

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM



BÁO CÁO ĐỒ ÁN MÔN HỌC
GIỚI THIỆU NGÀNH KỸ THUẬT PHẦN MỀM
ĐỀ TÀI GIỚI THIỆU TỔNG QUAN VỀ NHỮNG NGÔN NGỮ LẬP TRÌNH

Giảng viên hướng dẫn : Trần Anh Dũng

NHÓM 15

Sinh viên thực hiện 1 : Kim Thành Tiến
Mã sinh viên 1 : 23521571
Sinh viên thực hiện 2 : Võ Sĩ Trí Thông
Mã sinh viên 2 : 23521531
Sinh viên thực hiện 3 : Lữ Nguyễn Duy Đức
Mã sinh viên 3 : 23520305
Sinh viên thực hiện 4 : Tiêu Hoàng Phúc
Mã sinh viên 4 : 23521220
Sinh viên thực hiện 5 : Từ Minh Khoa
Mã sinh viên 5 : 23520757

Lớp: SE005.O13

TP. HỒ CHÍ MINH, THÁNG 11 NĂM 2023

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành nhất tới thầy Trần Anh Dũng - giảng viên khoa Công nghệ phần mềm, đã hướng dẫn chúng em thực hiện đồ án môn học này. Thầy đã tận tình hướng dẫn và góp ý cho chúng em từng bước để chúng em có thể hoàn thành tốt đồ án. Đồ án không chỉ cung cấp cho chúng em những kiến thức và kỹ năng cần thiết cho tương lai mà còn giúp định hướng công việc sau này. Đây là kinh nghiệm quý báu mà chúng em sẽ ghi nhớ mãi trong quá trình học tập. Một lần nữa, chúng em xin cảm ơn thầy vì tất cả những gì thầy đã dành cho chúng em! Chúc thầy luôn mạnh khỏe, hạnh phúc và thành công trong sự nghiệp cao cả của mình.

This image shows a full page of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.

(Ký tên và ghi rõ họ tên)

MỤC LỤC

LỜI CẢM ƠN	1
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	2
LỜI MỞ ĐẦU	4
CHƯƠNG 1. GIỚI THIỆU YÊU CẦU	5
CHƯƠNG 2. NỘI DUNG BÁO CÁO.....	6
1. Python.....	6
2. JavaScript	9
3. Java.....	12
4. C#.....	15
5. C++	18
6. Ruby	20
7. PHP.....	22
8. Swift.....	24
9. Kotlin.....	26
10. Go	29
11. Rust	31
12. SQL	33
13. Scala	35
14. TypeScript	37
15. Haskell.....	39
CHƯƠNG 3. KẾT LUẬN.....	41
3.1. Kết quả đạt được.....	41
3.1. Bảng phân công công việc	42
TÀI LIỆU THAM KHẢO	43

LỜI MỞ ĐẦU

Sự phát triển vũ bão của công nghệ thông tin đã làm thay đổi mọi mặt đời sống xã hội. Việc ứng dụng công nghệ thông tin vào mọi lĩnh vực đang trở nên phổ biến, từ giáo dục, y tế cho đến thương mại điện tử, giao thông vận tải, ngân hàng, ... Chính vì thế, việc học tập và nắm vững kiến thức về công nghệ thông tin là hết sức cần thiết cho sinh viên ngày nay. Cốt lõi của công nghệ thông tin chính là các ngôn ngữ lập trình. Kể từ khi ra đời cho đến nay, đã có rất nhiều ngôn ngữ lập trình được phát triển, mỗi ngôn ngữ đều có những đặc điểm và ứng dụng riêng. Đồ án này sẽ giới thiệu tổng quan về những đặc điểm của ngôn ngữ lập trình phổ biến hiện nay như: lịch sử ra đời, cú pháp cơ bản, tính năng, ứng dụng và ví dụ minh họa về mỗi ngôn ngữ. Việc tìm hiểu và nắm vững các ngôn ngữ lập trình sẽ giúp sinh viên có cái nhìn tổng quan về công nghệ thông tin, từ đó lựa chọn hướng đi phù hợp cho bản thân trong tương lai.

CHƯƠNG 1. GIỚI THIỆU YÊU CẦU

- Những yêu cầu cần phải thực hiện trong đề tài giới thiệu về các ngôn ngữ lập trình:
 - Giới thiệu được khái niệm cơ bản của lập trình, các loại ngôn ngữ lập trình phổ biến hiện nay.
 - Nắm được nguồn gốc, lịch sử hình thành của các ngôn ngữ phổ biến, quá trình phát triển từ quá khứ đến hiện tại.
 - Hiểu được cú pháp và cấu trúc cơ bản, cách ứng dụng vào để giải các bài toán trong lập trình.
 - Nắm được ứng dụng và lĩnh vực sử dụng, ưu điểm và nhược điểm của các loại ngôn ngữ lập trình phổ biến.
 - Đưa ra một số ứng dụng minh họa về ứng dụng của các loại ngôn ngữ lập trình phổ biến.
 - Tham khảo và biết được một số nguồn tài liệu và một số cộng đồng lớn để tự học và tìm hiểu phát triển kỹ năng lập trình.
- Điều quan trọng nhất là sau khi hiểu được những điều cơ bản thì phải nắm bắt được xu hướng thay đổi của công nghệ ngày nay và đưa ra được định hướng phù hợp cho bản thân, tránh tình trạng mất phương hướng trong việc chọn chuyên ngành của đại đa số sinh viên công nghệ thông tin hiện nay.

