**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**======\*\*\*======**

****

BÁO CÁO THUỘC HỌC PHẦN:

LẬP TRÌNH JAVA

**CHƯƠNG TRÌNH ỨNG D** **ỤNG CHUYỂN ĐỔI MỘT MẢNG SANG SET VÀO THỰC TIỄN.**

|  |  |
| --- | --- |
| GVHD: | Ths Hoàng Quang Huy |
| Sinh viên: | Quách Văn Long - 2019605768 |

Hà nội, Năm 2022

MỤC LỤC

[MỤC LỤC 2](#_Toc113205083)

[1. Phân tích bài toán 3](#_Toc113205084)

[1.1. Đặt vấn đề 3](#_Toc113205085)

[1.1.1. Giới thiệu 3](#_Toc113205086)

[1.1.2. Cấu trúc gói và thư mục 4](#_Toc113205087)

[1.2. Nội dung tóm tắt các lớp 5](#_Toc113205088)

[2. Chạy chương trình Demo 9](#_Toc113205089)

[2.1. Các trường hợp thử nghiệm 9](#_Toc113205090)

[2.1.1. Thử nghiệm với danh sách 1.000 sản phẩm 9](#_Toc113205091)

[2.1.2. Thử nghiệm với danh sách 5.000 sản phẩm 9](#_Toc113205092)

[2.1.3. Thử nghiệm với danh sách 10.000 sản phẩm 10](#_Toc113205093)

[2.2. Kết quả và đánh giá 10](#_Toc113205094)

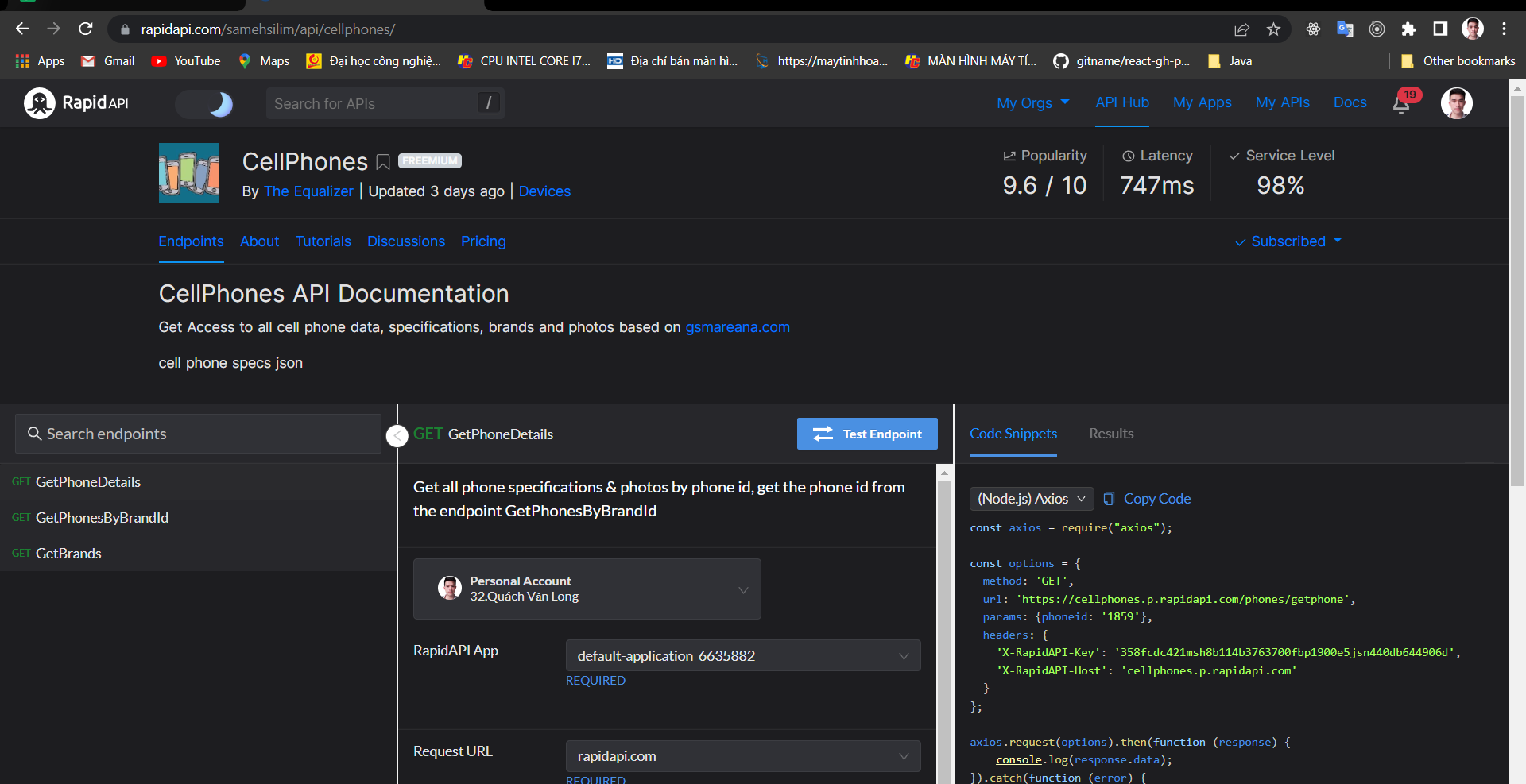
[TÀI LIỆU THAM KHẢO 11](#_Toc113205095)

