Java Fundamentals: Main Method, Imports, and More

This repository provides an overview and detailed explanations of key Java concepts, including the `main()` method, import statements, and file naming conventions. Whether you're new to Java or refreshing your knowledge, this guide will walk you through the basics.

Table of Contents

The `main()` Method

In Java, the `main()` method is the entry point of any application. It is the method that the JVM invokes to start program execution.

What Does `main()` Mean?

- The `main()` method is always the first method the JVM calls when executing a Java program.
- It must have a specific signature:

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

- The `String[] args` allows the program to accept command-line arguments.

Signature of the `main()` Method

- `public` : Accessible from anywhere.
- `static` : Belongs to the class itself (no instance needed).
- `void` : The method does not return any value.
- `String[] args` : An array of `String` that holds command-line arguments.

Example: `main()` Method

```
public class Main {
    public static void main(String[] args) {
        System.out.println("Hello, World!");  // Prints a message
        if (args.length > 0) {
            System.out.println("Command-line argument: " + args[0]);  // Prints command-line a
        }
    }
}
```

Import Statements

Java uses import statements to bring in classes and packages that are not part of the current class.

Types of Imports

1. **Single Class Import**
   - Allows importing a single class.
   - **Syntax**: `import packageName.ClassName;`

   ```
   import java.util.Scanner;  // Importing Scanner class
   ```

2. **Wildcard Import**
   - Imports all classes from a specified package.
   - **Syntax**: `import packageName.*;`

   ```
   import java.util.*;  // Import all classes from java.util package
   ```

3. **Static Import**

   - Imports static members (fields and methods) of a class.
   - **Syntax**: `import static packageName.ClassName.memberName;`

   ```
   import static java.lang.Math.PI;   // Importing static field PI
   import static java.lang.Math.sqrt; // Importing static method sqrt()
   ```

Example: Import Statements

```
import java.util.Scanner;  // Import Scanner class
import static java.lang.Math.PI;  // Import static PI

public class Example {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter your name:");
        String name = scanner.nextLine();
        System.out.println("Hello, " + name + "!");
        System.out.println("The value of PI is: " + PI);
    }
}
```

File Naming Conventions

In Java, the filename must follow certain conventions to ensure proper structure and readability.

Class Name & File Name Matching

- **The file name should match the class name** if the class is public.
- Example: For a class named `Main`, the file should be named `Main.java`.

Other Naming Conventions

- **Class Names**: Use **PascalCase** (e.g., `MyClass`).
- **Method Names**: Use **camelCase** (e.g., `calculateTotal`).
- **Constants**: Use **UPPERCASE** with underscores (e.g., `MAX_VALUE`).
- **Packages**: Use lowercase (e.g., `com.example.app`).

Summary

This README provided an overview of several core Java concepts:

- **The `main()` Method**: The starting point for any Java program, defined as `public static void main(String[] args)`.
- **Import Statements**: Used to include external classes and packages. Types include single class imports, wildcard imports, and static imports.
- **File Naming Conventions**: The Java file name must match the class name for public classes, and Java follows certain conventions for class, method, and package names.

By adhering to these best practices and conventions, your Java code will be cleaner, easier to read, and more maintainable.