CHƯƠNG 2. NỘI DUNG BÁO CÁO

1. Python

a. Lịch sử, nguồn gốc

- Do lập trình viên Hà Lan ra mắt vào 1991.
- Phiên bản đầu tiên ngôn ngữ này bao gồm các tính năng hữu ích như một số kiểu dữ liệu và hàm để xử lý lỗi.
- Được tạo ra ban đầu chỉ để “giết thời gian” trong dịp Giáng sinh của ông.

b. Cú pháp

- Khai báo biến:

Python là một ngôn ngữ có thụ động kiểu dữ liệu, nghĩa là bạn không cần phải xác định kiểu dữ liệu của biến trước khi sử dụng nó.

```
age = 30          # Biến age lưu trữ giá trị 30
name = "Alice"    # Biến name lưu trữ chuỗi "Alice"
x = 3.14          # Biến x lưu trữ số thập phân 3.14
is_student = True # Biến is_student lưu trữ giá trị Boolean True
```

- If- else :

```
if điều_kiện:
    # Khối mã sẽ thực hiện nếu điều_kiện là True
else:
    # Khối mã sẽ thực hiện nếu điều_kiện là False
```

- While-do:

```
while điều_kiện:
    # Khối mã sẽ lặp lại trong khi điều_kiện là True
```

- Lưu ý: không hỗ trợ do while.

- **Function:** không như các ngôn ngữ khác, python là ngôn ngữ lập trình động nên chỉ cần từ khóa def khi tạo hàm

```
def add(a, b):  
    return a + b  
  
def say_hello(name):  
    return "Xin chào, " + name
```

- **Class:**

```
class Dog:  
    def __init__(self, name):  
        self.name = name  
  
    def bark(self):  
        print(f"{self.name} gâu gâu!")
```

c. Ứng dụng

- Python là ngôn ngữ lập trình đa năng được ứng dụng rộng rãi trong nhiều lĩnh vực và ứng dụng khác nhau:
 - **Phát triển web, game :** framework Django và Flask. Nó giúp xây dựng các ứng dụng web mạnh mẽ và dễ bảo trì.
 - **Khoa học dữ liệu và Machine Learning:** Các thư viện như NumPy, Pandas, TensorFlow và PyTorch được sử dụng rộng rãi để phân tích dữ liệu và xây dựng mô hình máy học.
 - **Automating Tasks và Scripting:** Python thường được sử dụng để tự động hóa các tác vụ và quy trình, cũng như để viết các script. Điều này bao gồm tự động hóa xử lý dữ liệu, quản lý tệp tin, gửi email tự động, và nhiều công việc khác

d. Ưu, nhược điểm

- Ưu:
 - Dễ học và đọc, cộng đồng lớn
 - Đa năng, hỗ trợ đối tượng
 - Thư viện, framework phong phú, tương thích đa nền tảng
- Nhược:
 - Tốc độ thực thi chậm
 - Note: Python sử dụng thụt lề để định nghĩa khối lệnh, điều này đòi hỏi duy trì một thụt lề chính xác để tránh lỗi cú pháp.
 - Kích thước tệp lớn cho ứng dụng đóng gói

2. JavaScript

a. Lịch sử nguồn gốc:

- JavaScript là một ngôn ngữ lập trình được phát triển bởi Brendan Eich vào năm 1995
- JavaScript đã trải qua nhiều giai đoạn phát triển và tiêu chuẩn hóa, và hiện nay là một trong những ngôn ngữ lập trình phổ biến nhất thế giới.

b. Cú pháp:

- Khai báo biến:

```
var tenBien = 1;  
let tenBien = "Hello World";
```

- If-else:

```
if (condition1) {  
    // code1  
} else if (condition2) {  
    // code2  
} else {  
    // code3  
}
```

- While-do:

```
while (condition) {  
    // code  
}
```

```
do {  
    // code  
} while (condition);
```

- Function:

```
function tenHam(thamSo1, thamSo2,  
...) {  
    // code  
}
```

- Class:

```
class ClassName {  
    constructor(parameters) {  
        // Constructor code  
    }  
  
    method1() {  
        // Method 1 code  
    }  
  
    method2() {  
        // Method 2 code  
    }  
}
```

c. Ứng dụng:

- JavaScript có thể được ứng dụng trong nhiều lĩnh vực khác nhau, chẳng hạn như:
 - **Lập trình web:** Javascript có thể tạo ra các hiệu ứng, chức năng, hoạt ảnh và xử lý các sự kiện trên các trang web bằng cách sử dụng framework như React, Angular, Vue...
 - **Xây dựng ứng dụng cho website máy chủ:** JavaScript có thể phát triển các ứng dụng web phía máy chủ bằng cách sử dụng Node.js, một công cụ thời gian chạy JavaScript
 - **Ứng dụng di động, app, trò chơi:** JavaScript cũng có thể được sử dụng để tạo ra các ứng dụng di động, app, và trò chơi bằng cách sử dụng các công cụ và engine như React Native, Ionic, Phaser, Babylon.

d. Ưu, nhược điểm:

- Ưu:
 - Cộng đồng người dùng JavaScript rất đa dạng và phong phú.
 - Về hiệu suất, JavaScript được xem là một ngôn ngữ lập trình động (dynamic programming language) đã được tối ưu hóa nhằm nâng cao hiệu suất cho lập trình viên.
- Nhược:
 - Hiệu năng trình duyệt: Hiệu năng của mã JavaScript có thể bị ảnh hưởng bởi tải trình duyệt và tài nguyên của trang web
 - Về bảo mật, JavaScript không cho phép đọc và ghi các file vì lý do bảo mật. Nó không được sử dụng cho việc kết nối mạng các ứng dụng bởi vì không có những hỗ trợ có sẵn. Tuy nhiên, nếu trang web bị hacker kiểm soát, họ có thể sử dụng các đoạn code độc hại để chạy một chương trình

3. Java

a. Lịch sử nguồn gốc:

- Ra đời những năm 1990, được phát triển ban đầu bởi James Gosling
- Java ra đời với mục tiêu tạo ra một ngôn ngữ lập trình đa năng, có khả năng chạy trên nhiều nền tảng, đặc biệt là trên Internet

b. Cú pháp:

- Khai báo biến
- If-else

```
if (condition) {  
    // Code khi điều kiện đúng  
} else {  
    // Code khi điều kiện sai  
}
```

- While do

```
while (condition) {  
    // Khối mã sẽ được lặp lại cho đến khi điều kiện sai  
}
```

- Function

```
public int add(int a, int b) {  
    return a + b;  
}
```

- Class

```
public class Person {  
    String name;  
    int age;  
  
    public Person(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
}
```

c. Ứng dụng:

- Lập trình ứng dụng di động
- Lập trình web
- Lập trình nhúng

d. Ưu, nhược điểm:

- Ưu:
 - **Độc lập nền tảng:** Java có khả năng chạy trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn gốc.
 - **Bảo mật:** Java có các tính năng bảo mật tích hợp, giúp ngăn chặn các lỗ hổng bảo mật.
 - **Hướng đối tượng:** Java hỗ trợ lập trình hướng đối tượng, giúp dễ dàng quản lý và tái sử dụng mã nguồn.
 - **Thư viện phong phú:** Java có nhiều thư viện và framework mạnh mẽ để phát triển các ứng dụng phức tạp.
 - **Quản lý bộ nhớ tự động:** Java sử dụng Garbage Collection để quản lý bộ nhớ tự động, giúp tránh lỗi liên quan đến quản lý bộ nhớ.
- Nhược:
 - **Tốc độ:** Java thường chậm hơn các ngôn ngữ lập trình biên dịch khác.

- Đòi hỏi tài nguyên: JVM và thư viện Java có thể tiêu tốn nhiều tài nguyên hệ thống.
- Phiên toái: Java có thể yêu cầu nhiều mã nguồn hơn so với các ngôn ngữ khác để thực hiện cùng một chức năng.
- Khó học: Đôi khi, người mới học lập trình có thể cảm thấy Java khá phức tạp.

4. C#

a. Lịch sử nguồn gốc:

- Là một ngôn ngữ lập trình được phát triển bởi Microsoft vào cuối những năm 1990.
- C# được thiết kế để kết hợp tính năng của C/C++ và đơn giản hóa việc lập trình so với C++.

b. Cú pháp cơ bản:

- Khai báo biến
 - Cách khai báo biến trong C# sử dụng từ khoá var hoặc kiểu dữ liệu cụ thể.

```
var age = 30;  
int count = 0;
```

- If else

```
if (condition)  
{  
    // Thực hiện khi điều kiện đúng  
}  
else  
{  
    // Thực hiện khi điều kiện sai  
}
```

- While do

```
while (condition)  
{  
    // Các câu lệnh được thực hiện khi điều kiện đúng  
}
```



```
do
{
    // Các câu lệnh được thực hiện ít nhất một lần
} while (condition);
```

- Function

```
[modifiers] return_type function_name(parameters)
{
    // Mã thực hiện bởi hàm
    return value; // Trả về giá trị (nếu có)
}
```

- **`modifiers`** (tùy chọn): Các từ khóa như **`public`**, **`private`**, **`static`**, **`protected`**,... để xác định quyền truy cập và tính chất của hàm.
- **`return_type`**: Kiểu dữ liệu mà hàm trả về sau khi thực hiện xong.
- **`function_name`**: Tên của hàm, theo quy tắc đặt tên biến.
- **`parameters`** (tùy chọn): Danh sách các tham số mà hàm nhận để thực hiện công việc. Mỗi tham số có thể bao gồm kiểu dữ liệu và tên của tham số.