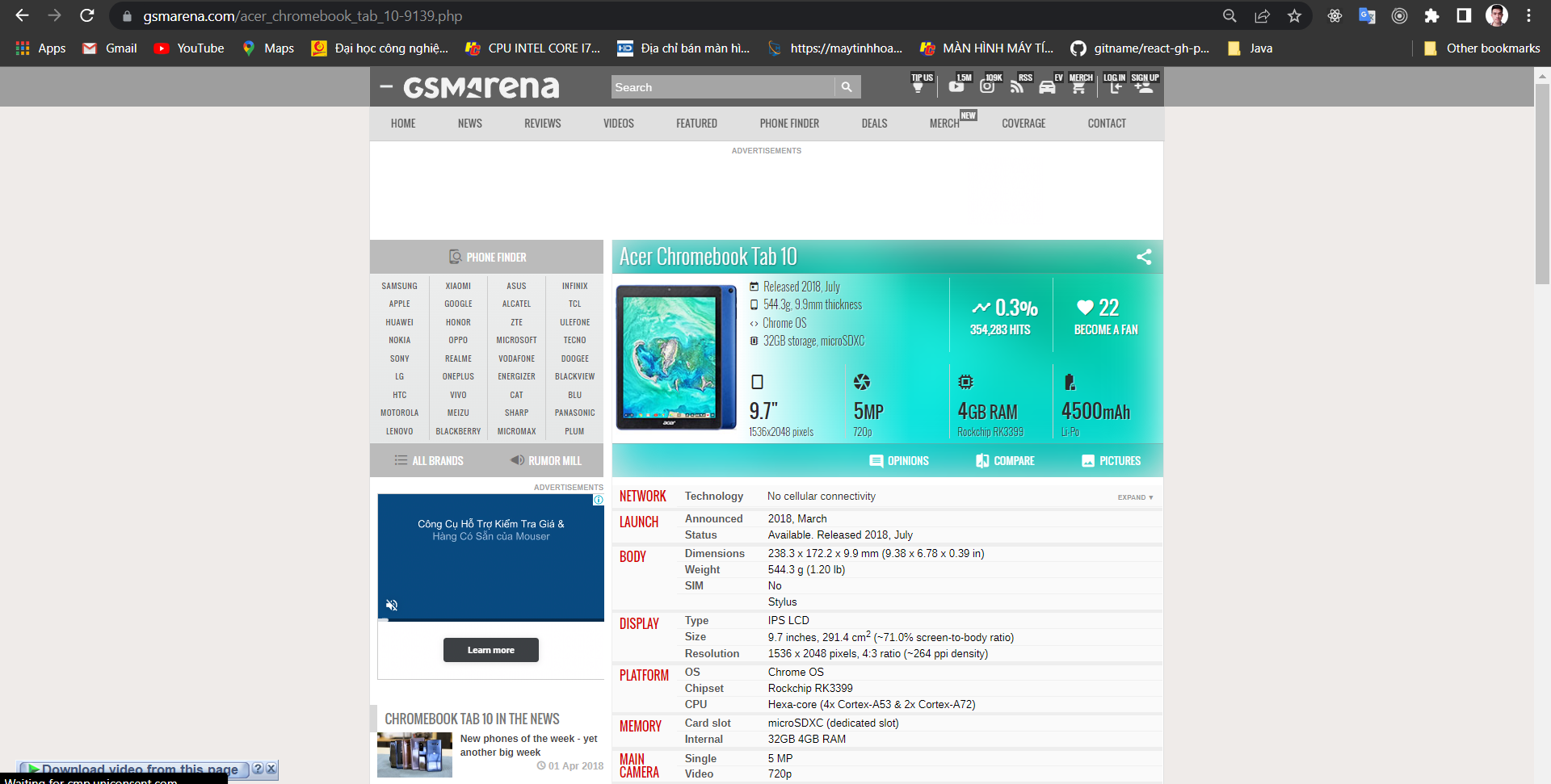
# Phân tích bài toán

## Đặt vấn đề

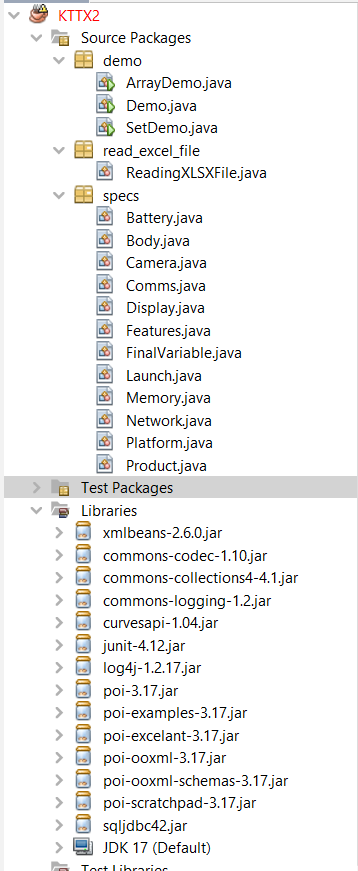
### Giới thiệu

* Trong bài kiểm tra thường xuyên một đã nộp, em đã phân tích kĩ những điểm tốt và cả những điểm còn chưa tốt của hai đối tượng mảng(Array) và Set trong ngôn ngữ Java.Trong báo cáo lần này, em xin phép ứng những gì đã tìm hiểu được trong bài trước vào ứng dụng thương mại điện từ, cụ thể ở đây là chương trình demo xử lý dữ liệu quản lý thông tin điện thoại di động.
* Nguồn dữ liệu em đã lấy từ kênh CellphoneS trên website Rapidapi.com với hơn 11000 bản ghi khác nhau để sử dụng,đồng thời API còn được sự dụng cho trang bán hàng điện tử gsmarena.com:





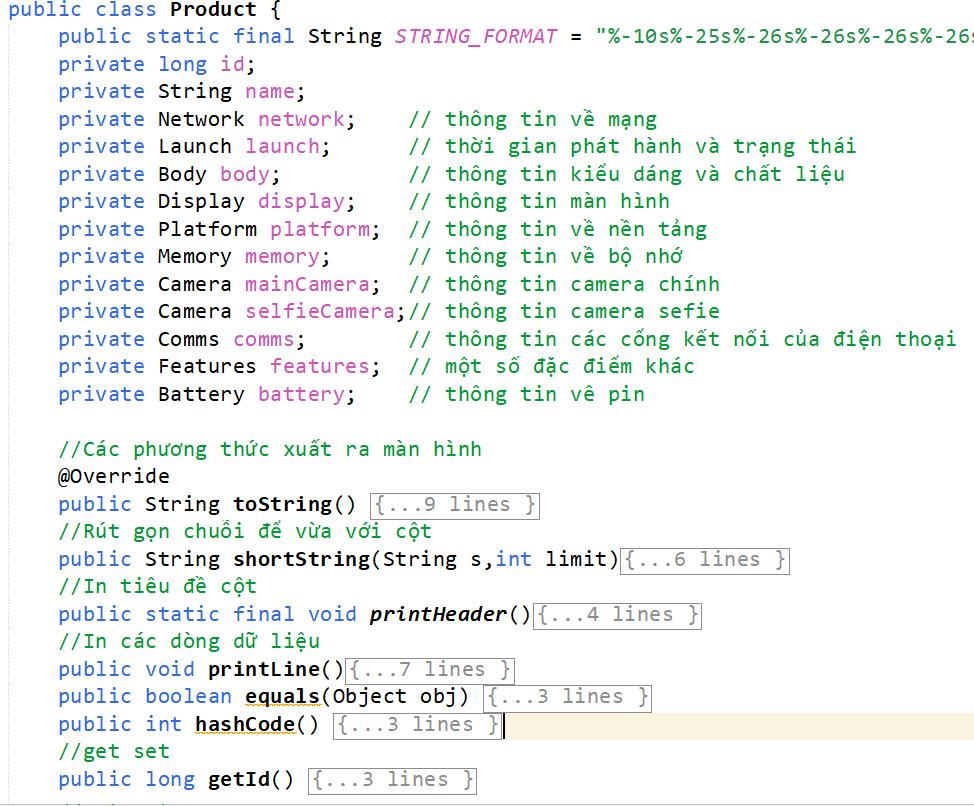
### Cấu trúc gói và thư mục



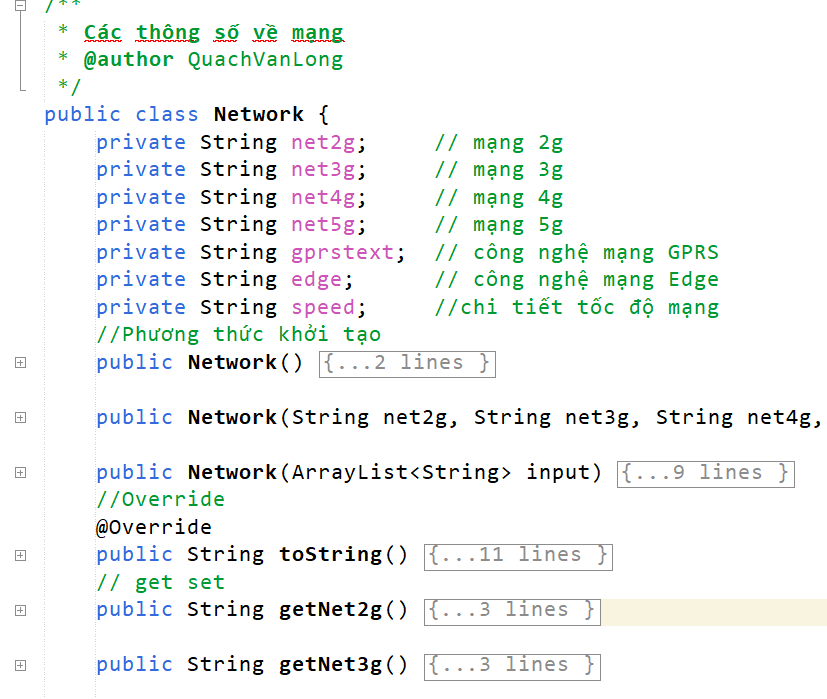
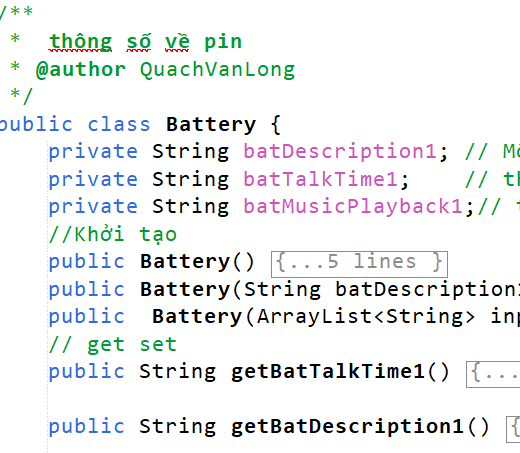
* Vì cần phải đọc file excel nên trong thư mục libraries ta cần phải import một vài thư viện để hỗ trợ đọc file.
* Bên trong thư mục Sources Packages phân ra làm 3 packages nhỏ:
* Demo : là nơi cạy chương trình chính và hai class sử dụng Array và Set riêng biệt.
* Read\_excel\_file : là nơi chưa class đọc file excel.
* Specs(specifications) : là nơi chứa class Product chứa các nhóm thuộc tính, mỗi nhóm thuộc tính bao gồm các thuộc tính có cùng ý nghĩa với nhau.

