



Урок 11. Python: Логические функции и контракт

Часть 1. Материалы ученика

Программирование на Python и C++ · 9 класс

Урок 11 из 34 · 45 минут · Модуль «Функции»

Уровень: начинающий программист; Python 3.

Программа: <https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

Проверка условия должна возвращать ответ, который можно использовать дальше. Печать True на экране и возврат True из функции — разные действия.

Результат урока: Написать логическую функцию по контракту, проверить включённые границы и применить её как фильтр без изменения входа.

01. Проблемная ситуация

0-5 минут. Правило учебной выборки: чётные числа от -2 до 6 включительно. Для -2,-1,0,6,8 запишите да/нет. Что потеряется, если заменить $\leq$ на $<$?

02. Понятия и разобранный пример

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

Предикат возвращает bool

Предикат — функция, результат которой True или False. Контракт задаёт допустимые входы, смысл результата и побочные эффекты. Здесь вход — целое число от -1000 до 1000; `accepted(x)` возвращает истину ровно для чётных x в замкнутом диапазоне $-2 \dots 6$, ничего не печатает и не меняет.

`return` передаёт результат вызывающему коду и завершает текущий вызов. `print` только выводит текст; функция без `return` возвращает `None`. Поэтому проверку удобно отделять от вывода: один результат применяется в `if`, тесте и фильтре.

`and` требует выполнения обеих частей. Сначала проверяется диапазон, затем чётность. В Python цепочка $-2 \leq x \leq 6$ обозначает две границы. Для отрицательных чётных и нуля $x \% 2 == 0$ остаётся истинным.

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

`accepted(-2)=True, accepted(-1)=False, accepted(0)=True, accepted(6)=True, accepted(8)=False`. В списке `[-2,-1,0,6,8,6]` сохраняем `[-2,0,6,6]`; повтор 6 — отдельная запись.

```
def accepted(x):  
    return -2 <= x <= 6 and x % 2 == 0  
  
data = [-2,-1,0,6,8,6]  
for x in data:  
    print(x, accepted(x))  
selected = [x for x in data if accepted(x)]  
print(selected)  
print(data)
```

Вход x	-2 -1 0 6 8
В диапазоне	да да да да нет
Чётное	да нет да да да
accepted	True False True True False

Логическое И принимает значение только при выполнении обоих требований.



03. Практика 1. Проверяем материал

Границы проверяются отдельно

14–25 минут. Сначала заполните таблицу истинности по двум частям условия, затем проверьте функцию и фильтр.

Вход	Ожидаемый результат
-3; -2; -1	False; True; False
0; 5; 6; 7	True; False; True; False
[] как список для фильтра	[]
[-2,6,6]	[-2,6,6]

2 минуты — прогноз; 4 — запуски; 3 — подмена return на print и объяснение None; 2 — восстановление. Форма: x / границы / чётность / результат. Критерии: семь чисел, обе границы, отсутствие изменения списка, различие печати и возврата.

04. Практика 2. Перенос способа

Соединяем два условия

25–33 минуты. Новый контракт: `accepted_pair(x,y)` принимает пару, если оба числа проходят `accepted` и их сумма не больше 6. За 2 минуты рассчитайте `(-2,6),(4,4),(0,6),(2,3)`; за 4 напишите функцию и тесты; за 2 объясните отклонение каждой неподходящей пары. Критерии: переиспользование функции, И трёх условий, четыре верных результата.

05. Обсуждение и границы вывода

33–40 минут. Сравните «функция сразу печатает ответ» и «функция возвращает bool». За 2 минуты перечислите потребителей результата, за 3 обсудите тестирование, за 2 обоснуйте выбор для фильтра.

06. Индивидуальная проверка

40–45 минут. Для `x=6` укажите результаты исходного предиката, варианта со строгими границами, функции с `print` без `return` и `accepted_pair(6,0)`. Четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

При `print` вместо `return` функция печатает логическое значение, но возвращает `None`. Фильтр с `if accepted(x)` отбросит все элементы, потому что `None` ложно. Пустой список фильтруется в пустой; повторы и порядок сохраняются.

Практика 2: разбор и критерии

Ожидаются `True`, `False`, `True`, `False`. (4,4) отклоняется по сумме 8; (2,3) — по нечётному `y`. Самостоятельная полная программа:

```
def accepted(x):  
    return -2 <= x <= 6 and x % 2 == 0  
  
def accepted_pair(x, y):  
    return accepted(x) and accepted(y) and x+y <= 6  
  
for pair in [(-2,6),(4,4),(0,6),(2,3)]:  
    print(pair, accepted_pair(*pair))
```

Обсуждение: возможный вывод

Возвращённый `bool` можно использовать в фильтре, условии и `assert` без разбора текста. Вывод остаётся отдельным действием. Печать полезна в интерфейсе и отладке, но не заменяет возвращаемое значение.

Индивидуальная проверка: ответ и критерии

`True`; `False`; `None` как возвращаемое значение (при этом печатается `True`); `True`. По одному баллу за ответ.

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, 9 класс. Предпосылки: сравнения, `and`, остаток от деления, списки. 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
Функции. Логические функции. Рекурсивные функции.	Границы проверяются отдельно	Написать логическую функцию по контракту, проверить включённые границы и применить её как фильтр без изменения входа.
Структурирование решения и отладка	Соединяем два условия	Исходник, журнал обычных и граничных тестов; индивидуальный ответ

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-9-klass-programma>

Python: функции и 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.