

Занятие 3. Ветвление и циклы

План занятия

- Проверка домашнего задания
- Повторение теории
- Условный оператор if
- and и or
- Тернарный оператор
- Вложенные условия
- Циклы
 - for
 - while

1

Условный оператор if

if

Зачем?

- чтобы выполнять какие-то действия в зависимости от условий

```
: age = 19  
if age >= 18:  
    print(f"Этот человек взрослый!")
```

Этот человек взрослый!

if - elif - else - проверка нескольких условий

```
age = 5

if age == 0:
    print("Это младенец")
elif age < 6:
    print("Дошкольник")
elif age < 17:
    print("Школьник")
else:
    print("Это взрослый человек")
```

Дошкольник

Оператор and

Зачем?

- проверяет что выполняется несколько условий сразу

```
age = 18
profession = "programmer"

if age >= 18 and profession == "programmer":
    print("Программист старше 18")
```

Программист старше 18

Оператор or

Зачем?

- проверяет что выполняется хотя бы одно условие

```
: money = 15
products_in_freeze = ["салат", "борщ", "яйца"]

if money > 5 or products_in_freeze:
    print("Я сегодня поем!")
```

Я сегодня поем!

Комбинация and & or

Важно

- and выполняется в первую очередь
- лучше расставлять скобки для наглядности

```
money = 30
products_in_freeze = ["салат", "борщ", "яйца"]
have_free_time = True

if money > 15 or have_free_time and products_in_freeze:
    print("Я могу поесть дома, потому что у меня есть продукты и свободное время")
    print("Также я могу заказать еду, потому что у меня есть деньги")
```

Я могу поесть дома, потому что у меня есть продукты и свободное время
Также я могу заказать еду, потому что у меня есть деньги

Тернарный оператор

Зачем?

- чтобы проверить условие в одну строку

```
age = 20  
message = "Старше 18" if age >= 18 else "Младше 18"  
message
```

Вложенные условия

- Условия могут быть любой вложенности, то есть код под условием - обычный код, тут могут быть циклы, другие условия и тд

```
money = 30
products_in_freeze = ["салат", "борщ", "яйца"]
have_free_time = True

if have_free_time:
    if products_in_freeze:
        print("Я могу приготовить еду дома")
    elif money > 15:
        print("У меня есть свободное время и деньги. Я могу пойти покушать в заведении")
elif money > 15:
    print("У меня нет свободного времени, но я могу заказать еду")
```

Я могу приготовить еду дома

2

Циклы

Цикл for

Зачем?

- Чтобы повторить действие N раз

Пример:

```
for i in [1, 2, 3]:  
    print(i)
```

1
2
3

Цикл for

- Он также может содержать в себе любой код, в том числе другие циклы и условия

Пример:

```
for i in [1, 3, 5, 3, 6, 8]:  
    if not i % 2:  
        print(f"Число четное: {i}")
```

Число четное: 6

Число четное: 8

Цикл for в цикле for

- Он также может содержать в себе любой код, в том числе другие циклы и условия

Пример:

```
for i in [1,[1, 2, 3], 5]:  
    if isinstance(i, list):  
        for j in i:  
            print(f"Value from list: {j}")
```

```
Value from list: 1  
Value from list: 2  
Value from list: 3
```

Ключевое слово continue

- Не заканчивать текущий проход цикла, начать следующий

Пример:

```
: for i in [2, 3, 4, 0, 9, 8]:  
    if i == 0:  
        continue  
    print(f"100 / {i} = {100 / i}")
```

100 / 2 = 50.0

100 / 3 = 33.333333333333336

100 / 4 = 25.0

100 / 9 = 11.111111111111111

100 / 8 = 12.5

Ключевое слово break

- Прервать выполнение цикла

Пример:

```
for i in [2, 3, 4, 0, 9, 8]:  
    if i == 0:  
        print("Деление на 0 запрещено!")  
        break  
    print("А это сообщение уже не выведется")  
print(f"100 / {i} = {100 / i}")
```

100 / 2 = 50.0

100 / 3 = 33.33333333333333

100 / 4 = 25.0

Деление на 0 запрещено!

