IT CAREER HUB

Python

Цикл while

Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)

Важно

- Камера должна быть включена на протяжении всего занятия
- В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы
- Вести себя уважительно и этично по отношению к остальным участникам занятия
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение

- [x] Встроенные математические функции
- [x] Модуль
- [x] Оператор import
- [x] Стандартная библиотека Python
- [x] Модуль math
- [x] Хранение чисел в памяти
- [x] Точность операций с вещественными числами
- [x] Модуль Decimal

План занятия

- Цикл
- Оператор break
- Оператор continue
- Бесконечный цикл
- Конструкция while/else
- Модуль random

ICH IT CAREER HUB

ОСНОВНОЙ БЛОК

IT CAREER HUB

Цикл

Полезно знать

ICH
IT CAREER HUB

Цикл

Это конструкция в программировании, которая позволяет повторять выполнение блока кода (тела цикла) несколько раз, пока выполняется определённое условие или пока не пройдётся по всем элементам последовательности.

Полезно знать

Итерация

Это одно выполнение тела цикла. Итерации происходят до тех пор, пока выполняется условие цикла.

Полезно знать

Цикл while

Это цикл с предусловием, который выполняет блок кода до тех пор, пока условие остаётся истинным.

Синтаксис цикла while

```
while условие: # условие цикла

    # блок кода или тело цикла
```

Синтаксис цикла while

Условие: Это выражение, которое проверяется перед каждой итерацией.
Если оно истинно (`True`), выполняется блок кода. Как только условие становится ложным (`False`), цикл завершает свою работу.

Блок кода:
Код, который будет выполняться на каждой итерации, пока условие истинно.
После выполнения блока кода интерпретатор всегда возвращается к проверке условия.

Пример

```
i = 1

while i <= 5:

    print(i)

    i += 1
```

Объяснение

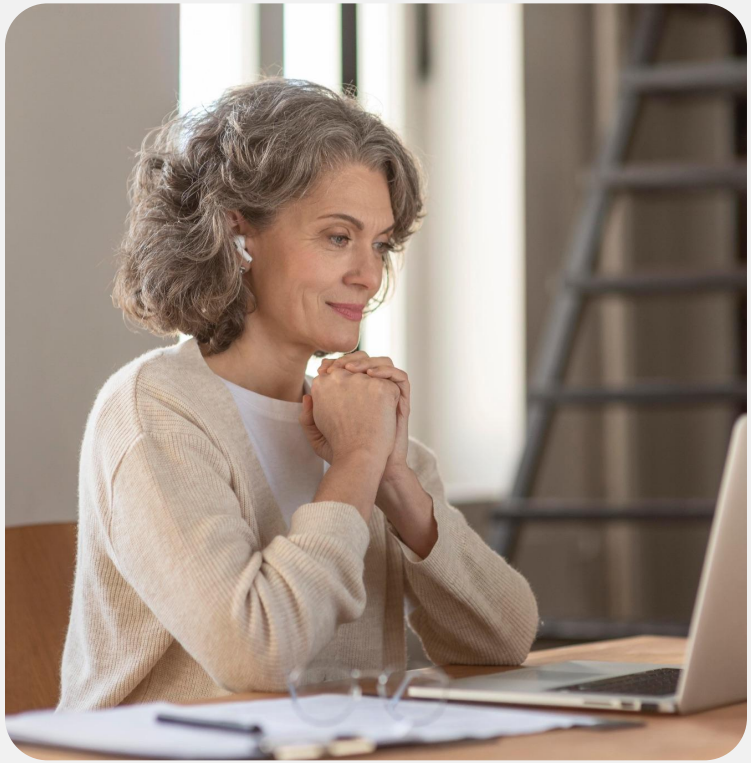
- Переменная `i` инициализируется значением 1
- Цикл продолжается до тех пор,
пока значение переменной `i` меньше или равно 5
- Внутри цикла переменная увеличивается на 1 с каждой итерацией с помощью оператора `i += 1`
- Как только `i` становится больше 5, цикл завершается

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

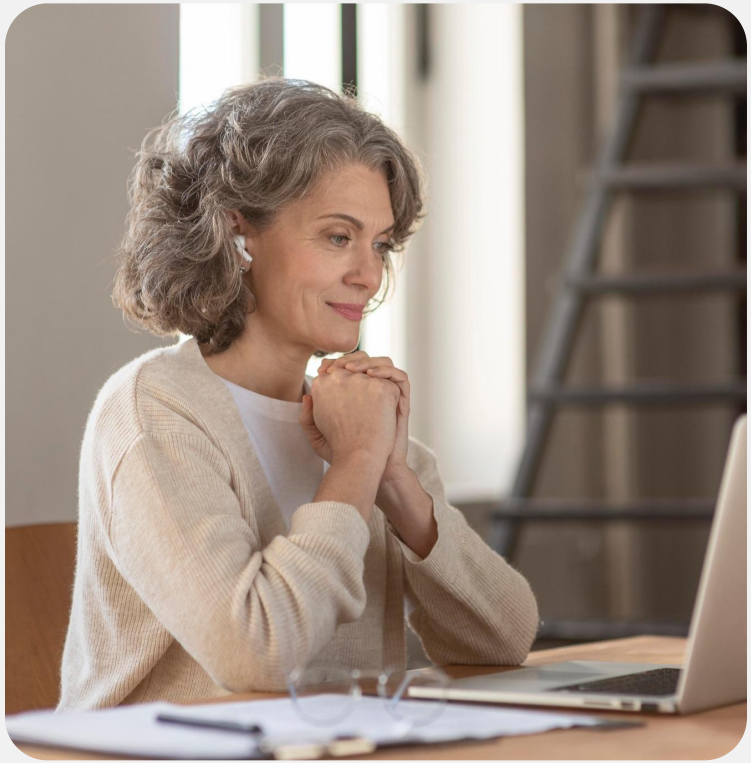


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1

while i <= 3:

    print(i)
```

a. Код выведет числа от 1 до 3
b. Код выведет числа от 1 до 4
c. Код выведет 1 один раз
d. Код зациклится и будет выводить числа бесконечно

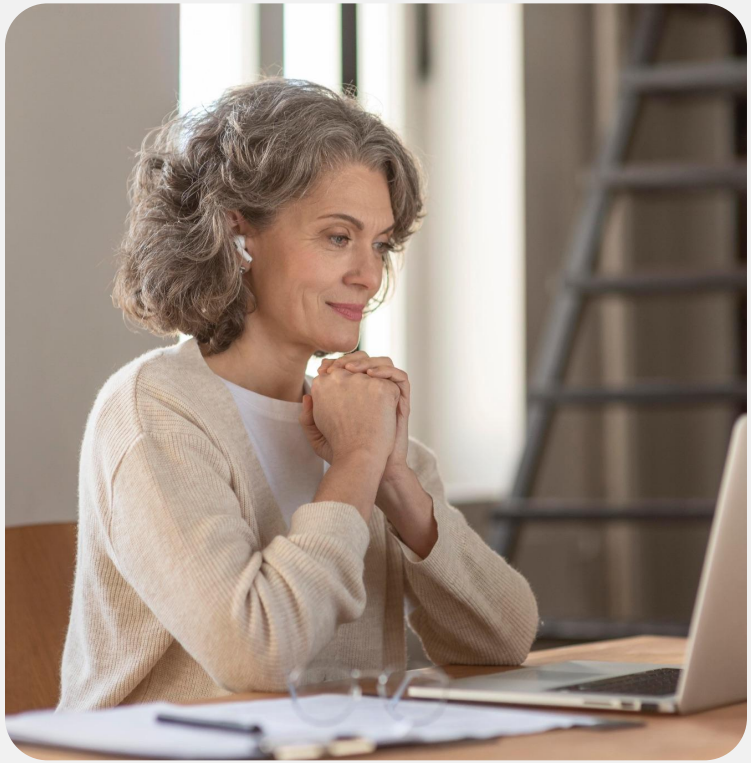


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1

while i <= 3:

    print(i)
```

a. Код выведет числа от 1 до 3
b. Код выведет числа от 1 до 4
c. Код выведет 1 один раз
d. Код зациклится и будет выводить числа бесконечно

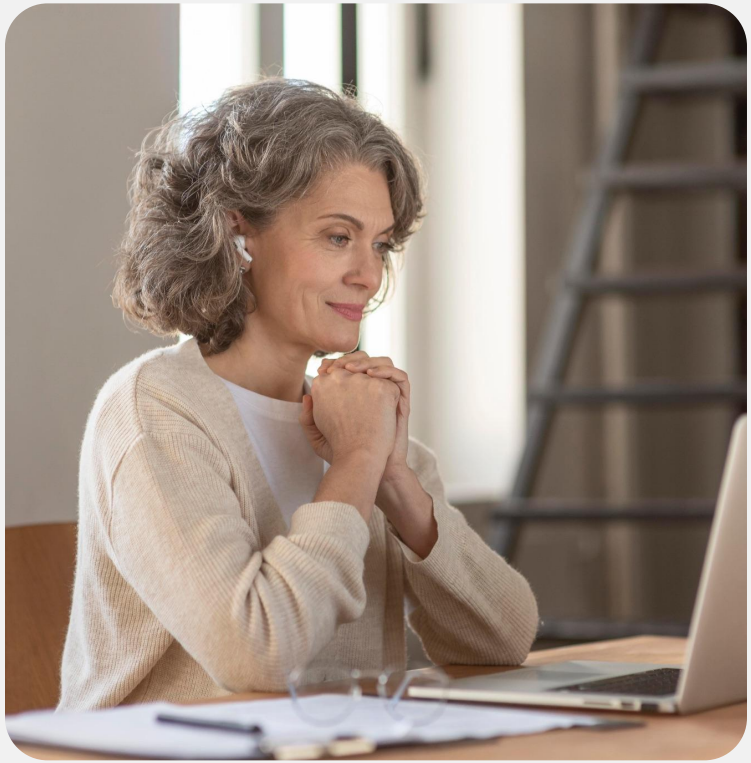


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
i = 1

while i <= 3:

    print(i)

    i += 1
```

a. Код выведет числа от 1 до 2
b. Код выведет числа от 1 до 3
c. Код выведет 1 один раз
d. Код зациклится и будет выводить числа бесконечно

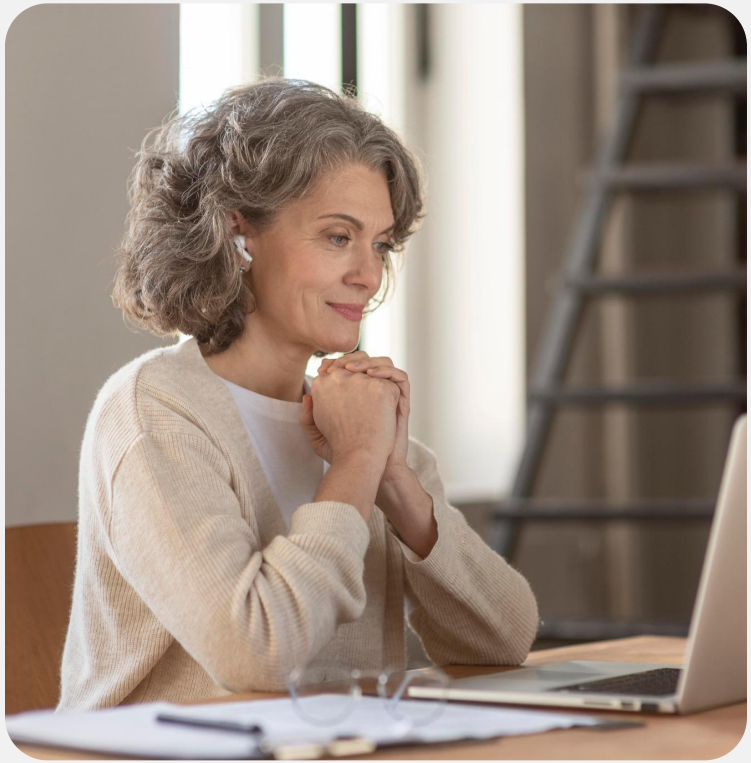


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
i = 1

while i <= 3:

    print(i)

    i += 1
```

a. Код выведет числа от 1 до 2
b. Код выведет числа от 1 до 3
c. Код выведет 1 один раз
d. Код зациклится и будет выводить числа бесконечно

