Bài: Phương thức main trong Java

Xem bài học trên website để ủng hộ Kteam: Phương thức main trong Java

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage How Kteam nhé!

Dẫn nhập

Ở bài trước, chúng ta đã tìm hiểu INTERFACE TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG. Hôm nay, Kteam sẽ giải thích rõ cấu trúc của **chương trình main**.

Nội dung

Để đọc hiểu bài này, tốt nhất các bạn nên có kiến thức cơ bản về các phần sau:

- CÁC BIẾN TRONG JAVA.
- CÁC KIỂU DỮ LIỆU TRONG JAVA.
- CÁC HẠNG TOÁN TỬ TRONG JAVA
- CẤU TRÚC RẼ NHÁNH TRONG JAVA
- VÒNG LẶP WHILE TRONG JAVA
- VÒNG LẶP FOR TRONG JAVA
- MẢNG TRONG JAVA
- VÒNG LẶP FOR-EACH TRONG JAVA
- VAI TRÒ BREAK, CONTINUE TRONG VÒNG LẶP JAVA
- SWITCH TRONG JAVA
- LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- CLASS TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- CÁC LOẠI PHẠM VI TRUY CẬP TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- TỪ KHÓA STATIC TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- TỪ KHÓA THIS TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- THỪA KẾ TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- SETTER & GETTER TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- OVERRIDING VÀ OVERLOADING TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- TÍNH TRỪU TƯỢNG TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG
- INTERFACE TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Bài này chúng ta sẽ tìm hiểu những vấn đề sau:

- Cú pháp main. Giải thích

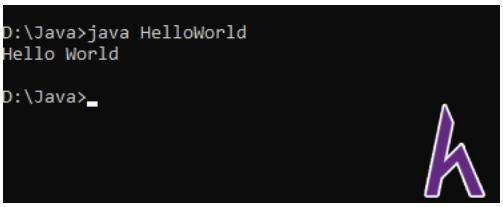
Cú pháp main? Giải thích

Khi khai báo phương thức main ta thường viết như sau:

```
public static void main(String[] args) {

}
```

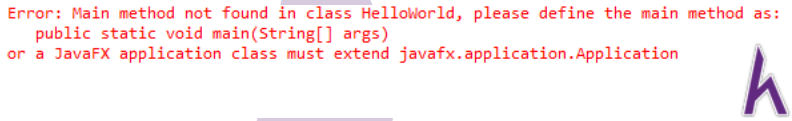
Ý nghĩa của phương thức này là khi ta dùng lệnh java gọi đến lớp nào đó, Java VM sẽ cố gắng tìm phương thức main trong lớp. Quay lại những bài đầu tiên, ta dùng CMD gõ lệnh như sau:



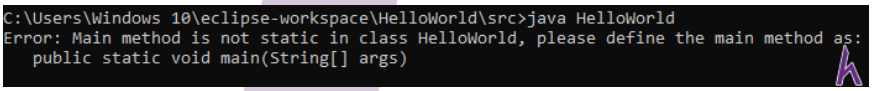
Như ta viết **java Helloworld** thì Java VM sẽ tìm phương thức main trong lớp HelloWorld để chạy chương trình. Nếu ta muốn viết main trong lớp khác cũng tương tự.

Ý nghĩa từng cú pháp của phương thức Main:

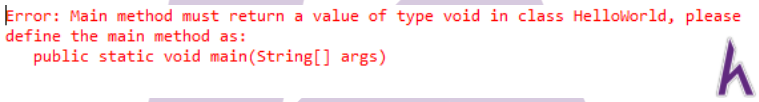
- **public**: Ta phải để quyền truy cập ở dạng **public** để JRE ở bên ngoài có thể truy cập được phương thức để thực thi. Nếu ta không có từ khóa **public** thì chương trình không thể tìm được như sau:



- **static**: Khi JRE bắt đầu, chưa có đối tượng nào được khởi tạo. Vì vậy ta nên để phương thức ở dạng **static** để JVM có thể load class vào bộ nhớ và có thể gọi phương thức. Nếu không có **static** thì có lỗi như sau:



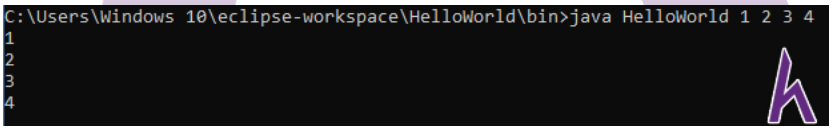
- **void**: phương thức main bắt buộc là **void**, khi bạn thử **return** giá trị khác thì Eclipse sẽ cảnh báo:



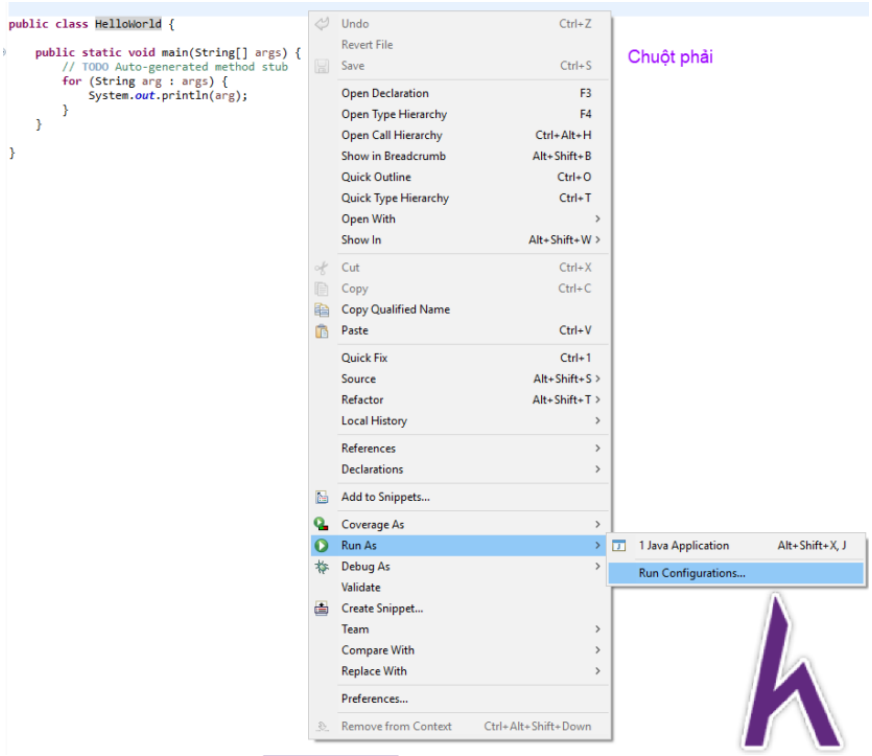
- **String[] args**: Đây là những tham số mà người dùng truyền vào sau tên lớp. Mặc định những giá trị người dùng sẽ nằm trong một mảng kiểu String tên **args**. Ta sẽ thử in ra như sau:

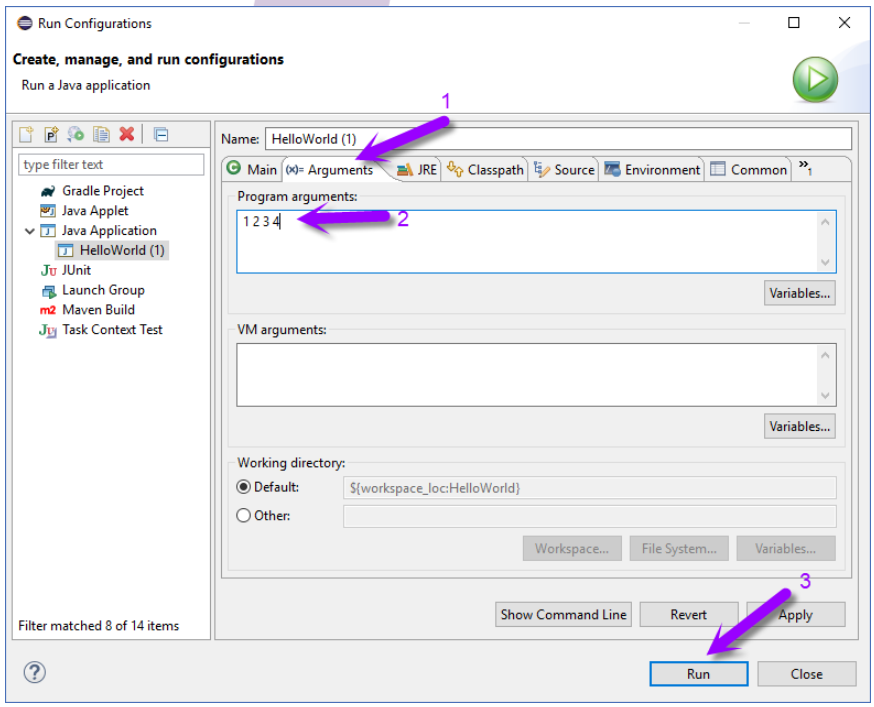
java:

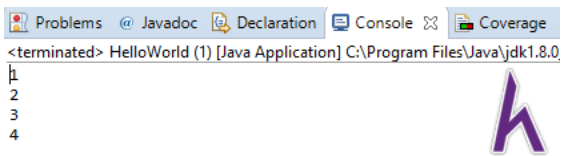
```
public static void main(String[] args) {
        // TODO Auto-generated method stub
        for (String arg : args) {
                System.out.println(arg);
        }
}
```



Nếu dùng Eclipse ta sẽ thêm các tham số như sau:







Kết

Như vậy chúng ta đã tìm hiểu phương thức main trong JAVA

Ở bài sau, Kteam sẽ giới thiệu đến bạn về TRY CATCH TRONG JAVA

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên "**Luyện tập – Thử thách – Không ngại khó**".