

Đề xuất sử dụng ngôn ngữ lập trình Python thay thế cho C/C++

HUỲNH ĐỆ THỦ (*)

Trường Đại học Kinh tế - Tài chính TP.HCM
Nhận bài: 23/04/2022 - Duyệt đăng: 15/06/2022

(*) Liên hệ: thuhd@uef.edu.vn - Tel: 0907878385

Tóm tắt:

Hiện nay, Trường Đại học Kinh tế - Tài chính TP.HCM cũng như nhiều trường Đại học tại Việt Nam và trên thế giới thường sử dụng ngôn ngữ truyền thống C/C++ để giảng dạy cho sinh viên nhóm ngành Công nghệ thông tin. Tuy nhiên, cú pháp phức tạp của nó là một thách thức cho người mới bắt đầu, gây trở ngại cho việc học của họ. Thay vào đó, Python là một ngôn ngữ có cú pháp tương đối đơn giản, dễ hiểu đồng thời hỗ trợ các cấu trúc dữ liệu, thư viện lớn, giúp cho chương trình được viết bằng Python trở nên ngắn gọn. Vì vậy, Python là một lựa chọn nên được xem xét để thay thế cho C/C++. Trong bài báo này, chúng tôi tập trung phân tích các tính năng của ngôn ngữ Python so sánh với C/C++ trên nhiều khía cạnh khác nhau như biên dịch và thực thi, biến, cấu trúc điều khiển, cấu trúc dữ liệu... Bài báo cũng trình bày mức độ phổ biến của Python và lĩnh vực ứng dụng của nó trong thực tế. Trên cơ sở đó, chúng tôi đề xuất về việc sử dụng ngôn ngữ lập trình Python thay thế cho C/C++ trong chương trình đào tạo thuộc nhóm ngành Công nghệ thông tin tại Trường Đại học Kinh tế - Tài chính TP.HCM.

Từ khóa: Python, C/C++, ngôn ngữ lập trình.

Abstract:

Currently, the Ho Chi Minh City University of Economics and Finance as well as many universities in Vietnam and around the world often used the traditional C/C++ language to teach for students of the information technology majors. However, its complicated syntax is a challenge for beginners. Instead, Python is the language with relatively simple syntax, easy to understand, supports large data structures and libraries, making programs written in Python become concise. Therefore, Python is an option that should be considered as an alternative to C/C++. In this paper, we focus on analyzing the features of Python language compared with C/C++ on many different aspects such as compilation and execution, variables, control structures, data structures. The paper also presents the popularity of Python and its field of application in practice. We then propose to using the Python programming language as an alternative to C/C++ in the teaching program of the information technology majors at the Ho Chi Minh City University of Economics and Finance.

Keywords: Python, C/C++, Programming language.

1. Giới thiệu

Lập trình là một kỹ năng cơ bản và quan trọng của sinh viên thuộc nhóm ngành Công nghệ thông tin tại Trường Đại học Kinh tế - Tài chính TP.HCM. Tuy nhiên, kỹ năng này là một thử thách lớn đối với những người mới bắt đầu và thậm chí là với

nhiều sinh viên đã ra trường. Bởi vì nó bao gồm nhiều khái niệm phức tạp và kỹ năng tư duy giải quyết bài toán bằng máy tính. Sinh viên mới bắt đầu thường khó có thể hiểu và áp dụng được các kiến thức lập trình một cách đúng đắn (Lo và cộng sự, 2015). Sinh viên được dạy các cấu trúc

ngữ pháp nhưng không biết làm thế nào kết hợp chúng với nhau để giải quyết một bài toán cụ thể. Sinh viên luôn gặp trở ngại trong việc cấu trúc một chương trình sao cho hợp lý để ra được kết quả. Bên cạnh đó, một vấn đề mà hầu hết những sinh viên mới bắt đầu thường gặp phải là phạm

rất nhiều lỗi liên quan đến cú pháp. Điều này làm giảm sự hứng thú của sinh viên đối với môn học lập trình. Vì vậy, ngôn ngữ được lựa chọn để bắt đầu việc giảng dạy lập trình cho sinh viên là đặc biệt quan trọng.