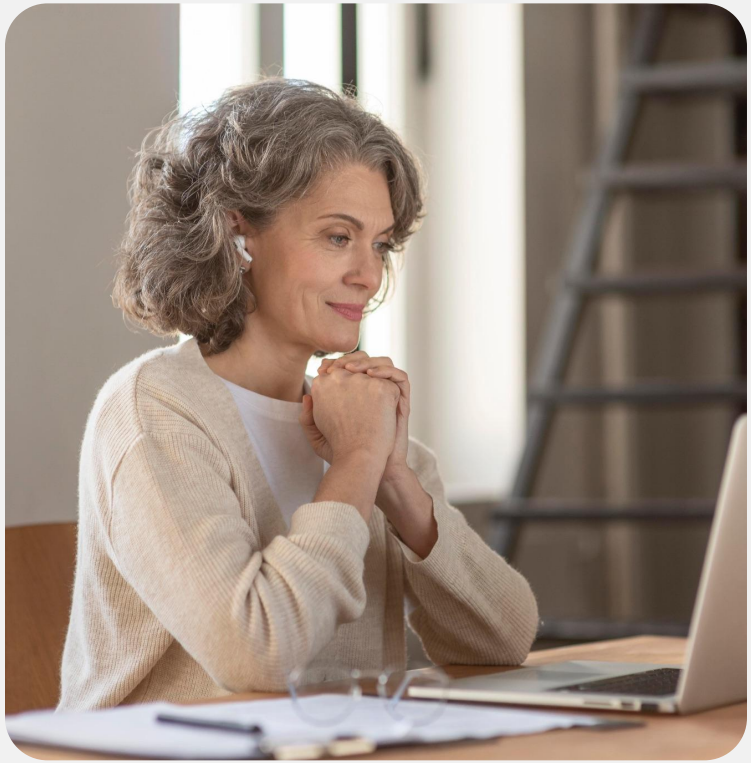


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
i = 10
while i > 7:
    i -= 1
print(i)
```

a. Число 8
b. Число 7
c. Числа от 10 до 8
d. Код выведет ошибку

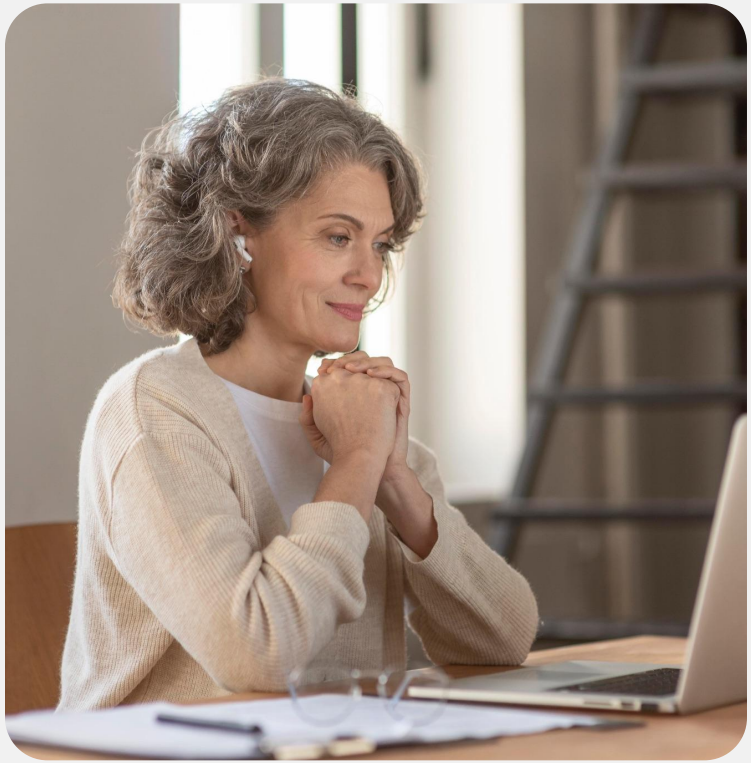


Выберите правильные варианты ответа

Какой результат выведет следующий код?

```
i = 10
while i > 7:
    i -= 1
print(i)
```

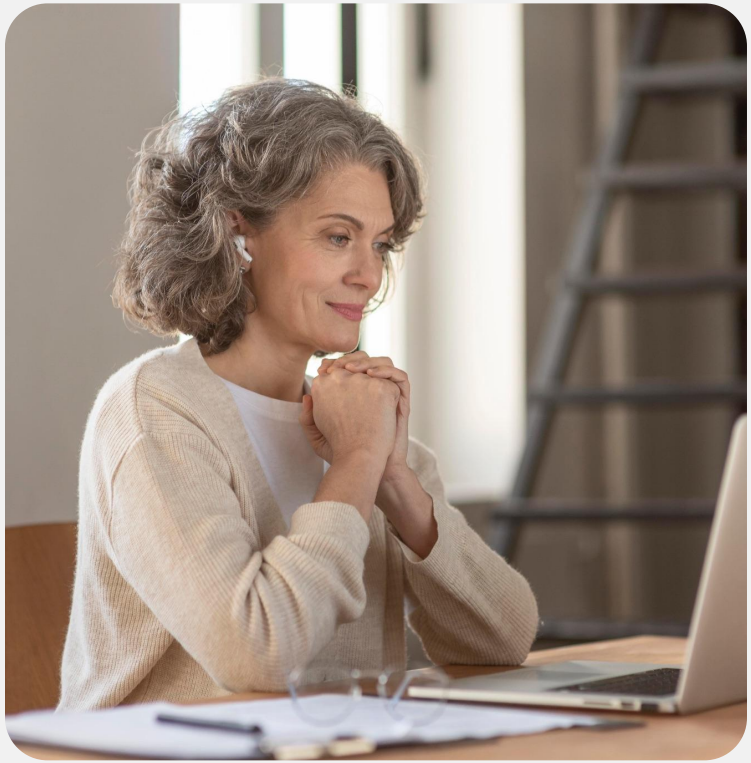
a. Число 8
b. Число 7
c. Числа от 10 до 8
d. Код выведет ошибку



Выберите правильные варианты ответа

Найди ошибку в коде