- Class

```
[modifiers] class Class_Name
{
    // Các thành viên của lớp (thuộc tính và phương thức)
}
```

c. Ứng dụng:

- Ứng dụng web, game
- Ứng dụng di động

- Ứng dụng máy chủ

d. Ưu nhược điểm:

- Ưu:
 - Là ngôn ngữ mạnh mẽ và linh hoạt
 - Hỗ trợ OOP giúp dễ bảo trì
 - Tương thích nền tảng Windows.
 - Hỗ trợ mạnh mẽ từ Microsoft và cộng đồng c# rộng lớn
- Nhược:
 - Không phù hợp ứng dụng nhẹ và tài nguyên thấp
 - Thường sử dụng chủ yếu trong hệ sinh thái Microsoft, do đó hạn chế tính di động và đa nền tảng

5. C++

a. Lịch sử nguồn gốc:

- C++ được phát triển bởi Bjarne Stroustrup vào những năm 1980 tại Bell Labs.
- Nó được phát triển như một mở rộng của ngôn ngữ C, nên có thể sử dụng hầu hết mã C trong mã C++ mà không cần chỉnh sửa.

b. Cú pháp cơ bản:

- Khai báo biến:

```
int a = 5;  
double pi = 3.14159;
```

- If else

```
cpp Copy code  
  
if (điều kiện) {  
    // Mã lệnh sẽ thực thi nếu điều kiện là đúng  
} else {  
    // Mã lệnh sẽ thực thi nếu điều kiện là sai (tùy chọn, có thể không có)  
}
```

- While do

```
cpp Copy code  
  
while (điều kiện) {  
    // Mã lệnh trong vòng lặp  
}
```

- Function

```
int add(int a, int b) {  
    return a + b;  
}
```

- Class

```
class Circle {  
public:  
    double radius;  
    double getArea() {  
        return 3.14159 * radius * radius;  
    }  
};
```

c. Ứng dụng:

- Phát triển phần mềm, trò chơi
- Hệ thống nhúng
- C++ là ngôn ngữ lựa chọn cho việc phát triển các phần mềm yêu cầu hiệu suất cao và kiểm soát gần về phần cứng.

d. Ưu nhược điểm:

- Ưu :
 - Hỗ trợ lập trình hướng đối tượng.
 - Hiệu suất cao với quyền kiểm soát cận thiết về bộ nhớ và tài nguyên hệ thống.
 - Thư viện chuẩn mạnh mẽ với nhiều công cụ và tài liệu phong phú.
 - Tốc độ thực thi nhanh vì vậy thường được ứng dụng lập trình thi đấu
- Nhược :
 - Phức tạp hơn so với một số ngôn ngữ lập trình khác.
 - Dễ gặp lỗi như lỗi quản lý bộ nhớ (memory management).
 - Đòi hỏi nhiều kiến thức để sử dụng hiệu quả.

6. Ruby

a. Lịch sử nguồn gốc:

- Năm 1993, bởi lập trình viên người Nhật Bản tên là Matsumoto Yukihiro (biệt danh Matz)
- Muốn tạo ra một ngôn ngữ kịch bản sử dụng lập trình hướng đối tượng và tăng khả năng tái sử dụng mã để giúp tăng tốc độ phát triển

b. Cú pháp cơ bản

- Khai báo biến:

```
ruby

age = 25
name = "John"
```

- If else:

```
ruby

if condition
  # Thực hiện khi condition đúng
else
  # Thực hiện khi condition sai
end
```

- While do:

```
ruby

while condition
  # Thực hiện mã này trong khi condition đúng
end
```

- Function:

ruby

```
def greet(name)
  puts "Hello, #{name}!"
end
```

c. Ứng dụng:

- Ruby được sử dụng rộng rãi trong việc phát triển web và ứng dụng web thông qua framework Ruby on Rails.
- Ngoài ra, Ruby cũng được sử dụng trong scripting, xử lý văn bản, và nhiều mục đích khác.

d. Ưu nhược điểm:

- Ưu
 - Cú pháp dễ đọc, gần với ngôn ngữ tự nhiên, làm cho việc viết mã trở nên thân thiện và dễ hiểu.
 - Hỗ trợ lập trình hướng đối tượng mạnh mẽ.
 - Ruby on Rails cung cấp môi trường phát triển web nhanh chóng và hiệu quả.
- Nhược
 - Hiệu suất không cao so với một số ngôn ngữ khác như C++ hoặc Java.
 - Khả năng tương thích ngược với một số thư viện cũ hoặc các phiên bản khác nhau của Ruby có thể tạo ra vấn đề.

7. PHP

a. Lịch sử nguồn gốc:

- Ban đầu, PHP được tạo ra như một ngôn ngữ để quản lý các trang web động (dynamic web pages) và biến đổi thông tin trên server.
- Sau đó phát triển thành một ngôn ngữ lập trình web mạnh mẽ với khả năng tương tác với cơ sở dữ liệu và xây dựng ứng dụng web phức tạp.

b. Cú pháp cơ bản:

- Khai báo biến:

php

```
$name = "John";  
$age = 30;
```

- If else:

php

```
if ($condition) {  
    // Thực hiện khi condition đúng  
} else {  
    // Thực hiện khi condition sai  
}
```

- While do:

php

```
while ($condition) {  
    // Thực hiện mã này trong khi condition đúng  
}
```

php

```
do {  
    // Thực hiện ít nhất một lần, sau đó kiểm tra điều kiện  
} while ($condition);
```

- Function:

php

```
function greet($name) {  
    echo "Hello, $name!";  
}
```

c. Ứng dụng:

- PHP thường được sử dụng để phát triển ứng dụng web, xây dựng trang web động và tương tác với cơ sở dữ liệu.
- Nó là ngôn ngữ chủ đạo trong các framework web như Laravel, Symfony, và WordPress.

d. Ưu nhược điểm:

- Ưu:
 - Dễ học và triển khai.
 - Hỗ trợ tốt cho kết nối cơ sở dữ liệu, phù hợp cho việc xây dựng ứng dụng web động.
 - Có cộng đồng lớn, nhiều tài liệu và thư viện hỗ trợ.
- Nhược:
 - Cú pháp không linh hoạt và có thể gây nhầm lẫn.
 - Hiệu suất có thể thấp so với một số ngôn ngữ khác như Java hoặc C++.
 - Bị chỉ trích về một số vấn đề an ninh và quản lý mã nguồn mở.