## Nội dung tóm tắt các lớp

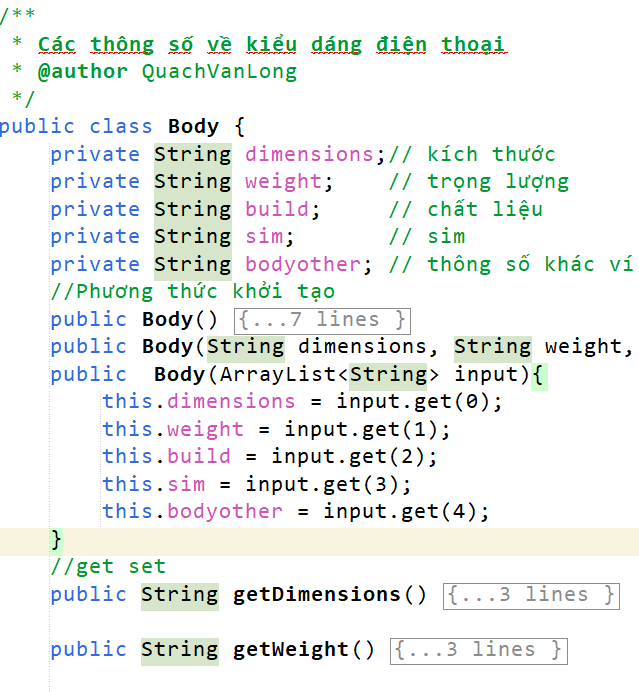
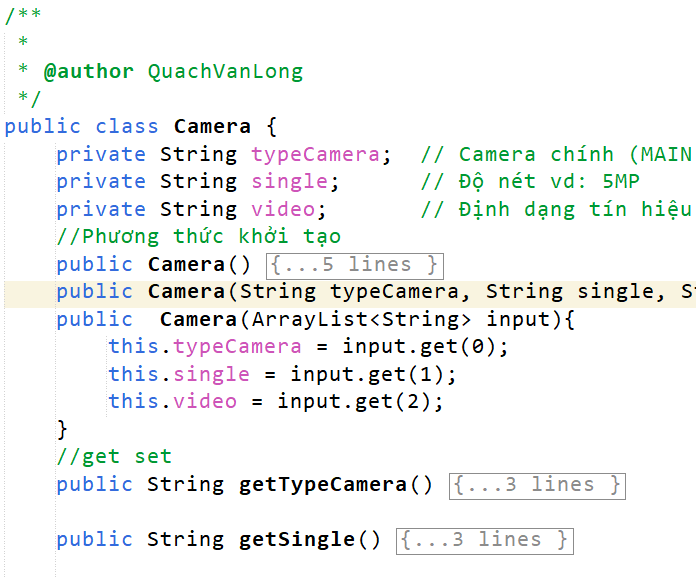
* Lớp Product:



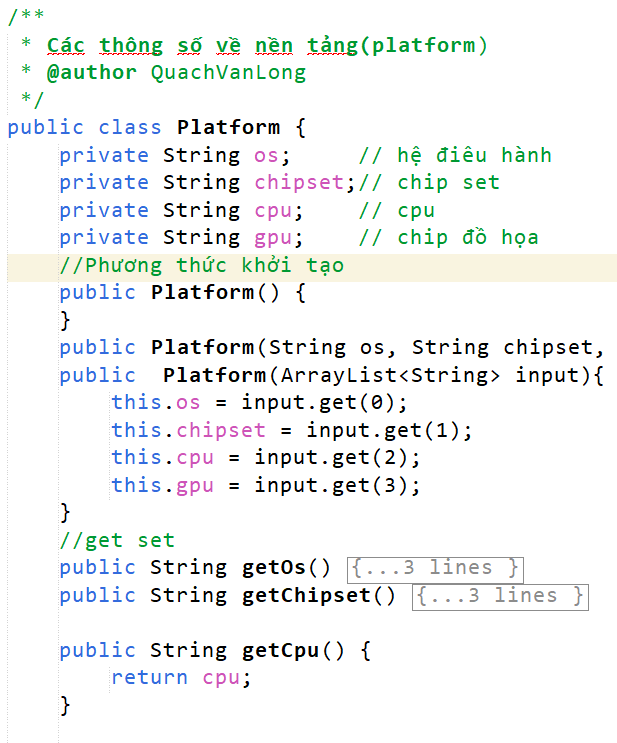
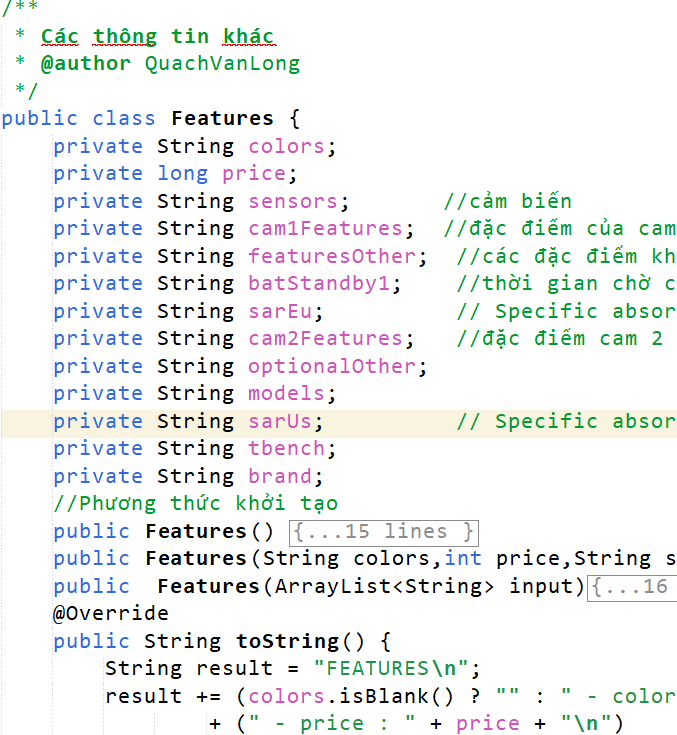
* Lớp Network: - Lớp Baterry:

* Lớp Body: - Lớp Camera

* Lớp Platform: - Lớp Features:

* Một số lớp còn lại như: Launch,Memory,Network cũng có cùng cấu trúc tương tự.
* Class ArrayDemo.java thuộc gói demo:
* Phương thức sắp xếp tăng hoặc giảm với hai giá trị là tên(Name) hoặc giá tiền(Price):

**Giải thích:** Sử dụng phương thức Array.sort() để sắp xếp và thực hiện implements Comparators bởi class sortedByName.

public static void sortedListProductByNameWithArray(Product[] arr, boolean isINC) {

if (isINC) {

is\_Inc = 1;

} else {

is\_Inc = -1;

}

Arrays.sort(arr, new sortedByName());

printAray(arr);

}

* Class SetDemo.java thuộc gói demo:
* **Giải thích**: Sử dụng đặc điểm của TreeSet để sắp xếp và thực hiện implements Comparators bởi class sortedByName.

public static void sortedListProductByNameWithSet(Set<Product> set, boolean isINC) {

if (isINC) {

is\_Inc = 1;

} else {

is\_Inc = -1;

}

Set<Product> treeSet = new TreeSet<>(new sortedByName());

treeSet.addAll(set);

printSet(treeSet);

}

* **Giải thích:** Sử dụng file ItemDemo làm form nhập id của sản phẩm cần sửa và những thông tin liên quan, đồng thời sử dụng cho phương thức tìm kiếm theo id, và xóa sản phẩm theo id.

public static void editProductIntoSet(HashSet<Product> set) throws IOException{

String pathFile = "C:\\Users\\QuachVanLong\\Desktop\\Java\\ItemDemo.xlsx";

HashSet<Product> setReadFile = ReadingXLSXFile.getDataFromExcelFile(1, pathFile);

Product[] arr = setReadFile.toArray(new Product[setReadFile.size()]);

Product item = arr[0];

for (Product product : set) {

if (product.getId() == item.getId()){

System.out.println("Thông tin điện thoại cũ:");

Product.printHeader();

product.printLine();

product = item;