```
i = 0

while i < 5:

print(i)

i += 1
```

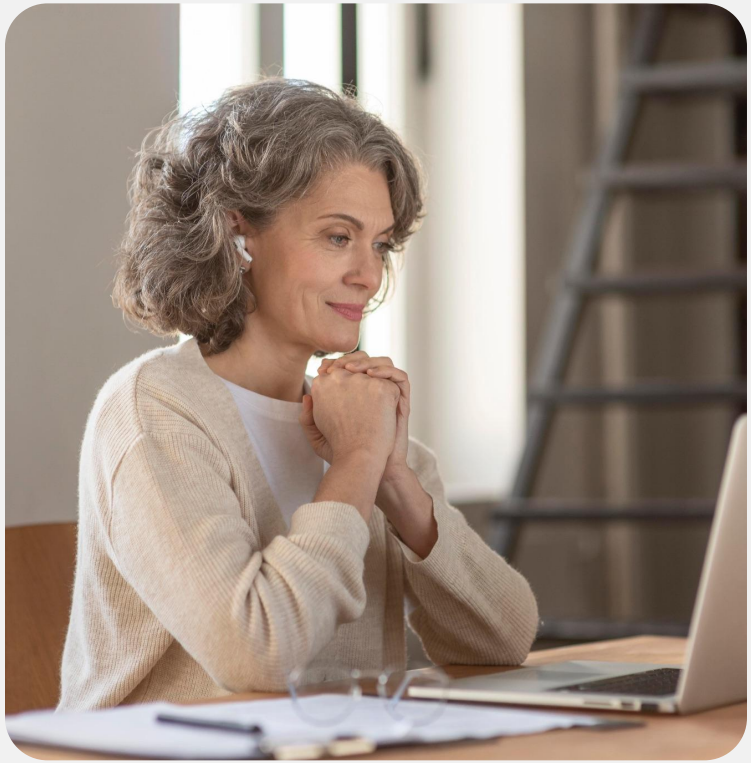


Выберите правильные варианты ответа

Ошибка: Внутри цикла блок кода должен быть с отступом.

Исправленный код:

```
i = 0
while i < 5:
    print(i)
    i += 1
```

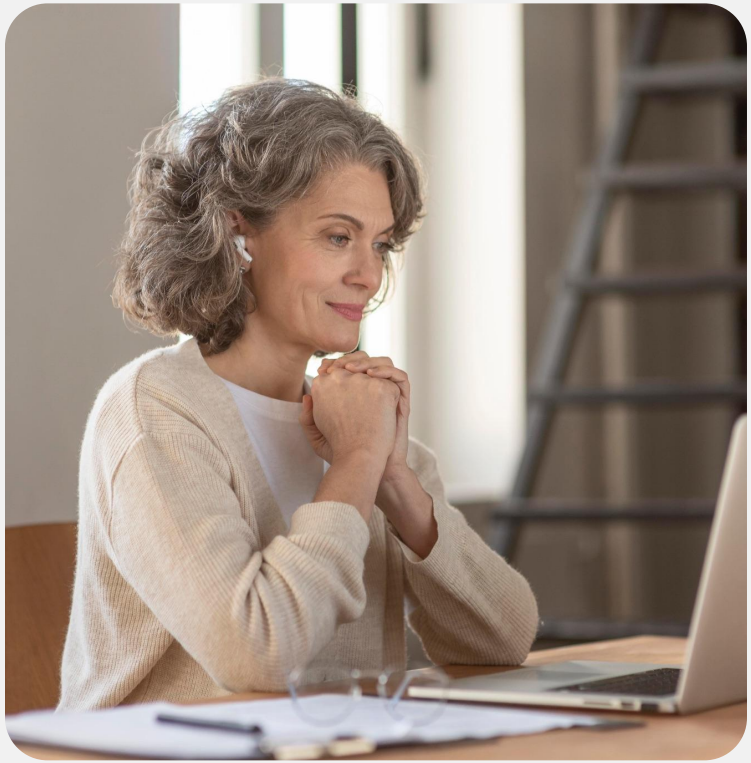


Выберите правильные варианты ответа

Какое значение примет переменная i после выполнения следующего цикла?

```
i = 2
while i <= 8:
    i += 2
```

a. 6
b. 8
c. 10
d. 12



Выберите правильные варианты ответа

Какое значение примет переменная i после выполнения следующего цикла?

```
i = 2
while i <= 8:
    i += 2
```

a. 6
b. 8
c. 10
d. 12

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Оператор break

Полезно знать

IT CAREER HUB

Оператор break

Используется для принудительного завершения выполнения цикла.

Пример

```
i = 1
while i <= 10:
    print(i)
    if i == 5:
        break  # Прерывание цикла, когда i станет равно 5
    i += 1
```

Полезно знать

IT CAREER HUB

Использование break с пользовательским вводом

Оператор `break` может быть полезен для выхода из цикла, когда программа ожидает ввода от пользователя и требуется завершить цикл при определённых условиях.

Пример

```
while True:
    user_input = input("Введите 'exit', чтобы завершить цикл: ")
    if user_input == "exit":
        break
    print("Вы ввели:", user_input)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Оператор continue

Полезно знать

Оператор continue

В циклах используется для пропуска оставшейся части кода текущей итерации и перехода к следующей итерации цикла.