8. Swift

a. Lịch sử nguồn gốc:

- Swift là một ngôn ngữ lập trình mới được phát triển bởi Apple vào năm 2014 để thay thế Objective-C trong việc phát triển ứng dụng cho hệ điều hành iOS và các sản phẩm khác của Apple.
- Nó được thiết kế để cung cấp hiệu suất cao, đồng thời bảo đảm tính an toàn và dễ đọc.

b. Cú pháp cơ bản:

- Khai báo biến

```
var tenSinhVien = "Nguyễn Văn A"
var tuoiSinhVien = 20

let pi = 3.14159
let soLuongBanGhe = 50
```

- If else

```
swift

if condition {
    // Thực hiện khi condition đúng
} else {
    // Thực hiện khi condition sai
}
```

- While do

```
swift

while condition {
    // Thực hiện mã này trong khi condition đúng
}
```

- Function

```
swift

func greet(name: String) {
    print("Hello, \(name)!")
}
```

c. Ứng dụng

- Swift chủ yếu được sử dụng để phát triển ứng dụng cho hệ điều hành iOS, macOS, watchOS và tvOS.
- Nó được sử dụng trong việc xây dựng ứng dụng di động, từ ứng dụng đơn giản đến ứng dụng phức tạp.

d. Ưu nhược điểm

- Ưu:
 - Cú pháp rõ ràng và dễ đọc, giúp làm giảm số lượng lỗi và tăng khả năng bảo trì mã nguồn.
 - An toàn về kiểu (type-safe) giúp tránh nhiều lỗi phổ biến khi lập trình.
 - Hiệu suất cao và tương thích tốt với các sản phẩm Apple.
- Nhược:
 - Môi trường phát triển (IDE) thường chỉ hỗ trợ trên các hệ điều hành của Apple.
 - Thư viện và tài liệu có thể ít so với các ngôn ngữ khác đã tồn tại lâu.

9. Kotlin

a. Lịch sử nguồn gốc:

- Kotlin là ngôn ngữ lập trình tĩnh (static) do JetBrains phát triển nhằm mục đích khắc phục những hạn chế của Java trong khi vẫn duy trì khả năng tương thích hoàn toàn với các cơ sở mã và khung công tác Java hiện có.

b. Cú pháp cơ bản:

- Khai báo biến

kotlin

```
var age = 25  
val name = "John"
```

- If else

kotlin

```
if (condition) {  
    // Thực hiện khi condition đúng  
} else {  
    // Thực hiện khi condition sai  
}
```

- While do

kotlin

```
do {  
    // Thực hiện ít nhất một lần, sau đó kiểm tra điều kiện  
} while (condition)
```

kotlin

```
while (condition) {  
    // Thực hiện mã này trong khi condition đúng  
}
```

- Function

kotlin

```
fun greet(name: String) {  
    println("Hello, $name!")  
}
```

c. Ứng dụng:

- Lĩnh vực sử dụng: Kotlin được sử dụng để lập trình các ứng dụng trên Android, lập trình web và lập trình phần mềm cho PC.

d. Ưu nhược điểm:

- Ưu:
 - Hiện nay, cộng đồng phát triển của Kotlin khá rộng lớn và trải dài khắp thế giới. Chính vì vậy nên nguồn tài liệu để học tập và nghiên cứu cũng khá dễ dàng tìm được.

- Hiệu suất: so sánh với Java. Về cơ bản, Kotlin kế thừa tất cả những ưu điểm mà Java có, song nó cũng khắc phục một số nhược điểm của Java khiến cho hiệu suất làm việc của Kotlin cao hơn hẳn.
- Nhược:
 - Cộng đồng người dùng và tài liệu có thể ít hơn so với một số ngôn ngữ đã tồn tại lâu.
 - Môi trường phát triển (IDE) không tương thích tốt với một số công cụ.

10.Go

a. Lịch sử nguồn gốc:

- Go, hay còn được gọi là Golang, là một ngôn ngữ lập trình được phát triển bởi Google và ra mắt lần đầu vào năm 2009.
- Nó được thiết kế để kết hợp giữa hiệu suất cao, tính đa nhiệm, và việc phát triển ứng dụng dễ dàng.

b. Cú pháp cơ bản

- Khai báo biến

```
go

var age int = 25
name := "John" // cú pháp ngắn gọn để khai báo biến
```

- If else

```
go

if condition {
    // Thực hiện khi condition đúng
} else {
    // Thực hiện khi condition sai
}
```

- While do
 - Trong ngôn ngữ Go, không có cú pháp cụ thể cho vòng lặp while hay do-while như các ngôn ngữ khác

go

```
for i := 0; i < 5; i++ {  
    // Thực hiện vòng lặp  
}
```

- Function

go

```
func greet(name string) {  
    fmt.Println("Hello, " + name + "!")  
}
```

c. Ứng dụng

- Go thường được sử dụng cho các ứng dụng có yêu cầu về hiệu suất cao, như phát triển hệ thống, ứng dụng máy chủ, công cụ và các dịch vụ mạng.
- Nó cũng được sử dụng rộng rãi trong việc phát triển các ứng dụng dựa trên hệ thống cloud.

