



Методические указания для практической работы №1

Программирование на Python

Создание генератора случайных рецептов на Python

Цель

Изучить базовый синтаксис Python путём создания программы, генерирующей случайные названия блюд и рецепты, используя предопределенные списки ингредиентов.

1. Объявление переменных

С:

```
char ingredients[] = "курица";  
char* array[3] = {"курица", "рыба", "говядина"};
```

Python:

```
ingredient: str = "курица"  
ingredients: list[str] = ["курица", "рыба", "говядина"]
```

2. Типы данных

- В Python не нужно объявлять тип переменной заранее
- `list[str]` - это подсказка типа (type hint), помогает понять, что переменная содержит список строк

3. Функции

В С:

```
char* generate_name() {  
    return "название";  
}
```

В Python:

```
def generate_name() → str:  
    return "название"
```

Пошаговое выполнение задания

Шаг 1: Импорт модуля random

```
import random
```

Этот модуль содержит функции для работы со случайными числами и выбором случайных элементов.

Шаг 2: Создание списков ингредиентов

Создайте пять глобальных списков (констант):

```
MAIN_INGREDIENTS: list[str] = [  
    "а",
```

```
"б",  
"в",  
]
```

Важно:

- Имена констант пишутся БОЛЬШИМИ_БУКВАМИ
- `list[str]` означает "список строк" (list of strings)
- Запятая после последнего элемента допустима и рекомендуется

Аналогично создайте списки:

- `SIDES` (гарниры)
- `SAUCES` (соусы)
- `SPICES` (специи)
- `ADJECTIVES` (прилагательные)
- `NOUNS` (существительные)

Шаг 3: Функция генерации названия блюда

- `random.choice(<list>)` - выбирает случайный элемент из списка
- `<str>.capitalize()` - делает первую букву заглавной
- `f"{переменная}"` - f-строка для вставки значений переменных в текст

Шаг 4: Основная функция генерации рецепта

```
def generate_recipe() → str:  
    """Создает случайный рецепт с одним ингредиентом из каждой категории."""
```

В этой функции:

1. **Выберите случайные ингредиенты:**
2. **Создайте список шагов приготовления:**
3. **Сформируйте итоговый рецепт:**

Новые концепции:

- `\n` - символ новой строки (как в C)
- `"\n".join(список)` - соединяет элементы списка через указанный разделитель
- Многострочные строки можно создавать, заключив их в скобки

Шаг 5: Точка входа программы

```
if __name__ == "__main__":  
    print(generate_recipe())
```

Объяснение:

- `__name__` - специальная переменная Python
- `"__main__"` - значение, когда файл запускается напрямую
- Аналог `main()` функции в C

Важные отличия Python от C

Отступы вместо фигурных скобок

В C:

```
if (condition) {  
    statement1;
```

```
statement2;  
}
```

В Python:

```
if condition:  
    statement1  
    statement2
```

Нет точек с запятой

В Python точки с запятой не нужны в конце строк.

Строки

- В Python строки можно складывать: `"Hello" + " World"`
- f-строки для форматирования: `f"Имя: {name}"`

Возможные ошибки и их решения

1. **IndentationError** - проверьте отступы (4 пробела или табуляция, но одинаково везде)
2. **NameError** - проверьте правильность написания имен переменных
3. **ImportError** - убедитесь, что `import random` находится в начале файла