Синтаксис

```
while условие:
    # код до continue
    if условие_для_continue:
        continue  # пропуск оставшегося кода в этой итерации
    # код, который будет пропущен, если сработает continue
```

Пример использования continue

```
i = 0
while i < 5:
    i += 1
    if i % 2 == 0::
        continue  # Пропускаем итерацию, когда i четное
    print(i)
```

Пример 2: Пропуск ввода при условии

```
result = 1
while True:
    user_input = input("Введите число для перемножения: ")
    if user_input == "0":
        print("Пропуск итерации")
        continue  # Пропускаем оставшуюся часть текущей итерации
    if user_input == "exit":
        print("Выход из программы")
        break  # Прерывание цикла
    result *= int(user_input)
    print("Результат перемножения:", result)
```

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Бесконечный цикл

Полезно знать

Бесконечный цикл

Это цикл, который никогда не завершает своё выполнение, потому что его условие всегда остаётся истинным.

Пример

```
while True:

    print("Этот цикл будет выполняться бесконечно")
```

Пример: использование break

```
while True:
    user_input = input("Введите 'stop', чтобы завершить цикл: ")
    if user_input == "stop":
        print("Цикл завершён.")
        break  # Прерывание цикла, если пользователь ввёл 'stop'
```

Полезно знать

ICH
IT CAREER HUB

Бесконечный цикл

Может возникнуть случайно, если неправильно прописано условие завершения.

Примеры

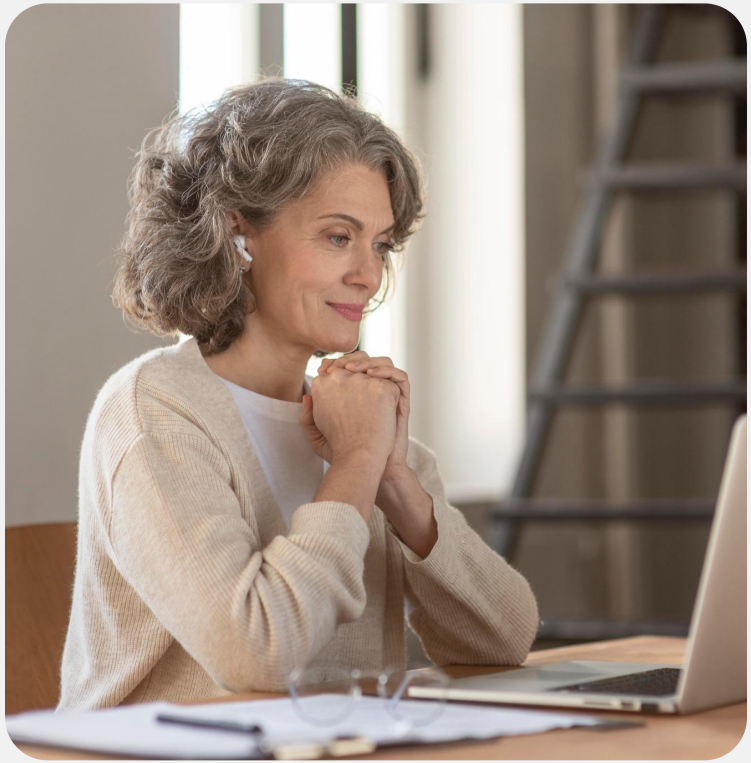
```python
i = 0
while i < 10:
    print(i)
    # Пропущено увеличение i, поэтому условие всегда истинно, и цикл бесконечен


i = 0
while i < 10:
    print(i)
    i -= 1
    # i меняется в другую сторону, не приближаясь к условию выхода
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

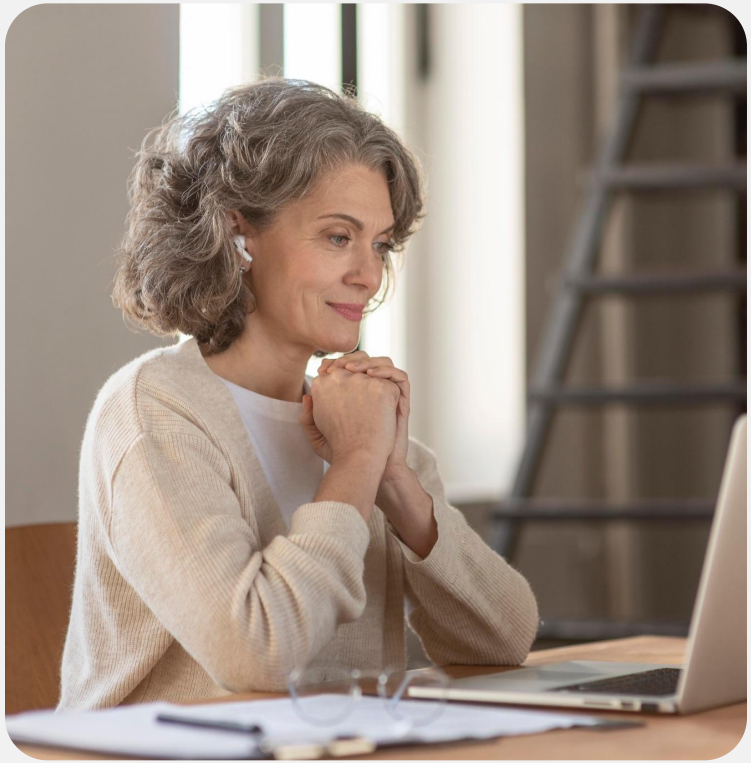


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 5:
    if i == 3:
        break
    print(i)
    i += 1
```