C/C++ là ngôn ngữ lập trình truyền thống được Trường Đại học Kinh tế - Tài chính TP.HCM lựa chọn. Đây là một ngôn ngữ rất phổ biến, cung cấp cho sinh viên nhiều khái niệm căn bản về lập trình. Tuy nhiên, cú pháp của nó tương đối chặt chẽ, phức tạp khiến cho sinh viên gặp rất nhiều lỗi cú pháp và mất nhiều thời gian cho việc sửa lỗi. Với sinh viên mới bắt đầu, họ nên tập trung học cách phân tích và chia nhỏ bài toán hơn là luôn lo lắng quá nhiều về cú pháp (Lo và cộng sự, 2015). Điều mà sinh viên nên được tiếp cận đầu tiên là các nguyên lý lập trình cơ bản và phương pháp giải quyết bài toán chứ không phải là những đặc điểm của một ngôn ngữ cụ thể (Ahmed và cộng sự, 2021). Cho nên, ngôn ngữ dùng để minh họa cần phải đơn giản, giúp cho sinh viên dễ dàng thực hiện các tác vụ. Ngôn ngữ Python với cú pháp đơn giản, dễ hiểu, dễ đọc, dễ viết, ngắn gọn và nhận kết quả nhanh chóng là một lựa chọn tuyệt vời cho họ. Nó là ngôn ngữ động, hỗ trợ lập trình hướng đối tượng và hướng chức năng, được thiết kế gần gũi với ngôn ngữ tự nhiên, thân thiện với người dùng.

Ra mắt vào năm 1991, Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra (Nagpal và cộng sự, 2019). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là một dự án mã nguồn mở và được cải tiến liên tục từ cộng đồng lớn. Nó được chất lọc từ các tính năng tốt nhất của các ngôn ngữ lập trình khác. Python cung cấp tất cả các công cụ và tính năng cơ bản mà các lập trình viên mong đợi.

2. So sánh ngôn ngữ lập trình Python với C/C++

Trong phần này, chúng tôi thực hiện so sánh ngôn ngữ lập trình Python với C/C++ trên một số khía cạnh quan trọng như cấu trúc chương trình, biên dịch và thực thi, thụt lùi, biến và kiểu dữ liệu, cấu trúc điều khiển, cấu trúc dữ liệu.

2.1. Cấu trúc chương trình

Sự khác biệt lớn nhất giữa Python và C/C++ là cấu trúc cơ bản của một chương trình. Bắt đầu từ những chương trình “Hello World”, sinh viên học ngôn ngữ C/C++ phải viết những thứ mà họ thực

Bảng 1. So sánh 2 chương trình của Python và C/C++

Ngôn ngữ C/C++	Ngôn ngữ Python
<pre>#include<stdio.h> int main (){ printf("Hello, world!"); return 0; }</pre>	<pre>print "Hello, world!"</pre>

Nguồn: Tác giả

sự không hiểu. Cấu trúc cơ bản của một chương trình C/C++ phải chứa hàm “main” và các lệnh cần thiết để giải quyết tác vụ được đặt trong hàm này. Còn với Python, sinh viên có thể bắt đầu ngay với từng lệnh đơn mà không cần phải viết hàm “main”. Bảng 1 là ví dụ minh họa về một tác vụ đơn giản được viết bởi C/C++ và Python.

