Assignment 2 (Streams)

Instructions:

1. Try to attempt all the questions given below. Most of them would require only 1 single line code, some might require 2-3 lines as well
2. All the questions need to be answered using Java Streams, not in the general looping way

For the first seven questions below, you will need this Person.java class, so create this one class and then attempt questions

```java
public class Person {

    private int age;
    private String name;
    private String country;

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

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
```

```java
        this.country = country;
    }

    @Override
    public String toString() {
        return name;

    }

    public static void main(String[] args) {

        List<Person> people = Arrays.asList(
                new Person(20, "John", "USA"),
                new Person(35, "Sam", "Italy"),
                new Person(15, "Jamie", "England"),
                new Person(30, "Robert", "Italy"),
                new Person(20, "James", "Ireland"),
                new Person(25, "Peter", "USA"),
                new Person(5, "Jessica", "Norway"),
                new Person(40, "Roger", "Netherlands"),
                new Person(50, "Jim", "USA")
        );
    }

}
```

Q1. Print the avg age of all the people
Q2. Create a list of all the people who are either greater than 20 or contain any vowel in their name (uppercase or lowercase)
Q3. Create a list of people, sorted in ascending order on the basis of age, if age is the same then sort in descending order of name
Q4. Create a map from this people list where the key is country name and value is count which means a map will tell how many people live in a particular country
Q5. Create a map which stores avg age of people per country (key should be country and value should be average age i.e, double)
Q6. Print the oldest person in every country
Q7. Print the country with most people
Q8. Create a list of 20 random integers in the range 0 - 1000 using Java 8 streams