d. Ưu nhược điểm

- Ưu
 - Hiệu suất cao với việc tối ưu hóa tốt.
 - Cú pháp đơn giản, dễ đọc và dễ viết.
 - Hỗ trợ đa nhiệm tốt và tương thích tốt với môi trường phân tán.
- Nhược
 - Thư viện tiêu chuẩn có thể không phong phú như các ngôn ngữ khác.
 - Hỗ trợ lập trình hướng đối tượng có thể không linh hoạt như một số ngôn ngữ khác.

11. Rust

a. Lịch sử nguồn gốc:

- Rust được tạo ra bởi Graydon Hoare tại Mozilla vào năm 2006. Nó được phát triển từ dự án Mozilla Research với mục tiêu ban đầu là thay thế ngôn ngữ C và C++.
- Mục tiêu của Rust là cung cấp một ngôn ngữ lập trình an toàn, hiệu suất cao và có thể đồng thời xử lý các vấn đề lập trình cũng như các vấn đề hệ thống.

b. Cú pháp cơ bản

- Khai báo biến

rust

```
let age = 25;  
let name = "John";
```

- If else

rust

```
if condition {  
    // Thực hiện khi condition đúng  
} else {  
    // Thực hiện khi condition sai  
}
```

- While do

rust

```
while condition {  
    // Thực hiện mã này trong khi condition đúng  
}
```


rust

```
loop {  
    // Thực hiện vòng lặp vô hạn hoặc sử dụng break để thoát  
}
```

- Function

rust

```
fn greet(name: &str) {  
    println!("Hello, {}!", name);  
}
```

c. Ứng dụng

- Rust thường được sử dụng để phát triển các ứng dụng an toàn, hiệu suất cao và độ tin cậy, đặc biệt trong lĩnh vực hệ thống, phần mềm nhúng, công cụ và các ứng dụng web.

d. Ưu nhược điểm

- Ưu
 - An toàn về bộ nhớ với kiểm tra lỗi tĩnh.
 - Hiệu suất cao và có thể đồng thời xử lý nhiều vấn đề lập trình.
 - Hỗ trợ đa luồng mạnh mẽ và bảo mật.
- Nhược
 - Cú pháp và quy tắc mượt mà của Rust có thể gây khó khăn ban đầu cho người mới học.
 - Thư viện có thể chưa phong phú như một số ngôn ngữ lập trình khác.

12. SQL

a. Lịch sử nguồn gốc:

- SQL ra đời vào những năm 1970 khi Edgar F. Codd, một nhà khoa học máy tính, công bố mô hình dữ liệu quan hệ và ngôn ngữ truy vấn cơ sở dữ liệu quan hệ tương ứng.
- Từ đó, SQL đã trở thành ngôn ngữ chuẩn để truy cập và quản lý cơ sở dữ liệu quan hệ.
- SQL ra đời vào những năm 1970 khi Edgar F. Codd, một nhà khoa học máy tính, công bố mô hình dữ liệu quan hệ và ngôn ngữ truy vấn cơ sở dữ liệu quan hệ tương ứng.
- Từ đó, SQL đã trở thành ngôn ngữ chuẩn để truy cập và quản lý cơ sở dữ liệu quan hệ.

b. Cú pháp cơ bản


- Tạo database: `CREATE DATABASE <tên database>;`
- Select: Dùng để chèn dữ liệu mới vào bảng

sql

```
SELECT column1, column2 FROM table_name WHERE condition;
```

- Insert into: Dùng để chèn dữ liệu mới vào bảng


sql

 Copy code

```
INSERT INTO table_name (column1, column2) VALUES (value1, value2);
```

- Update: Dùng để cập nhật dữ liệu có trong bảng

sql

 Copy code

```
UPDATE table_name SET column1 = value1, column2 = value2 WHERE condition;
```

- Delete: Dùng để xóa dữ liệu trong bảng

sql

```
DELETE FROM table_name WHERE condition;
```

c. Ứng dụng

- SQL được sử dụng rộng rãi trong việc quản lý cơ sở dữ liệu quan hệ, truy vấn dữ liệu, thao tác dữ liệu và quản lý dữ liệu trong hệ thống thông tin.

d. Ưu nhược điểm

- Ưu
 - Linh hoạt và mạnh mẽ trong việc truy vấn dữ liệu từ cơ sở dữ liệu.
 - Chuẩn và có thể mở rộng cho nhiều hệ quản trị cơ sở dữ liệu khác nhau.
- Nhược
 - Cú pháp có thể phức tạp trong một số trường hợp.
 - Không thể sử dụng để thực hiện các tính toán phức tạp hay điều khiển luồng như một ngôn ngữ lập trình thông thường.