Qua ví dụ trên ta dễ dàng nhận thấy sự ngắn gọn, súc tích và dễ hiểu của Python so với C/C++. Chỉ cần một dòng lệnh duy nhất thay cho 6 dòng gồm khai báo thư viện và viết hàm “main”. Khi viết hàm “main” cần phải tuân thủ theo đúng nguyên mẫu của hàm gồm giá trị trả về, tên hàm, cặp dấu () và trong thân hàm phải có lệnh “return 0;”. Tất cả những điều này đều gây khó hiểu cho sinh viên và rất nhiều sinh viên phạm lỗi liên quan đến chúng. Vì họ chưa thể nhớ và hiểu hết được những từ khóa cũng như cú pháp. Khi viết những chương trình dài hơn, phức tạp hơn, hầu hết các sinh viên đều gặp phải rào cản về số lượng lỗi cú pháp và tốn rất nhiều thời gian để sửa lỗi. Và đối với họ, hàng loạt thông báo lỗi hiển thị khi biên dịch gây rối rắm và thậm chí là không biết lỗi ở đâu, sửa như thế nào.

2.2. Biên dịch và thực thi

Ngôn ngữ C/C++ sử dụng trình biên dịch để biên dịch chương trình, vì thế mọi chương trình C/C++ đều phải trải qua một chu trình “viết mã – biên dịch – thực thi” (Nagpal và cộng sự, 2019). Sinh viên phải viết xong một chương trình hoàn chỉnh, biên dịch toàn bộ chương trình và sửa hết các lỗi biên dịch trước khi thực thi. Còn Python là một ngôn ngữ thông dịch cho phép dịch và thực thi theo từng dòng lệnh (Nagpal và cộng sự, 2019). Sinh viên biết được ngay lỗi ở đâu và giải quyết nó thay vì phải thấy bối rối như cách biên dịch ở ngôn ngữ C/C++. Quá trình dịch và thực thi được tích hợp với nhau, nên hầu như sinh viên sẽ không có khái niệm về biên dịch. Họ chỉ việc viết mã, chạy và nhận kết quả. Nếu dòng mã lệnh bị lỗi thì kết quả sẽ là một thông báo lỗi. Các thông báo lỗi tương đối rõ ràng

và đặc biệt nếu sử dụng Jupyter Notebook để lập trình Python, thông báo lỗi sẽ chỉ ra cụ thể vị trí sai ở từng câu lệnh. Điều này giúp cho quá trình sửa lỗi dễ dàng và nhanh chóng. Hơn nữa, việc thực thi ngay từng lệnh hỗ trợ kiểm tra tính chính xác của mỗi dòng lệnh, đồng thời nó là kết quả trung gian giúp hạn chế lỗi thực thi của toàn bộ chương trình.

Bên cạnh đó, việc biên dịch toàn bộ chương trình cũng làm lộ ra nhược điểm trong tình huống sửa lại một số dòng mã lệnh sau khi đã biên dịch hoặc thực thi trước đó. Quá trình biên dịch phải thực hiện lại từ đầu bao gồm những dòng mã lệnh không có sự thay đổi. Đây chính là mấu chốt làm chậm quá trình thực thi chương trình.

2.3. Thụt lùi, khối lệnh và kết thúc lệnh

Đối với những ngôn ngữ như C/C++ thì khoảng thụt lùi chỉ dùng để định dạng cấu trúc các dòng lệnh cho dễ đọc hơn, hạn chế sai sót. Tuy nhiên, trong Python thì một khoảng thụt lùi rất quan trọng. Nó được dùng để tạo ra một khối lệnh và bắt buộc kể cả khi chỉ có một lệnh nằm trong thân của lệnh khác. Một khối lệnh sẽ kết thúc khi kết thúc thụt lùi. Ngôn ngữ C/C++ hoặc một số ngôn ngữ khác thường sử dụng cặp dấu “{” và “}” để đánh dấu sự bắt đầu và kết thúc của một khối lệnh (Nagpal và cộng sự, 2019). Đây cũng là một khó khăn nhỏ đối với người mới bắt đầu khi thường xuyên quên dấu đóng. Ví dụ minh họa sử dụng khoảng thụt lùi giữa 2 ngôn ngữ:

Trong Python, khoảng thụt lùi của lệnh “if” và “print” trong ví dụ trên là bắt buộc. Không thụt lùi hoặc thụt lùi không chính xác có thể sẽ sinh ra một thông báo lỗi về luồng logic của chương trình. Dòng lệnh `j++`; ở ngôn ngữ C/C++ mặc dù viết thụt lùi nhưng không có ý nghĩa gì cả, nó là một lệnh độc lập nằm ngoài vòng for. Tuy nhiên, lệnh tăng `j` tương đương ở Python được thụt lùi mang ý nghĩa nằm trong thân của vòng for sẽ khác với không thụt lùi. Việc bắt buộc sử dụng khoảng trắng thụt lùi đối với các lệnh con trong Python vừa tạo sự rõ ràng trong cấu trúc chương trình, làm cho nó trở nên dễ đọc (Ahmed và cộng sự, 2021), vừa giảm thiểu việc sử dụng thêm cặp dấu {}. Ngoài ra, tiêu chí của Python là tối giản mã lệnh, vì thế dấu ‘;’ để kết thúc mỗi dòng lệnh như trong C/C++ được lược bỏ. Những điều này giúp hạn chế bớt các lỗi thiếu dấu đóng hay lỗi quên kết thúc dòng lệnh.

Bảng 2. Ví dụ khoảng thụt lùi của Python và C/C++

Ngôn ngữ C/C++	Ngôn ngữ Python
<pre>#include<stdio.h> int main () { int i,j = 0; for(i = 0; i < 5; i++) if(i%2 == 0) printf("%d", i); j++; return 0; }</pre>	<pre>j = 0 for i in range(5): if i%2 == 0: print(i) j += 1</pre>

Nguồn: Tác giả

2.4. Kiểu dữ liệu, biến và quản lý bộ nhớ

Biến và kiểu dữ liệu là hai thành phần cơ bản của mọi ngôn ngữ lập trình. Trong ngôn ngữ C/C++, một biến phải được khai báo kiểu dữ liệu tường minh trước khi sử dụng. Ngược lại, trong Python lại áp dụng kiểu động, tức là không cần khai báo (Ahmed và cộng sự, 2021). Trình thông dịch của Python sẽ tự động phân tích giá trị gán cho biến để xác định kiểu dữ liệu của nó. Một ưu điểm nổi bật của Python so với C/C++ đó là khả năng cho phép gán nhiều giá trị thuộc các kiểu khác nhau cho cùng một biến trong cùng một phạm vi, điều này không được phép trong C/C++. Chẳng hạn như, những câu lệnh gán sau đây là hoàn toàn hợp lệ trong Python:

```
a = 1 # type(a) là int
a = 1.5 # type(a) là float
a = "abc" # type(a) là str
```

Ban đầu, `a` được gán giá trị là 1 có nghĩa là nó đang tham chiếu đến một đối tượng số nguyên. Khi biến `a` được gán 1.5 thì lúc đó nó đã tham chiếu đến một đối tượng khác là số thực. Qua ví dụ, ta nhận thấy rõ tính linh hoạt trong sử dụng biến và kiểu dữ liệu của Python. Trong khi điều này lại gây ra nhiều vấn đề trong C/C++, đặc biệt là người mới bắt đầu thường mắc lỗi khi chọn và khai báo kiểu dữ liệu sao cho đúng. Bên cạnh đó, Python còn hỗ trợ bộ tự động thu hồi bộ nhớ đối với những biến không còn được sử dụng nữa, tránh lãng phí bộ nhớ. Tính năng này không có trong ngôn ngữ C/C++.