System.out.println("Thông tin điện thoại mới:");

Product.printHeader();

product.printLine();

return;

}

}

System.out.println("Không tìm thấy sản phẩm nào có id là: " + item.getId());

}

public static void findItemIntoSet(HashSet<Product> set) throws IOException{

String pathFile = "C:\\Users\\QuachVanLong\\Desktop\\Java\\ItemDemo.xlsx";

HashSet<Product> setReadFile = ReadingXLSXFile.getDataFromExcelFile(1, pathFile);

Product[] arr = setReadFile.toArray(new Product[setReadFile.size()]);

Product item = arr[0];

if (set.contains(item)){

for (Iterator<Product> iter = setReadFile.iterator(); iter.hasNext();) {

Product next = iter.next();

if (next.equals(item)){

System.out.println("Thông tin sản phẩm có id là: " + item.getId());

Product.printHeader();

next.printLine();

}

}

} else {

System.out.println("Không tìm thấy sản phẩm có id là: "+ item.getId());

}

}

public static void removeItemIntoSet(HashSet<Product> set) throws IOException{

String pathFile = "C:\\Users\\QuachVanLong\\Desktop\\Java\\ItemDemo.xlsx";

HashSet<Product> setReadFile = ReadingXLSXFile.getDataFromExcelFile(1, pathFile);

Product[] arr = setReadFile.toArray(new Product[setReadFile.size()]);

Product item = arr[0];

if (set.contains(item)){

set.remove(item);

System.out.println("Đã xóa sản phẩm có id là: "+ item.getId());

} else {

System.out.println("Không tìm thấy sản phẩm có id là: "+ item.getId());

}

}

# Chạy chương trình Demo

## Các trường hợp thử nghiệm

### Thử nghiệm với danh sách 1.000 sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Phương thức sắp xếp theo tên** | **Phương thức sắp xếp theo giá trị** | **Phương thức Tìm kiếm** | **Phương thức Xóa** | **Phương thức sửa** |
| **Array** | 5,770,700 | 2,982,200 | 59,174,600 | 26,237,200 | 36,503,100 |
| **Set** | 4,736,000 | 2,697,100 | 34,283,500 | 26,141,700 | 6,692,000 |

Bảng 1 : Chạy chương trình sử dụng nano time để đánh giá.

### Thử nghiệm với danh sách 5.000 sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Phương thức sắp xếp theo tên** | **Phương thức sắp xếp theo giá trị** | **Phương thức Tìm kiếm** | **Phương thức Xóa** | **Phương thức sửa** |
| **Array** | 11,274,100 | 6,543,300 | 72,448,800 | 26,626,500 | 38,124,700 |
| **Set** | 13,739,500 | 8,110,800 | 29,507,400 | 17,473,700 | 23,271,700 |

Bảng 3 : Chạy chương trình sử dụng nano time để đánh giá

### Thử nghiệm với danh sách 10.000 sản phẩm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Phương thức sắp xếp theo tên** | **Phương thức sắp xếp theo giá trị** | **Phương thức Tìm kiếm** | **Phương thức Xóa** | **Phương thức sửa** |
| **Array** | 17,416,900 | 11,855,300 | 61,476,100 | 24,877,800 | 30,990,300 |
| **Set** | 20,369,500 | 19,102,200 | 32,191,600 | 16,987,500 | 28,214,000 |

Bảng 4 : Chạy chương trình sử dụng nano time để đánh giá

## Kết quả và đánh giá

* Khi thử nghiệm với 1000 sản phẩm, sự khác biệt về tốc độ giữa Array và Set không có quá nhiều khác biệt.
* Khi chạy với trường hợp 5000 sản phẩm và 10000 sản phẩm đã cho thấy Array thích hợp cho việc sắp xếp dữ liệu và truy xuất dữ liệu. Còn với Set thì thích hợp với thêm, sửa, xóa với lượng dữ liệu lớn.

TÀI LIỆU THAM KHẢO

[[1] - Program to convert Array to Set in Java](https://www.geeksforgeeks.org/program-to-convert-array-to-set-in-java/)

[[2] – Guava.dev](https://guava.dev/releases/22.0/api/docs/src-html/com/google/common/collect/Sets.html#line.231)