13. Scala

a. Lịch sử nguồn gốc:

- Scala là một ngôn ngữ lập trình đa mô hình, đa paradigms, được phát triển bởi Martin từ năm 2001.
- Scala được thiết kế để kết hợp tính năng của các ngôn ngữ lập trình hướng đối tượng và hàm, cũng như hỗ trợ tính toán song song.

b. Cú pháp cơ bản

- Khai báo biến

scala

```
var age: Int = 25
val name: String = "John"
```

- If else

scala

```
if (condition) {
    // Thực hiện khi condition đúng
} else {
    // Thực hiện khi condition sai
}
```

- While do

scala

```
while (condition) {
    // Thực hiện mã này trong khi condition đúng
}
```

- Function

scala

```
def greet(name: String): Unit = {  
    println(s"Hello, $name!")  
}
```

c. Ứng dụng

- Scala thường được sử dụng trong việc phát triển các ứng dụng phần mềm lớn, hệ thống đa luồng và công nghệ web.
- Nó được sử dụng rộng rãi trong cộng đồng công nghệ và thường được sử dụng trong hệ thống Apache Spark và Apache Kafka.

d. Ưu nhược điểm

- Ưu
 - Kết hợp các tính năng của lập trình hướng đối tượng và hàm.
 - Hỗ trợ tính toán song song một cách hiệu quả.
 - Cú pháp linh hoạt và mạnh mẽ.
- Nhược
 - Đôi khi cú pháp phức tạp và có thể đôi mắt với độ phức tạp cao đối với người mới học.
 - Hệ sinh thái có thể phong phú nhưng đôi khi có thể gây rối.

14. TypeScript

a. Lịch sử nguồn gốc:

- TypeScript là một ngôn ngữ lập trình mã nguồn mở do Microsoft phát triển, được giới thiệu lần đầu vào năm 2012.
- Nó mở rộng JavaScript bằng việc cung cấp kiểu dữ liệu tĩnh và các tính năng cải tiến, giúp tăng tính bảo mật và hiệu suất khi phát triển ứng dụng.

b. Cú pháp cơ bản

- Khai báo biến

typescript

```
let age: number = 25;
const name: string = "John";
```

- If else

typescript

```
if (condition) {
    // Thực hiện khi condition đúng
} else {
    // Thực hiện khi condition sai
}
```

- While do

typescript

```
while (condition) {
    // Thực hiện mã này trong khi condition đúng
}
```

- *TypeScript không hỗ trợ cú pháp do-while trực tiếp, nhưng bạn có thể sử dụng vòng lặp while để thực hiện tương tự.*

- Function

typescript

```
function greet(name: string): void {  
    console.log(`Hello, ${name}!`);  
}
```

c. Ứng dụng

- TypeScript thường được sử dụng trong việc phát triển các ứng dụng web và mobile, đặc biệt là các dự án lớn.
- Nó cũng được sử dụng rộng rãi trong việc phát triển các thư viện và framework JavaScript như Angular.

d. Ưu nhược điểm

- Ưu
 - Hỗ trợ kiểu dữ liệu tĩnh giúp tránh nhiều lỗi trong quá trình phát triển.
 - Nó mở rộng JavaScript, vì vậy có thể sử dụng những tính năng mới nhất của JavaScript.
 - Cung cấp tính năng lập trình hướng đối tượng mạnh mẽ.
- Nhược
 - Có thể cần thời gian để học và thích nghi với các tính năng kiểu dữ liệu tĩnh khi mới bắt đầu.
 - Có thể tạo ra các lỗi hoặc hạn chế nếu kiểu dữ liệu không được xác định một cách chính xác.

15. Haskell

a. Lịch sử nguồn gốc:

- Mục tiêu tạo ra một ngôn ngữ lập trình hàm hiện đại, có khả năng biểu diễn các khái niệm toán học một cách sáng tạo và có thể áp dụng vào các ứng dụng thực tế.
- Haskell được phát triển dựa trên các ngôn ngữ lập trình hàm khác như Miranda, ML và Scheme.
- Haskell được công bố lần đầu tiên vào năm 1990, và phiên bản chuẩn hiện tại là Haskell 2010.

b. Cú pháp cơ bản

- Khai báo biến

```
haskell

x = 5
name = "John"
```

- If else

```
haskell

if condition
  then expression1
  else expression2
```

- While do

- Haskell không có vòng lặp while hoặc do như các ngôn ngữ khác. Thay vào đó, nó thường sử dụng đệ quy để thực hiện các vòng lặp.

- Function


```
haskell
```

```
-- Tên hàm và tham số được đặt tên theo viết thường  
tên_hàm tham_số1 tham_số2 ... = biểu_thức
```

```
haskell
```

```
-- Hàm tính bình phương của một số nguyên  
square x = x * x  
  
-- Hàm tính tổng của hai số nguyên  
sum x y = x + y
```

c. Ứng dụng

- Haskell thường được sử dụng trong nghiên cứu và giáo dục, đặc biệt trong các lĩnh vực liên quan đến khoa học máy tính, toán học và logic.
- Nó cũng được áp dụng trong phát triển phần mềm thực tế, đặc biệt là trong các hệ thống cần tính toán cao và độ tin cậy.

d. Ưu nhược điểm

- Ưu
 - Cú pháp rõ ràng, dễ đọc, và dễ bảo trì.
 - Có kiểu dữ liệu tĩnh mạnh mẽ giúp tránh được nhiều lỗi thông thường trong quá trình phát triển.
- Nhược
 - Việc học và sử dụng đòi hỏi một cách tiếp cận mới với lập trình hàm và có thể gây khó khăn ban đầu cho người mới học.
 - Khó áp dụng trong các ứng dụng thực tế mà cần sự linh hoạt cao trong thao tác với dữ liệu như các ngôn ngữ lập trình khác.