2.5. Cấu trúc điều khiển

Khác với ngôn ngữ C/C++, Python chỉ hỗ trợ các cấu trúc điều khiển bao gồm if, for và while. Tuy nhiên, chức năng của cấu trúc điều khiển khác ví như do...while đều có thể thực hiện trong Python. Điểm khác biệt lớn nhất trong các cấu trúc điều khiển, Python không yêu cầu cặp dấu () bao quanh điều kiện. Ngoài ra, các điều kiện được nối

Bảng 3. Ví dụ minh họa sử dụng vòng for

Ngôn ngữ C/C++	Ngôn ngữ Python
<pre>#include<stdio.h> int main () { int a[3] = {1,2,3}; int b[3], j; for(i = 0, j = 0; i < 3; i++) if(a[i]%2 == 0) b[j++] = a[i]; return 0; }</pre>	<pre>a = [1,2,3] b = [i for i in a if i%2 == 0]</pre>

Nguồn: Tác giả

với nhau bằng các toán tử “and” hoặc “or” thay thế cho các dấu toán tử thường thấy trong C/C++, hoặc sử dụng “in” trong vòng for, rất gần gũi với ngôn ngữ tự nhiên, tạo nên sự dễ hiểu cho người viết và người đọc mã lệnh.

Để thực hiện yêu cầu trích ra một mảng gồm các phần tử chẵn của mảng a, ta nhận thấy rõ Python là một ngôn ngữ rất mạnh khi thực hiện cùng một yêu cầu chỉ với 2 dòng lệnh đơn giản. Rõ ràng lệnh for trong C/C++ phức tạp hơn nhiều, nó thường gây ra lỗi vòng lặp vô hạn, lỗi cú pháp về dấu ‘,’ hay ‘;’ hoặc thực hiện sai nếu sinh viên không nắm được thứ tự thực hiện các biểu thức trong for. Đồng thời, khi viết điều kiện lặp trong for thường dẫn đến lỗi liên quan đến vượt ngoài phạm vi mảng.

2.6. Cấu trúc dữ liệu

Ngôn ngữ Python cung cấp cho người lập trình rất nhiều những cấu trúc dữ liệu được xây dựng sẵn gồm list, tuple, set và dictionary. Thao tác sử dụng các cấu trúc này rất linh hoạt, ví dụ như list có thể chứa những phần tử thuộc các kiểu dữ liệu khác nhau hay cách truy xuất phần tử, trích xuất danh sách con rất dễ dàng. Ngoài ra, còn có hai thư viện đặc biệt quan trọng trong Python đó là Numpy và Pandas cũng cung cấp thêm các cấu trúc dữ liệu hữu ích hỗ trợ tốt cho việc tổ chức lưu trữ, xử lý với bộ dữ liệu lớn và có cấu trúc phức tạp. Numpy cung cấp cấu trúc dữ liệu ndarray để làm việc với mảng nhiều chiều, trong khi đó Pandas lại cung cấp 2 cấu trúc dữ liệu là Series và DataFrame. Đây là những cấu trúc dữ liệu rất hữu ích dành cho các nhà khoa học dữ liệu và được xem như là thế mạnh của Python so với các ngôn ngữ khác. Mặc dù, C/C++ cũng hỗ trợ khá nhiều cấu trúc dữ liệu khác nhau có trong thư viện STL, tuy nhiên nó khá phức tạp và ít linh hoạt.

3. Ứng dụng của ngôn ngữ lập trình Python

Hiện nay Python được sử dụng rộng rãi trong nhiều lĩnh vực (YK Sugi, 2022). Có thể kể đến các lĩnh vực như:

Phân tích dữ liệu: Python phổ biến cho việc phân tích dữ liệu vì các thư viện mạnh mẽ như Numpy và Pandas, giúp các tác vụ phân tích và dọn dẹp dữ liệu tương đối đơn giản, ngay cả khi làm việc với các bộ dữ liệu lớn. Ngoài ra còn có các thư viện Python hỗ trợ nhiều tác vụ phân tích dữ liệu khác nhau, từ khai thác dữ liệu web với BeautifulSoup đến hiển thị dữ liệu bằng Matplotlib.

