Practical work. File processing. Advanced

Task 1

Read and combine data of all data files into Country objects.
You will need to read:

- ➔ From countries.csv - Name, region, population, coastline, gdp per capita, birth rate, death rate
- ➔ From happiness_rank_2019.csv - Name, overall rank, score
- ➔ From suicides.csv - Name, suicides per 100k (only for males in the age group 15-24 years at 2010).

NB! Country list is to be based on countries.csv file. Those countries that are mentioned in happiness or suicide files but are not in countries.csv, should be skipped. Total count of countries - 226.

Task 2

From the Country objects calculate:

- ➔ How many countries with negative population growth are in the world's Happiness rating Top 20? (answer: 4)
- ➔ How many and which European countries have no coastline? (11)
- ➔ How many and which countries have GDP per capita and happiness index higher than Latvia? (36)
- ➔ How many and which countries have the asked suicide data? (88)
- ➔ How many and which countries have lower happiness index than the country with the highest suicides per 100k value (and which one is it)? (40 are lower than Lithuania - most suicidal).

Guided level

Open all the csv files with a text editor (notepad, VSC, Notepad++ or any other) or customly open with a spreadsheet program. Explore the contents, column names, order and invalid values.

Task 1

Create and call a method that reads the first file - countries.csv. To skip reading the header, call nextLine() method for the Scanner object once before the while loop. When reading lines, split each one by the file’s delimiter into an array of Strings.
Use an “if‟ statement and a “continue‟ statement to skip unneeded lines:

```
if (line.isBlank() and other line checking conditions) {
  continue; // skip this line
}
```

Determine the positions of all the necessary columns and print a couple of them to see if the file is read correctly.

Create a Country class with all the fields described at the top that a Country will have.
You will need a collection of all the Country objects. An ArrayList might work, but now a HashMap (where the country name is the key and Country object is the value) will be better.
Create a constructor that either receives a String array or separate String values that will be taken from the file.
In the constructor save simple values like country name and region and add the object to the hashmap.
Now overwrite the default toString() method of the Country:

```
public String toString() {
  return this.name; //or add something more
}
```

Now go back to the main() method and print the Country HashMap to see if you have populated the map correctly.
Now to continue the task you need to create methods that try to parse String values into int and double to get the file values as proper numbers.

As the files can have invalid data (empty cells) it’s ok to use try/catch in the value parsing. One method could look and be used like this:

```
static int getInt(String val) {
    try {
      return Integer.parseInt(val);
    } catch (Exception e) {
      return 0;
    }
}

//when used in a constructor:
this.population = getInt(populationString);
```

Now parse all the values to numbers and assign them to Country objects in the constructor.

Now the first file content is taken, time to take care of the rest.
Reading of the 2nd and 3rd file will be similar to each other. First, create a method that reads the Happiness rank file and skip the bad lines.
Again, split the line by the used delimiter into an array of String values and find which element contains the country name.
Now we need to determine if each line is needed - if the country is in our map. To do that, you need to use the containsKey() method for the map. If the map doesn’t contain the key, skip that line (using *continue*) as that is a country or a region we don’t need data for.
After skipping, get a corresponding Country object from the map by using the get() method for the map. It could look like this:

```
Country country = Country.map.get(lineParts[1]);
```

And now save the happiness position and happiness index for that country by reading the data from the line. You might use the same getInt() method from the Country class. It might (but doesn’t have to) look like this:

```
country.happyPos = Country.getInt(parts[0]);
```

Now do pretty much the same with the suicides data file. Only the valid/invalid lines will have a much more complex condition (read the task again). Only value you have to add it the suicide rate per 100k.

After getting and combining all the data from all the files into Country objects, now you can do regular country calculations. One of the ways how to iterate the Country map is this:

```
for (String name : Country.map.keySet()) {
        Country country = Country.map.get(name);
        //do something with the country object
}
```

All the rest is up to you.

Processes should be separated into designated methods.
Idea of how the main method could look like this:

```
public static void main(String[] args) {
        getBase("countries");
        getHappiness("Happiness_rank_2019");
        getSuicides("suicide_rates_per_country");
        Country.getHappyIncrease();
        Country.countEuropeCoastline();
        Country.countHappierRicherThan("Latvia");
        Country.countUnhappySuicidals();
}
```