Ключевое слово else

- Способ узнать, был ли цикл прерван при помощи break

Пример:

```
for i in [2, 3, 4, 9, 8]:  
    if i == 0:  
        print("Деление на 0 запрещено!")  
        break  
    print(f"100 / {i} = {100 / i}")  
else:  
    print("Мы не прерывали выполнение при помощи break. Цикл завершился естественным способом")
```

100 / 2 = 50.0

100 / 3 = 33.333333333333336

100 / 4 = 25.0

100 / 9 = 11.111111111111111

100 / 8 = 12.5

Мы не прерывали выполнение при помощи break. Цикл завершился естественным способом

Цикл while

Зачем?

- Можно организовать бесконечный цикл
- Не нужно изначально знать, сколько итераций у нас будет

Важно: работает медленнее чем for

```
count = 5
i = 0
while i < count:
    print(f"Итерация {i}")
    i += 1
```

```
Итерация 0
Итерация 1
Итерация 2
Итерация 3
Итерация 4
```

Цикл while

Пример

```
# просим пользователя ввести число до тех пор пока он не догадается, что за число мы загадали  
x = 10  
  
while True:  
    value = int(input("Введите число: "))  
    if value == x:  
        print("Вы угадали!")  
        break  
    else:  
        print("Вы не угадали, попробуйте еще раз!")
```

```
Введите число: 3  
Вы не угадали, попробуйте еще раз!  
Введите число: 8  
Вы не угадали, попробуйте еще раз!  
Введите число: 10  
Вы угадали!
```