Khoa học dữ liệu/AI: Python cũng cực kỳ phổ biến cho các tác vụ liên quan đến dữ liệu nâng cao trong lĩnh vực máy học. Các thư viện mạnh mẽ như Scikit-learn và TensorFlow giúp việc triển khai các thuật toán học máy phổ biến trở nên đơn giản. Đồng thời nhiều thư viện chuyên biệt giúp thực hiện nhiều tác vụ học máy từ nhận dạng hình ảnh đến tạo nội dung một cách dễ dàng.

Phát triển web: Python cũng được sử dụng để phát triển ứng dụng web. Nhiều trang web được xây dựng bằng Python và các Python framework phổ biến như Django và Flask. Mặc dù bản thân các trang được hiển thị bằng HTML và CSS, Python làm nền tảng cho các yếu tố trực quan này trên nhiều trang web, điều khiển chức năng, quản lý cơ sở dữ liệu, tài khoản người dùng và hơn thế nữa.

Phát triển trò chơi: Python được sử dụng trong việc phát triển các trò chơi điện tử độc lập, nhờ vào sự tồn tại của các thư viện tiện lợi như PyGame.

Phát triển phần mềm: Python được sử dụng rộng rãi trong phát triển phần mềm, trên nhiều ứng dụng trong thực tế. Ngày nay ranh giới giữa phát triển phần mềm và phát triển web hơi mờ nhạt vì hầu như tất cả phần mềm đều được xây dựng để hoạt động trên web ngay cả khi có cả ứng dụng dành cho máy tính để bàn. Dropbox là một ví dụ điển hình về một công ty phát triển phần mềm hiện đại làm cả hai và Python được sử dụng để xây dựng ứng dụng dành cho máy tính để bàn của Dropbox. Tương tự, Spotify có cả ứng dụng web và máy tính để bàn và Python được sử dụng để xây dựng một số dịch vụ nền giúp chúng hoạt động. Tất nhiên, Python cũng được sử dụng tại nhiều công ty để phát triển các phần mềm nội bộ.

Kỹ thuật Robotics: Python là một ngôn ngữ phổ biến trong lĩnh vực chế tạo người máy, cả chuyên

nghiệp và không chuyên. Với những người chỉ làm vì sở thích, Python thường được sử dụng cùng với nền tảng phần cứng Raspberry Pi, cho phép thử nghiệm linh hoạt với giá cả phải chăng. Trong kinh doanh, Python là một trong những ngôn ngữ thường được sử dụng để tự động hóa quy trình bằng rô bốt (robotic process automation – RPA) và nó được sử dụng để làm những việc như các cánh tay rô bốt công nghiệp được lập trình để có thể hoạt động song song với nhau.

Tự động hóa: Python tuyệt vời để tự động hóa các tác vụ lặp đi lặp lại và có gần như vô tận các trường hợp sử dụng trong thực tế để tự động hóa Python. Ví dụ: Python là một công cụ phổ biến trong DevOps vì nó giúp tự động hóa các hệ thống và quy trình hiệu quả và minh bạch. Nhưng bên ngoài lĩnh vực phát triển phần mềm, nó cũng được sử dụng rộng rãi để tự động hóa mọi thứ, từ các hệ thống phức tạp đến các quy trình đơn giản, cá nhân như điền vào bảng tính hoặc trả lời email.

Giao diện và điều khiển phần cứng: Khả năng điều khiển phần cứng của Python vượt ra ngoài lĩnh vực robotics. Trên thực tế, Python được sử dụng trong tất cả các loại ứng dụng điều khiển phần cứng trong thế giới thực.

Giáo dục và đào tạo: Bởi vì Python là một ngôn ngữ có thể đọc được (readable) rất cao và cũng có nhiều mục đích sử dụng thực tế khác nhau, nên Python là ngôn ngữ được lựa chọn đầu tiên cho những người muốn học lập trình. Nhiều hướng dẫn Python, video, các khóa học tương tác và các tài liệu giáo dục khác có sẵn cho Python khiến nó được cho là ngôn ngữ lập trình dễ học nhất.

