

Programación de Objetos Distribuidos

Trabajo Práctico 01

Ejercicio 1

1. Bajar de campus el archivo **java-8-playground.tar.gz**
2. Descomprimirlo, incorporarlo a la IDE de su preferencia (es un proyecto maven)
3. Lograr que cada uno de los test compile y ejecute correctamente. Para ello interpretar cada test y hacer las correcciones necesarias para lograr que las aserciones den ciertas (se puede requerir cambiar el código o el valor esperado de la aserción).

Resolver los test en el siguiente orden:

- **TryWithResourcesTest**
- **LambdaTest**
- **FunctionalApiTest**
- **MethodReferenceTest**
- **DefaultMethodTest**
- **OptionalTest**
- **StreamGenerateAndFinalizeTest**
- **StreamReduceTest**
- **StreamCollectTest**
- **StreamIntegrationTest**

Ejercicio 2

Dadas las siguientes clases

```
import static java.util.Objects.requireNonNull;
import static java.util.Optional.ofNullable;

import java.time.LocalDate;
import java.util.Optional;

public class Country {

    private final String name;
    private final Continent continent;
    private final Long population;
    private final Optional<LocalDate> independenceDay;
```

```
    public Country(final String name, final Continent continent, final long population,
final LocalDate independenceDay) {
        this.name = requireNonNull(name);
        this.continent = requireNonNull(continent);
        this.population = new Long(population);
        this.independenceDay = ofNullable(independenceDay);
    }

    public Continent getContinent() {
        return continent;
    }

    public Long getPopulation() {
        return population;
    }

    public String getName() {
        return name;
    }

    public Optional<LocalDate> getIndependenceDay() {
        return independenceDay;
    }

    @Override
    public String toString() {
        return name + " (" + population + " hab)";
    }
}

public enum Continent {
    AFRICA, AMERICA, ANTARTICA, ASIA, EUROPE, OCEANIA;
}
```

y teniendo la siguiente lista

```
Collection<Country> countries = Arrays.asList(
    new Country("Argentina", Continent.AMERICA, 43590400,
LocalDate.parse("1816-07-09")),
    new Country("China", Continent.ASIA, 1377939144, null),
    new Country("Brazil", Continent.AMERICA, 206284591,
LocalDate.parse("1822-09-07")),
    new Country("Nigeria", Continent.AFRICA, 189636000,
LocalDate.parse("1960-10-01")),
    new Country("Japón", Continent.ASIA, 126675000, null),
    new Country("México", Continent.AMERICA, 122273000,
LocalDate.parse("1810-09-16")),
    new Country("Turquía", Continent.ASIA, 79265000, LocalDate.parse("1923-10-29")),
    new Country("Australia", Continent.OCEANIA, 24117000, null),
```

```
        new Country("Chile", Continent.AMERICA, 18192000,
LocalDate.parse("1810-09-18")),
        new Country("Bélgica", Continent.EUROPE, 11338000,
LocalDate.parse("1930-10-04")),
        new Country("Portugal", Continent.EUROPE, 10262000,
LocalDate.parse("1640-12-01")),
        new Country("Bolivia", Continent.AMERICA, 10985000,
LocalDate.parse("1825-08-06")),
        new Country("Suecia", Continent.EUROPE, 9824000, LocalDate.parse("1523-06-06")),
        new Country("Israel", Continent.ASIA, 8391000, LocalDate.parse("1948-05-15")),
        new Country("Inglaterra", Continent.EUROPE, 53010000, null),
        new Country("Serbia", Continent.EUROPE, 7071000, LocalDate.parse("1804-02-15"))
    );
```

se pide, utilizando las clases y métodos de Java 8 (*stream*, *filter*, *forEach*, *collect*, *max*):

1. Imprimir en pantalla todos los países
2. Imprimir en pantalla a los países de América.
3. Crear una lista con los países de América.
4. Crear un set con los países que contengan en su nombre la letra 'a'.
5. Crear una lista con los nombres de los países que contengan en su nombre la letra 'a'.
6. Crear un mapa (K,V) con K: Continente y V: Lista de países.
7. Crear un mapa (K,V) con K: Continente y V: Total de habitantes.
8. Crear un mapa (K,V) con K: Continente y V: Promedio de habitantes.
9. Obtener un país que tenga una 'a' en su nombre y la cantidad de habitantes sea mayor a 1.000.000.000. En caso de no encontrar uno, devolver null.
10. Obtener el país con el mayor número de habitantes.
11. Listar todos los países cuya independencia fue en un año bisiesto.
12. Armar un mapa que muestre la cantidad de países que se independizaron por siglo. Donde el siglo se puede escribir como 1800 o 19 (para toda independencia ocurrida en un año 18XX).

Fuentes de los datos:

- https://en.wikipedia.org/wiki/List_of_national_independence_days
- <https://www.saberespractico.com/demografia/paises-del-mundo-ordenados-por-poblacion-actualizado/>