a. 1, 2
b. 1, 2, 3
c. 1, 2, 3, 4, 5
d. Код зациклится

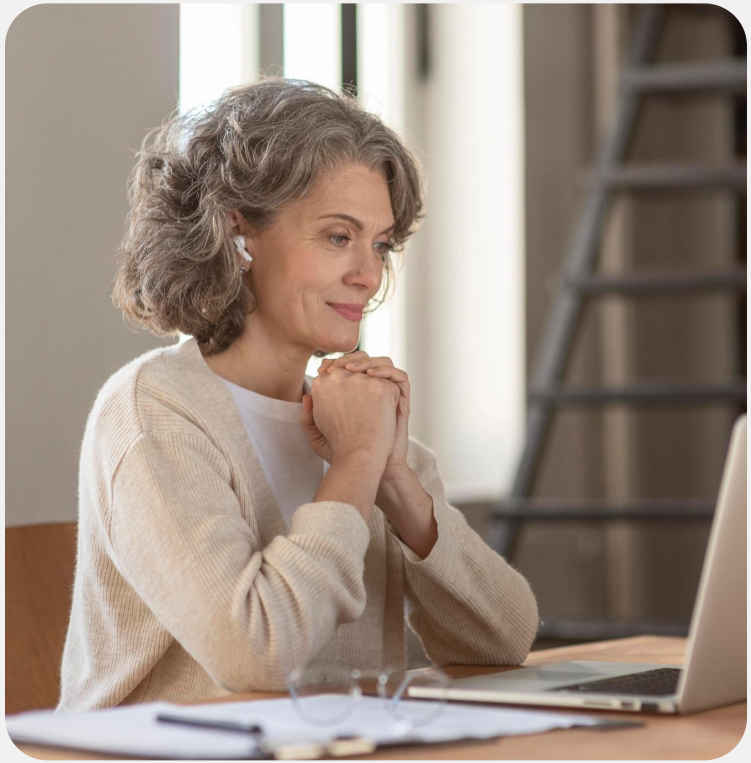


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 5:
    if i == 3:
        break
    print(i)
    i += 1
```

a. 1, 2
b. 1, 2, 3
c. 1, 2, 3, 4, 5
d. Код зациклится

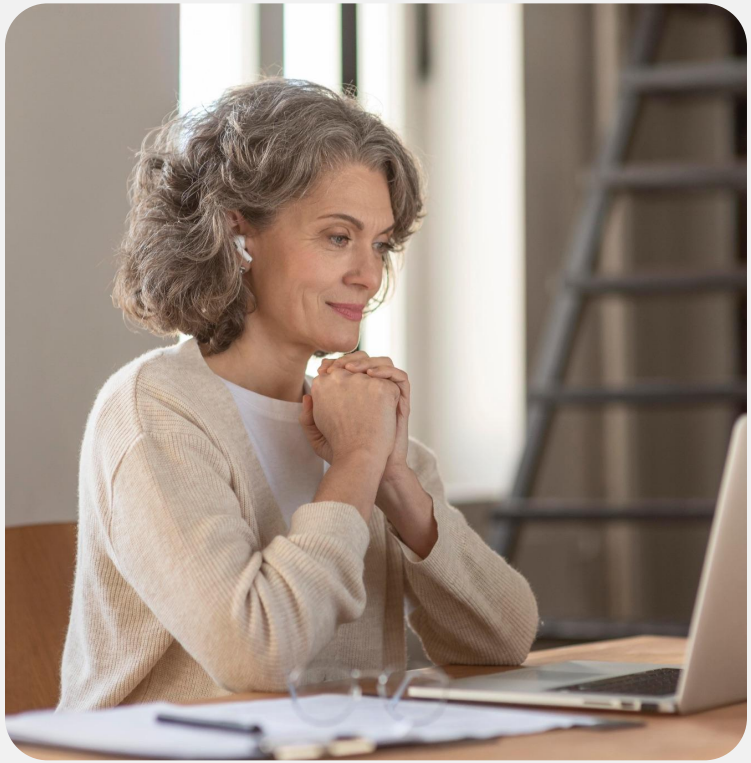


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 0
while i < 5:
    i += 1
    if i == 3:
        continue
    print(i)
```