CHƯƠNG 3. KẾT LUẬN

3.1. Kết quả đạt được

Kết thúc bài báo cáo, qua đề tài này nhóm chúng em đã giới thiệu và cho giảng viên và các bạn cùng khoá biết thêm những kiến thức về các ngôn ngữ lập trình. Đây sẽ là một hành trang, một bước khởi đầu trong chặng đường đại học cũng như việc lựa chọn ngôn ngữ lập trình sao cho phù hợp về tính chất công việc của mình. Bài báo cáo đã giúp chúng em có thêm tài liệu bổ ích và sự độc đáo của mỗi ngôn ngữ lập trình. Nó là 1 trợ thủ đắc lực trong việc chọn ngôn ngữ và định hướng nghề nghiệp cho em cũng như tất cả các bạn khoá K18 – Trường Đại học Công nghệ Thông tin – ĐHQG Thành phố Hồ Chí Minh. Quá trình làm bài báo cáo đã cho chúng em thêm nhiều kinh nghiệm trong cách làm việc nhóm cũng như lắng nghe tìm tiếng nói chung ở cả nhóm để hướng tới một bài báo cáo hiệu quả, bổ ích cho mọi người. Cuối lời, xin gửi tới giảng viên một lời cảm ơn sâu sắc vì đã hướng dẫn tận tình cho chúng em. Nhóm chúng em xin cảm ơn giảng viên đã đọc bài báo cáo này.

3.1. Bảng phân công công việc

Họ và tên	Phân công nhiệm vụ
Tiêu Hoàng Phúc	<ul style="list-style-type: none">Thực hiện giới thiệu các ngôn ngữ lập trình Python, Java, C#.Tổng hợp nội dung, Tài liệu tham khảo.
Võ Sĩ Trí Thông	<ul style="list-style-type: none">Thực hiện giới thiệu các ngôn ngữ lập trình Ruby, Scala, TypeScript.Làm phần Kết quả đạt được.
Kim Thành Tiến	<ul style="list-style-type: none">Thực hiện giới thiệu các ngôn ngữ lập trình PHP, Swift, Go.Làm phần Lời cảm ơn, Lời mở đầu.
Từ Minh Khoa	<ul style="list-style-type: none">Thực hiện giới thiệu các ngôn ngữ lập trình JavaScript, SQL, Haskell.Làm phần Giới thiệu yêu cầu.
Lữ Nguyễn Duy Đức	<ul style="list-style-type: none">Thực hiện giới thiệu các ngôn ngữ lập trình C++, Kotlin, Rust.Làm phần Mục lục, bảng phân công.

TÀI LIỆU THAM KHẢO

- [1] CareerBuilder, “C là gì? Tổng hợp kiến thức học C cơ bản cực dễ hiểu.” [Trực tuyến]. Có sẵn tại: <https://careerbuilder.vn/vi/talentcommunity/c-la-gi-tong-hop-kien-thuc-hoc-c-co-ban-cuc-de-hieu.35A520A9.html>. [Truy cập: 04-Jul-2025].
- [2] Viblo, “Ngôn ngữ Haskell.” [Trực tuyến]. Có sẵn tại: <https://viblo.asia/p/ngon-ngu-haskell-RnB5pykwKPG>. [Truy cập: 04-Jul-2025].
- [3] Glints Vietnam, “Ngôn ngữ lập trình Python là gì?” [Trực tuyến]. Có sẵn tại: <https://glints.com/vn/blog/ngon-ngu-lap-trinh-python-la-gi/>. [Truy cập: 04-Jul-2025].
- [4] DTech VN, “C là gì? Tổng quan về C.” [Trực tuyến]. Có sẵn tại: <https://www.dtechvn.com/c-la-gi-tong-quan-ve-c.html>. [Truy cập: 04-Jul-2025].
- [5] Lập trình căn bản, “Cấu trúc chương trình Java.” [Trực tuyến]. Có sẵn tại: <https://laptrinhcanban.com/java/java-co-ban-cho-nguoi-moi-bat-dau/kien-thuc-co-ban-ve-java/cau-truc-chuong-trinh-java/>. [Truy cập: 04-Jul-2025].
- [6] Tenten, “JavaScript là gì?” [Trực tuyến]. Có sẵn tại: <https://tenten.vn/tin-tuc/javascript-la-gi/>. [Truy cập: 04-Jul-2025].
- [7] TopDev, “TypeScript là gì?” [Trực tuyến]. Có sẵn tại: <https://topdev.vn/blog/typescript-la-gi/>. [Truy cập: 04-Jul-2025].
- [8] Wikipedia, “Go (ngôn ngữ lập trình).” [Trực tuyến]. Có sẵn tại: [https://vi.wikipedia.org/wiki/Go_\(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%A0Dp_tr%C3%ACnh\)](https://vi.wikipedia.org/wiki/Go_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%A0Dp_tr%C3%ACnh)). [Truy cập: 04-Jul-2025].
- [9] Vietnix, “Ngôn ngữ Ruby.” [Trực tuyến]. Có sẵn tại: <https://vietnix.vn/ngon-ngu-ruby/>. [Truy cập: 04-Jul-2025].