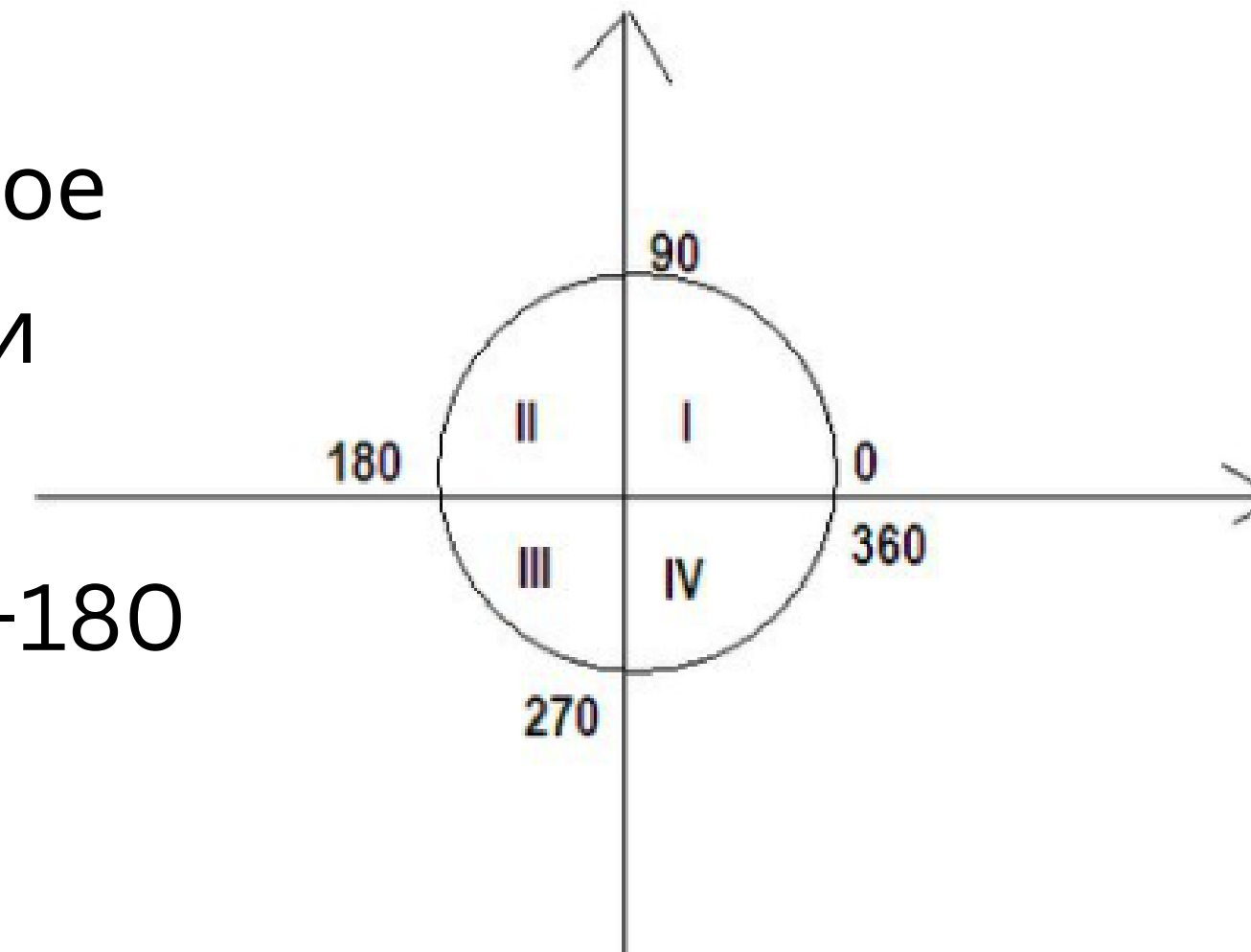
3

Домашнее задание

Домашнее задание (часть 1)

Часть 1. Работа с условиями:

1. Программа получает от пользователя число и проверяет, четное оно или нет. В зависимости от этого программа выводит на экран сообщение "четное" или "нечетное"
2. Программа получает от пользователя угол (целое число), нужно вывести на экран, какой четверти оно принадлежит, например "1 четверть",
напоминаю, что 0-90 градусов - 1 четверть, 90-180 - 2 четверть, 180 - 270 - 3 четверть, 270-360 - 4 четверть



Домашнее задание (часть 1)

Часть 1. Работа с условиями:

3. Фирма набирает сотрудников и ей нужны молодые кадры со стажем. Возраст от 20 лет, стаж от трёх. Приложение запрашивает у пользователя его возраст и стаж и говорит, подходит ли тот под требования фирмы.

4. Пользователь вводит номер месяца, приложение сообщает, какое это время года. Весна (3,4,5), лето (6,7,8), осень (9,10,11) зима (12,1,2))

Домашнее задание (часть 2)

Часть 2. Работа с циклами:

1. При помощи цикла нужно пройтись по числам от 0 до 100 и напечатать только те числа, которые без остатка делятся на 4
2. Есть словарь `dict_ = {"name": "Kirill", "age": 44}`, при помощи цикла нужно вывести на экран:

```
Name: Kirill  
Age: 44
```
3. Пользователь вводит число, нам нужно посчитать сумму чисел от 1 до того, которое ввел пользователь (например, пользователь ввел 4, нужно посчитать $1 + 2 + 3 + 4$)

Домашнее задание (часть 3)

Часть 3* (необязательное).

- 1. Написать калькулятор.** Программа должна принимать от пользователя операцию (например, $1 + 2$) и вычислять результат. Программа должна поддерживать операции $+$, $-$, $*$, $/$. Также программа должна выводить результат и просить пользователя ввести еще 1 выражение, если пользователь вводит "no", то программа завершает работу
- 2. Напечатать таблицу умножения** для чисел от 1 до 10- формат вывода неважен
- 3. Перевернуть число.** Есть число 1234, при помощи цикла нужно перевернуть это число и напечатать 4321 (подсказка: 4321 - строка)

The background is a vibrant yellow color, covered with a repeating pattern of various geometric shapes. These shapes include circles, squares, triangles, and lines, some of which are filled with a fine grid pattern. The shapes are scattered across the entire surface, creating a dynamic and modern aesthetic.

Спасибо за внимание!