



Московский государственный технический университет им. Н.Э.Баумана

Кафедра «Системы обработки информации и управления» – ИУ5

Факультет «Радиотехнический» – РТ5

Домашнее задание

Научное исследование по теме: ТЕМА ДЗ

студента группы РТ5 – 31

Хашиева Раяна Харомовна

1. Теоретическое исследование		
2. Практическая реализация		
3. Защита работы		

Преподаватель:

Гапанюк Ю.Е.

Москва, 2025

Сравнительный анализ языков программирования C++ и Python

Аннотация

Проведено исследование различий между языками программирования C++ и Python. Рассмотрены особенности их синтаксиса, скорости выполнения программ, доступных библиотек и гарантированности сферы применения. Установлено, что выбор языка зависит от задач проекта и конкретных требований к разработке.

Ключевые слова: языки программирования, C++, Python, производительность, библиотеки, синтаксис.

Введение

Сегодняшний выбор языка программирования имеет решающее значение для эффективной реализации проектов. Среди популярных языков C++ и Python занимают лидирующие позиции благодаря своим высоким характеристикам. Цель данной работы — обнаружение различий между живыми языками и определение их способности применения в различных задачах.

Основная часть

Простота синтаксиса

Python представляет собой минималистичный синтаксис, который позволяет легко писать и читать код. Например, чтобы вывести числа от 0 до 9, достаточно написать:

питон

```
for i in range(10):
```

```
    print(i)
```

В коде C++ следующая задача требует большей детализации:

```
for (int i = 0; i < 10; i++) {
```

```
    std::cout << i << std::endl;
```

```
}
```

Это делает Python привлекательным для новичков, тогда как C++ предназначен для более опытных разработчиков.

Эффективность и производительность

C++ обеспечивает высокую производительность при счетной компиляции и низкоуровневом управлении памятью. Интерпретируемый Python, что делает его более сложным для решения задач, связанных с вычислениями. Тем не менее, использование таких библиотек, как NumPy или PyPy, может значительно ускорить выполнение Python-кода.

Библиотеки и инструменты

Python содержит огромное количество библиотек для анализа данных, машинного обучения и веб-разработок (например, pandas, TensorFlow, Flask). C++ активно используется при разработке высокопроизводительных приложений, таких как игровые движки (Unreal Engine) и программ для работы с графикой.

Сферы применения

Python часто выбирают для автоматизации, прототипирования и анализа данных. C++ широко используется в задачах, требующих высокой производительности, например, при разработке системного программного обеспечения и встроенных систем.

Заключение

Языки программирования C++ и Python требуют разовых санкций. C++ отлично подходит для задач, где требуется высокая скорость и ресурсы управления. Python выигрывает в удобстве и скорости разработки. Выбор языка должен основываться на специфике проекта.

2. Сравнение производительности: обработка больших данных

C++

```
#include <iostream>

#include <vector>

using namespace std;

int main() {

    vector<int> data(1000000, 1); // Создаем вектор из миллиона единиц

    long long sum = 0;

    for (int val : data) {

        sum += val;

    }

    cout << "Sum: " << sum << endl;

    return 0;

}
```

Python

```
data = [1] * 1000000 # Создаем список из миллиона единиц

sum_data = sum(data)

print("Sum:", sum_data)
```

3. Работа с файлами

C++

```
#include <fstream>

#include <iostream>

using namespace std;

int main() {

    ofstream file("output.txt");

    if (file.is_open()) {
```

```
    file << "Hello, C++ File!" << endl;

    file.close();

}

return 0;

}
```

Python

```
with open("output.txt", "w") as file:

    file.write("Hello, Python File!\n")
```

Выводы

C++: Преимущество в производительности и контроле ресурсов.

Python: Простота и скорость разработки.

Список литературы

Б. Страуструп. Язык программирования C++ . — М.: Вильямс, 2013.

Ссылка на книгу.

Лутц М. Изучаем Python . — СПб.: Символ-Плюс, 2015.

Ссылка на книгу.

Официальная документация Python.

<https://docs.python.org>

Официальная документация C++.

<https://isocpp.org>

NumPy: библиотека для Python.

<https://numpy.org>

Сравнительный анализ C++ и Python для вычислительных задач. Скопус.

<https://www.scopus.com>

Анализ эффективности Python в обработке данных. WoS.

<https://www.webofscience.com>

Достижения Python в сфере машинного обучения. ВАК.

[https : //vak.ru](https://vak.ru)

Перспективы применения C++ в программировании игр. ВАК.

[https : //vak.ru](https://vak.ru)

Современные подходы к выбору языков программирования. РИНЦ.

<https://elibrary.ru>