a. 1, 2, 4, 5
b. 1, 2, 3, 4, 5
c. 0, 1, 2, 4, 5
d. Код зациклится

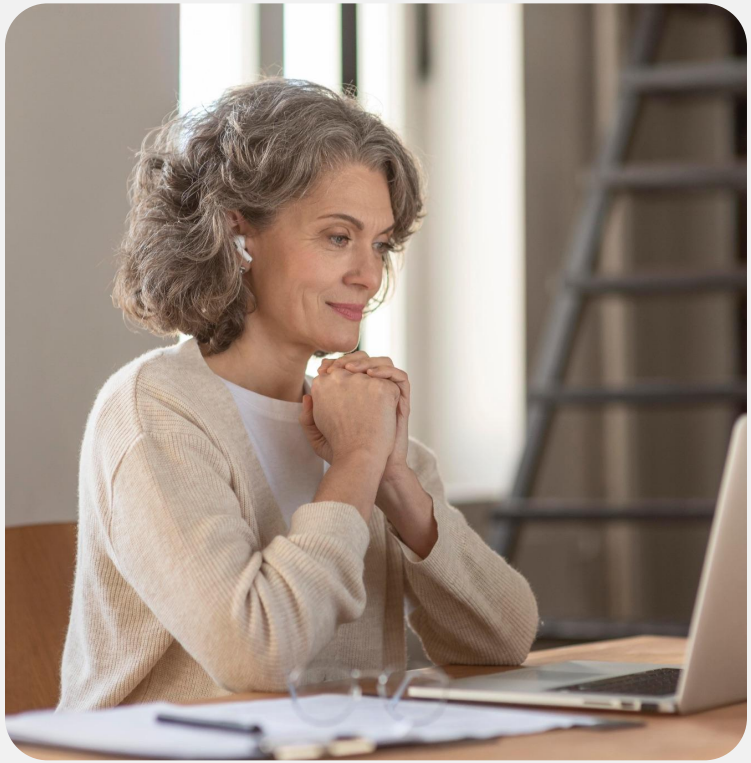


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 0
while i < 5:
    i += 1
    if i == 3:
        continue
    print(i)
```

a. 1, 2, 4, 5
b. 1, 2, 3, 4, 5
c. 0, 1, 2, 4, 5
d. Код зациклится

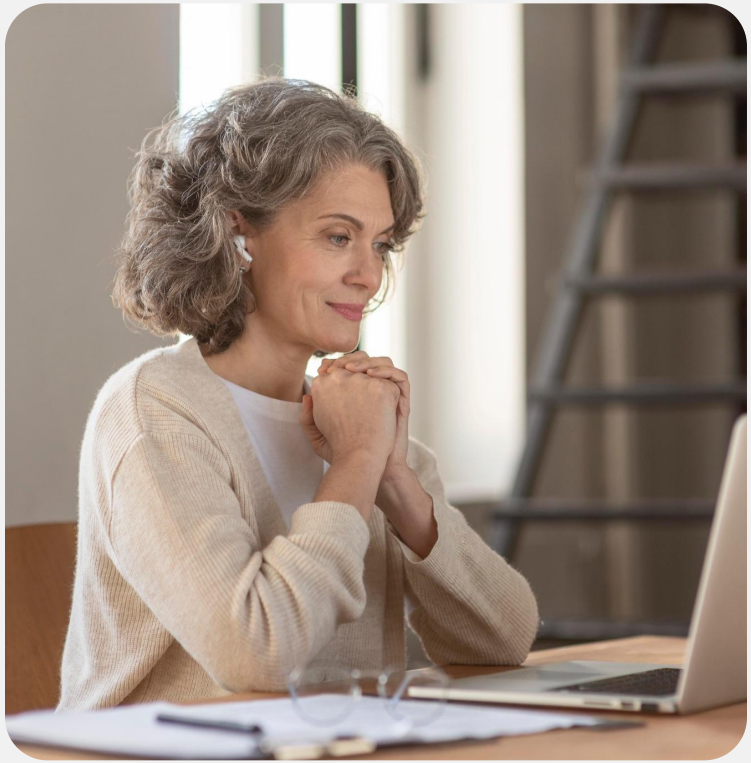


Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 5:
    if i % 2 == 0:
        i += 1
        continue
    print(i)
    i += 1
```

a. 1, 2, 3, 4
b. 1, 3, 5
c. 1, 3
d. 2, 4



Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 5:
    if i % 2 == 0:
        i += 1
        continue
    print(i)
    i += 1
```

a. 1, 2, 3, 4
b. 1, 3, 5
c. 1, 3
d. 2, 4

IT CAREER HUB

ВОПРОСЫ

ICH IT CAREER HUB

Конструкция while/else

Полезно знать

ICH
IT CAREER HUB

Конструкция while/else

Позволяет выполнить блок кода `else` после завершения цикла `while`, при нормальном завершении цикла, то есть без использования оператора `break`.

Синтаксис

```
while условие:
    # блок кода while
else:
    # блок кода else
```

Пример 1

```
i = 1

while i <= 5:

    print(i)

    i += 1

else:

    print("Цикл завершён без прерываний.")
```

Пример 2

```
result = 1
num = 1
while num < 6:
    user_input = input("Введите " + str(num) + "-е число для перемножения: ")
    num += 1
    if user_input == "0":
        print("Некорректные данные. Выход из программы")
        break  # Прерывание цикла
    result *= int(user_input)
else:
    print("Цикл завершён без прерываний.")
    print("Результат перемножения:", result)
```

Полезно знать

continue в while/else

Оператор `continue` не прерывает цикл, а только пропускает часть итерации, поэтому блок `else` всё равно выполнится.

Пример

```
result = 1
num = 1
while num < 6:
    user_input = input("Введите число №" + str(num) + " для перемножения: ")
    num += 1
    if user_input == "0":
        print("Некорректные данные. Выход из программы")/
        break # Прерывание цикла
    if int(user_input) > 1000:
        print("Слишком большое число. Пропуск итерации")
        continue # Пропускаем оставшуюся часть текущей итерации
    result *= int(user_input)
else:
    print("Цикл завершён без прерываний.")
print("Результат перемножения:", result)
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ЗАДАНИЕ

Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 3:
    if i == 2:
        i += 1
        continue
    print(i)
    i += 1
else:
    print("Цикл завершён.")
```

a. 1, 3
b. 1, 3, "Цикл завершён."
c. "Цикл завершён."
d. 1, 2, 3

Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 3:
    if i == 2:
        i += 1
        continue
    print(i)
    i += 1
else:
    print("Цикл завершён.")
```

a. 1, 3
b. 1, 3, "Цикл завершён."
c. "Цикл завершён."
d. 1, 2, 3

Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 4:
    if i == 3:
        break
    print(i)
    i += 1
else:
    print("Цикл завершён.")
```

a. 1, 2, "Цикл завершён."
b. 1, 2
c. 1, 2, 3
d. Цикл зациклится

Выберите правильный вариант ответа

Какой результат выведет следующий код?

```
i = 1
while i <= 4:
    if i == 3:
        break
    print(i)
    i += 1
else:
    print("Цикл завершён.")
```

a. 1, 2, "Цикл завершён."
b. 1, 2
c. 1, 2, 3
d. Цикл зациклится

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

Модуль random

Полезно знать

Модуль random

Это стандартный модуль Python, который предоставляет функции для генерации случайных чисел, а также для работы со случайным выбором элементов из коллекций.

Генерация случайного числа с плавающей запятой

```
import random


# Генерируем случайное число от 0.0 до 1.0
print(random.random())
```

Генерация случайного целого числа

```python
import random


# Генерируем случайное целое число от 1 до 10 (включая обе границы)
print(random.randint(1, 10))
```

Генерация случайного целого числа с шагом

