10, потому что меняется ставка всего заказа. Это свойство заданного вымышленного правила, а не ошибка ветвления.

```
def flat_cost(n):
    if n < 0 or n > 50:
        return -1
    if n <= 10:
        return 5 * n
    return 4 * n

for n in [0, 10, 11, 20, 50, 51]:
    print(n, flat_cost(n))
```

Обсуждение: возможный вывод

На 11 результаты 53 и 44, на 20 оба 80. Поэтому совпадение одного примера не доказывает тождество правил. Нужно уточнить: новая цена применяется ко всему объёму или только к части сверх порога.

Индивидуальная проверка: ответ и критерии

88;96;разность flat_cost—cost=8 жетонов. 80 — цена первых 20 деталей:10·5+10·3. Оба результата в жетонах.

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/else, 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	Реализовать накопительный тариф, проверить стыки диапазонов и отличить его от цены за весь заказ.	Исходник, таблица прогнозов и фактических результатов
Составлять и отлаживать программы; использовать условные операторы	Единая ставка на весь заказ	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.