Tiện ích cá nhân: Python có thể được sử dụng để phân tích dữ liệu của riêng bạn, để tự động hóa các yếu tố nhàm chán hoặc lặp đi lặp lại trong công việc của bạn, hoặc thậm chí để tạo ra tác phẩm nghệ thuật.

Phát triển ứng dụng di động: Python cũng có thể được sử dụng để phát triển ứng dụng di động với Kivy và BeeWare framework.

4. Mức độ phổ biến của ngôn ngữ lập trình Python

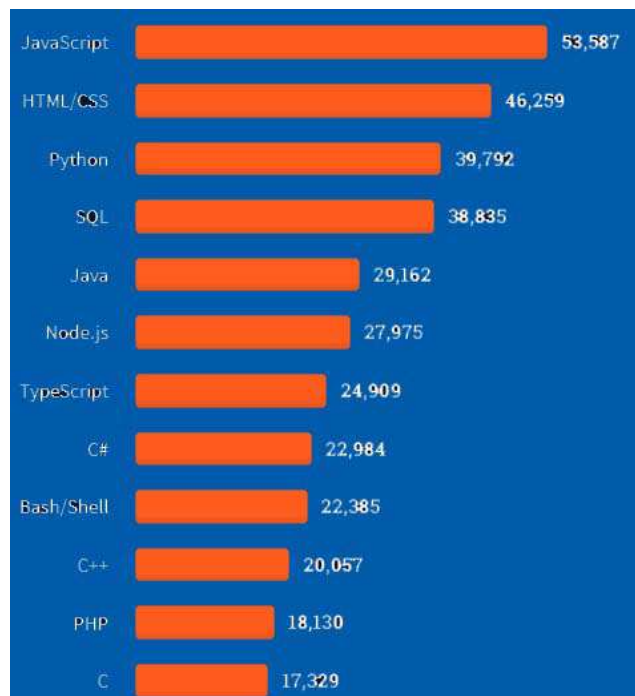
Theo báo cáo năm 2021 của Stack Overflow, một trong những cộng đồng lập trình viên lớn nhất thế giới hiện nay. Sau khi tiến hành khảo sát 83,052 lập trình viên về việc sử dụng ngôn ngữ lập trình hoặc tương tự ngôn ngữ lập trình. Kết quả thống kê tỉ lệ sử dụng các ngôn ngữ lập trình được thể hiện qua

hình 1. Năm 2021, Python đã vượt qua hàng loạt ngôn ngữ lập trình, xếp hạng 3 về mức độ phổ biến với 39,792 lập trình viên sử dụng (Stackoverflow, 2022).

5. Kết luận và kiến nghị

Trong bài báo này, chúng tôi đã thực hiện phân tích những tính năng nổi bật của ngôn ngữ lập trình Python so với C/C++. Trên cơ sở đó, chúng tôi đề xuất về việc sử dụng ngôn ngữ lập trình Python thay thế cho C/C++ trong chương trình đào tạo thuộc nhóm ngành Công nghệ thông tin tại Trường Đại học Kinh tế - Tài chính TP.HCM. Với cú pháp ngắn gọn, dễ hiểu, linh hoạt kết hợp với các cấu trúc dữ liệu đa dạng sẽ làm hạn chế nhiều lỗi thường gặp giúp cho sinh viên có một sự khởi đầu dễ dàng, tạo ra sự hứng thú đối với môn học. Chương trình được viết bằng Python có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Mac, Linux. Hơn nữa, Python là một ngôn ngữ mã nguồn mở với thư viện lớn, lĩnh vực ứng dụng đa dạng, cộng đồng hỗ trợ rộng lớn góp phần làm cho Python ngày càng được sử dụng phổ biến.

Hình 1. Khảo sát mức độ sử dụng của các ngôn ngữ lập trình



Nguồn: Stackoverflow, 2022

(Xem tiếp trang 132)