```
import random

# Генерируем случайное целое число от 1 до 10 (не включая число 10)
print(random.randrange(1, 10))
# Генерируем случайное целое число от 1 до 10 (не включая число 10) и с шагом 2 (только по
# нечетным числам)
print(random.randrange(1, 10, 2))
```

Когда использовать random

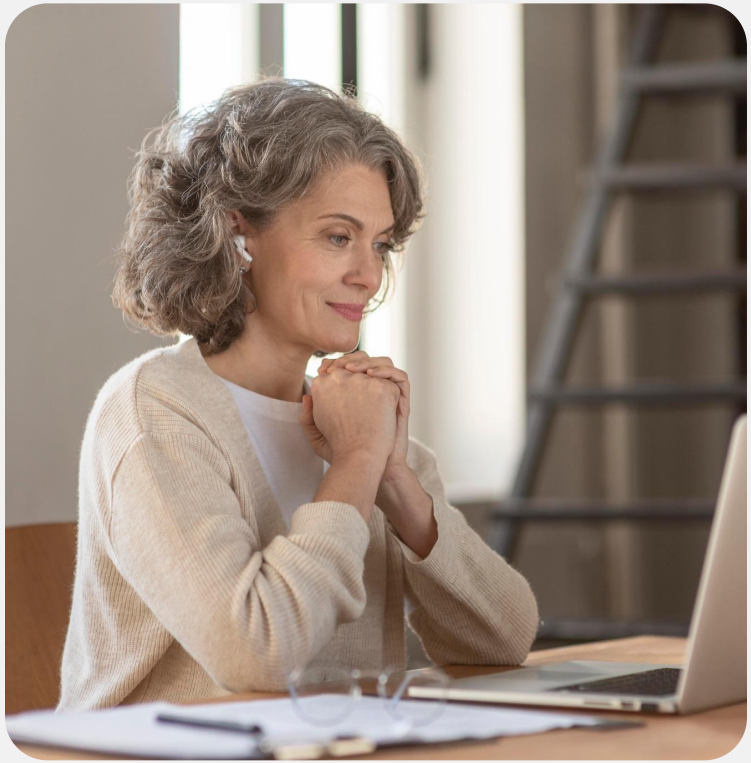
- **Симуляции:** Игры, моделирование случайных событий.
- **Генерация тестовых данных:** Создание случайных чисел или строк для тестирования алгоритмов.
- **Программы, требующие случайности:** Лотереи, случайное распределение задач.
- **Перемешивание данных:** Например, случайное перемешивание списка студентов для презентации или задания.

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

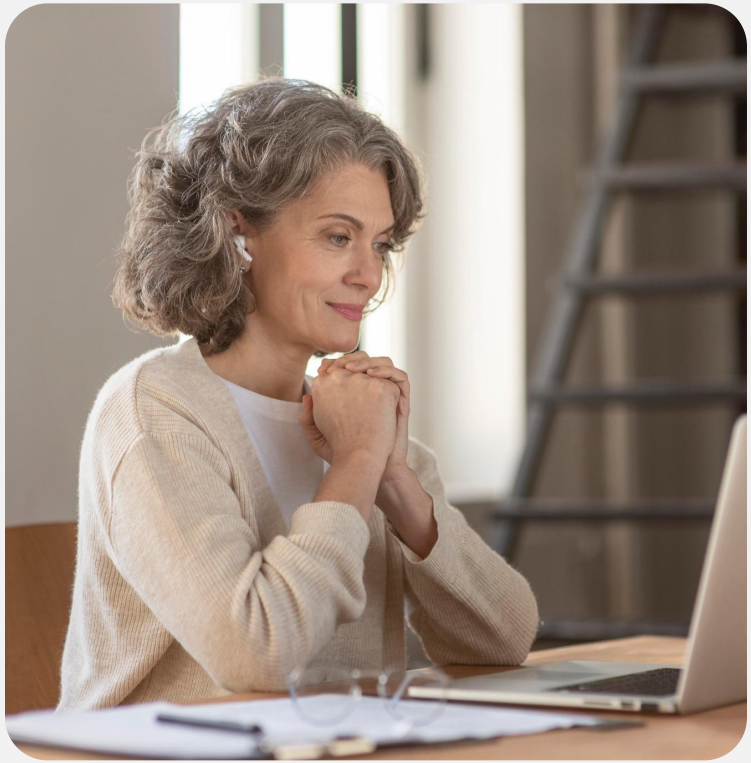
ЗАДАНИЕ



Выберите правильный вариант ответа

Найдите ошибку в коде:

```
import random

number = random.randint(1, 5)
if number == 6:
    print("Вы выиграли!")
```



Выберите правильный вариант ответа

Найдите ошибку в коде:

```
import random

number = random.randint(1, 5)
if number == 6:
    print("Вы выиграли!")
```

Ошибка заключается в том, что `random.randint(1, 5)` генерирует число от 1 до 5 включительно, поэтому проверка на `number == 6` никогда не выполнится.

IT CAREER HUB

Практическая работа

1. Обратный отсчёт

Напишите программу, которая выводит обратный отсчёт от введённого числа до 1 и в конце "Отсчёт завершён!".

Пример ввода:

```
Введите начальное число: 5
```

Пример вывода:

```
5
4
3
2
1
Отсчёт завершён!
```

2. Сумма положительных чисел

Напишите программу, которая запрашивает у пользователя числа и суммирует только положительные. Программа завершится и выведет общую сумму, когда пользователь введёт "0".

Пример вывода:

```
Введите число: 5
Введите число: 10
Введите число: -1
Введите число: 3
Введите число: 0
Общая сумма положительных: 18
```

IT CAREER HUB

ВОПРОСЫ

IT CAREER HUB

ДОМАШНЕЕ ЗАДАНИЕ

Домашнее задание

1. **Сумма цифр числа**

Напишите программу, которая вычисляет сумму цифр введённого числа.

Пример вывода:

```
Введите число: 12345
Сумма цифр: 15
```

Домашнее задание

2. Палиндром

Напишите программу, которая запрашивает у пользователя целое число и определяет, является ли оно палиндромом. Число является палиндромом, если оно читается одинаково слева направо и справа налево.

Пример вывода:

```
Введите число: 12321
Число 12321 является палиндромом.

Введите число: 1234
Число 1234 не является палиндромом.
```

Домашнее задание

3. Проверь интуицию

Напишите программу, которая генерирует случайное число от 1 до 100 включительно и дает пользователю 10 попыток, чтобы угадать это число. Программа должна подсказывать, больше или меньше загаданное число. После завершения игры программа оценивает, насколько хорошая интуиция у игрока.

Домашнее задание

3. Проверь интуицию

Пример вывода:

```
Угадайте число от 1 до 100. У вас 10 попыток.
Ваше предположение: 50
Загаданное число меньше вашего
Ваше предположение: 25
Загаданное число больше вашего
Ваше предположение: 30
Поздравляем! Вы угадали число: 30.

Вы угадали число за 3 попытки. Отличный результат!

--------------------------------------------------

Угадайте число от 1 до 100. У вас 10 попыток.
Ваше предположение: 2
Поздравляем! Вы угадали число: 2 за 1 попыток. Отличный результат!
Отличная интуиция! Вы угадали с первой попытки!
```

Заключение