

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Реферат на тему:

«Программирования Python для начинающих»

Выполнил
Студент группы ИУ5-31Б
Ван
Чжуэнь

Проверил:
Гапанюк Ю.Е.

2025 г.

Введение

Python — это высокоуровневый язык программирования общего назначения, который я выбрал для изучения, так как ранее не работал с ним. Его ключевые особенности:

- Простой и понятный синтаксис, похожий на псевдокод
- Динамическая типизация
- Поддержка нескольких парадигм (объектно-ориентированное, функциональное, императивное программирование)
- Кроссплатформенность
- Огромное количество библиотек

Python интерпретируемый язык, поэтому для работы нужно установить интерпретатор с официального сайта python.org и среду разработки (я использовал VS Code с расширением Python).

Установка и первый запуск

После установки Python, можно проверить его работу:

```
1 # hello.py
2 print("Привет, мир!") # Вывод текста на экран
```

Запуск в терминале: `python hello.py`

Основные конструкции языка

1. Переменные и типы данных

```
# Динамическая типизация - не указываем тип явно
name = "Анна"          # Стока (str)
age = 25                # Целое число (int)
height = 1.75           # Число с плавающей точкой (float)
is_student = True       # Логический тип (bool)

# Вывод нескольких значений
print(f"Имя: {name}, Возраст: {age}, Рост: {height}")
```

2. Основные структуры данных

Списки (аналог массивов)

```
# Создание списка
fruits = ["яблоко", "банан", "апельсин"]
print(fruits[0]) # Доступ по индексу: "яблоко"

# Изменение списка
fruits.append("киви")      # Добавление элемента
fruits.remove("банан")      # Удаление элемента
print(f"Всего фруктов: {len(fruits)}")

# Перебор элементов
for fruit in fruits:
    print(f"Фрукт: {fruit}")
```

Словари (пары ключ-значение)

```
# Создание словаря
student = {
    "имя": "Иван",
    "возраст": 20,
    "курс": 3
}

# Доступ к значениям
print(student["имя"])          # "Иван"
print(student.get("возраст"))   # 20

# Добавление новой пары
student["специальность"] = "информатика"

# Перебор словаря
for key, value in student.items():
    print(f"{key}: {value}")
```

Условные операторы

```
temperature = 25

if temperature > 30:
    print("Жарко")
elif temperature > 20: # elif = else if
    print("Тепло")
else:
    print("Прохладно")

# Короткая запись
result = "Высокая" if temperature > 28 else "Нормальная"
print(f"Температура: {result}")
```

4. Циклы

```
# Цикл for с range
print("Счёт от 1 до 5:")
for i in range(1, 6):
    print(i)

# Цикл while
counter = 3
while counter > 0:
    print(f"Осталось: {counter}")
    counter -= 1 # Уменьшаем на 1
```

5. Функции

```
# Определение функции
def greet(name, time_of_day="день"):
    """Функция приветствия
    name - имя для приветствия
    time_of_day - время дня (по умолчанию 'день')
    """
    return f"Добрый {time_of_day}, {name}!"

# Вызов функции
message = greet("Мария")
print(message) # Добрый день, Мария!

# Вызов с указанием параметра
print(greet("Алексей", "вечер"))
```

6. Работа с файлами

```
# Запись в файл
with open("пример.txt", "w", encoding="utf-8") as file:
    file.write("Первая строка\n")
    file.write("Вторая строка\n")

# Чтение из файла
with open("пример.txt", "r", encoding="utf-8") as file:
    content = file.read()
    print("Содержимое файла:")
    print(content)
```

Практический пример: простой калькулятор

```

def calculator():
    """Простой калькулятор с базовыми операциями"""

    print("доступные операции:")
    print("+ сложение")
    print("- вычитание")
    print("* умножение")
    print("/ деление")

    try:
        # Ввод данных
        num1 = float(input("Введите первое число: "))
        operation = input("Введите операцию (+, -, *, /): ")
        num2 = float(input("Введите второе число: "))

        # Выполнение операции
        if operation == "+":
            result = num1 + num2
        elif operation == "-":
            result = num1 - num2
        elif operation == "*":
            result = num1 * num2
        elif operation == "/":
            if num2 == 0:
                print("Ошибка: деление на ноль!")
                return
            result = num1 / num2
        else:
            print("Неизвестная операция!")
            return

        # Вывод результата
        print(f"Результат: {num1} {operation} {num2} = {result}")

    except ValueError:
        print("Ошибка: введите числа корректно!")

# Запуск калькулятора
if __name__ == "__main__":
    calculator()

```

Особенности Python

1. Отступы вместо скобок - отступы (табуляция или пробелы) определяют блоки кода
2. Динамическая типизация - типы определяются во время выполнения
3. Интерпретируемость - код выполняется построчно без компиляции
4. Большая стандартная библиотека - много модулей "из коробки"
5. Простота чтения - код похож на английский язык

Преимущества для новичков

1. Минимальный синтаксический шум - меньше служебных символов
2. Читаемость - код понятен даже без комментариев
3. Быстрый старт - можно писать полезные программы уже через час изучения

4. Поддержка сообщества - огромное количество учебных материалов

Заключение

Python оказался идеальным языком для начала изучения программирования. Его простой синтаксис позволяет сосредоточиться на логике решения задач, а не на технических деталях языка. Динамическая типизация и автоматическое управление памятью делают код лаконичным. Огромное сообщество и богатая экосистема библиотек позволяют решать практически любые задачи — от веб-разработки до научных вычислений.

Для запуска примеров из этого реферата достаточно сохранить код в файл с расширением .py и выполнить команду `python имя_файла.py` в терминале.