

Итоговый проект

Цели для выполнения проекта:

1. Выбрать свой вариант задания согласно номеру в таблице.
2. Разработать набор классов с необходимым составом полей и методов для хранения и обработки всех данных, хранящихся в CSV-файле вашего варианта задания.
3. Распарсив данные в файле CSV, нужно по ним создать набор объектов, заполнив все необходимые поля.
4. Создать файл БД SQL Lite и подключить ее к проекту.
5. В БД создать набор таблиц (по 3-ей нормальной форме) согласно набору объектов.
6. Все данные из набора объектов сохранить в БД.
7. Сделать набор SQL-запросов к БД согласно своему варианту задания.
8. Полученные данные вывести в текстовом виде в консоль, сделать скриншоты всех полученных данных.
9. Числовые данные по заданию визуализировать в виде диаграмм, сделать скриншоты всех полученных данных.
10. Для проекта создать открытый репозиторий на github и закоммитить туда всю работу по проекту.
11. В файле README описать последовательность работы по вашему проекту, а также прикрепить все скриншоты.
12. Ссылку на github проект прикрепить в этом задании.

Ход выполнения работы:

1. Мое место в ведомости 120 , значит мой вариант 10(Counter).

Ведомость Java (Ulearn) 2022-2023 - Осень ☆ 🔍 ↶

файл Правка Вид Вставка Формат Данные Инструменты

🔍 100% 🔒 Только просмотр

fx Мохаммед Амджед Ельсади́г Ахмед

A	B	
	ФИО	Фамил
103	Мичоогүллары Эфе Гази	Мичоогүлла
104	Моденов Михаил Васильевич	Моденов Ми
105	Молчанов Тимофей Семенович	Молчанов Ти
106	Мохамед Магди Магди Хашем	Мохамед Ма
107	Мохамед Мансур Мохамед Махмуд	Мохамед Ма
108	Мусин Эдуард Борисович	Мусин Эдуа
109	Мусина Диана Ильгамовна	Мусина Диа
110	Мустафа Ахмед Саид Ахмед Салех	Мустафа Ах
111	Наджафулиев Эмир Маликович	Наджафули
112	Нгуен Суан Минь	Нгуен Суан
113	Ненилин Владислав Олегович	Ненилин Вла
114	Николаев Егор Николаевич	Николаев Ег
115	Новгородцев Никита Сергеевич	Новгородцев
116	Новиков Артём Вячеславович	Новиков Арт
117	Новокрещенова Алёна Вячеславовна	Новокрещен
118	Нурмухаметова Екатерина Раулевна	Нурмухаметс
119	Олексюк Екатерина Андреевна	Олексюк Ека
120	Пашенко Никита Алексеевич	Пашенко Ни

2. В моем варианте Counter 5 значений строк, значит и в основную базу пойдут все 5 (фото оригинальной таблицы ниже). Первые 3 строчки имеют текстовое значение (TEXT), последние 2 имеют целочисленный тип данных (INTEGER). Код для создания таблицы будет выглядеть так - CREATE TABLE CountryTable ("Country or area" TEXT(100), "Subregion" TEXT(100), "Region" TEXT(100), "Internet users" INTEGER, "Population" INTEGER);

Country or area	Subregion	Region	Internet users	Population
China	Eastern Asia	Asia	1,010,740,000	1,427,647,786
India	Southern Asia	Asia	833,710,000	1,352,642,280
United States	Northern America	Americas	312,320,000	327,096,265

Для удобства и более красивой визуализации таблиц я использовал приложение SQLite Studio (так как ограничений в использовании каких либо программ в условии прописано не было)

Проверил наличие в базе таблицы при помощи команды .table (название базы CountryJava.db а сама таблица Country Table)

```

D:\SQL Lite\sqlite3.exe
SQLite version 3.40.1 2022-12-28 14:03:47
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open CountryJava.db
sqlite> .table
CountryTable
sqlite> .mode csv
sqlite> .import D:/SQL Lite/Country.csv cities
  
```

Получившаяся таблица со своими типами данных, в более приятном виде.

Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальность	Проверка	Не NULL	Сравнение	Сгенерированное
1 Country or area	TEXT (100)							NL
2 Subregion	TEXT (100)							NL
3 Region	TEXT (100)							NL
4 Internet users	INTEGER							NL
5 Population	INTEGER							NL

3. Импортировал значения файла CSV в готовый шаблон таблицы при помощи приложения (GIF с заполненной таблицей)

Структура

Данные

Ограничения








Индексы

Триггеры






DDL

Табличный вид


Форма



1



Отфильтровать да...

Всего загружено строк: 214

	Country or area	Subregion	Region	Internet users	Population
1	China	Eastern Asia	Asia	1,010,740,000	1,427,647,786
2	India	Southern Asia	Asia	833,710,000	1,352,642,280
3	United States	Northern America	Americas	312,320,000	327,096,265
4	Indonesia	South-eastern Asia	Asia	196,000,000	267,670,543
5	Brazil	South America	Americas	160,010,801	209,469,323
6	Nigeria	Western Africa	Africa	136,203,231	195,874,685
7	Bangladesh	Southern Asia	Asia	129,180,000	166,303,498
8	Russia	Eastern Europe	Europe	124,000,000	145,734,038
9	Pakistan	Southern Asia	Asia	118,800,000	213,756,286
10	Japan	Eastern Asia	Asia	117,400,000	127,202,192
11	Mexico	Northern America	Americas	92,010,000	128,972,439
12	Iran	Western Asia	Asia	78,086,663	81,800,188
13	Germany	Western Europe	Europe	77,794,405	83,124,418
14	Philippines	South-eastern Asia	Asia	73,003,313	106,651,394
15	Turkey	Southern Europe	Europe	69,945,905	82,340,088
16	Vietnam	South-eastern Asia	Asia	68,172,134	95,545,962
17	United Kingdom	Northern Europe	Europe	65,001,016	67,141,684
18	France	Western Europe	Europe	59,470,000	64,990,511
19	Egypt	Northern Africa	Africa	54,740,141	98,423,598
20	South Korea	Eastern Asia	Asia	49,421,084	51,171,706
21	Spain	Southern Europe	Europe	42,400,756	46,692,858
22	Thailand	South-eastern Asia	Asia	54,043,108	69,428,453
23	Italy	Southern Europe	Europe	50,540,000	60,627,291
24	Poland	Eastern Europe	Europe	34,697,848	37,921,592
25	Canada	Northern America	Americas	33,950,632	37,064,562
	Argentina	South America	Americas	22,561,076	44,261,150

В файле CSV 215 строк, из них 1 с названиями колонок и того 214 с данными о странах(как видно на гифке строчек 214, что означает – импорт прошел без потерь данных)

4. Подключил базу данных к проекту (в первый раз получил ошибку тк как не использовал драйвера для подключения “sqlite-jdbc-3.40.0.0.jar”)

```

D:\SQL Lite>javac ProjectJava.java

D:\SQL Lite>java ProjectJava.java
org.sqlite.JDBC

D:\SQL Lite>d
main 2023-01-04T11:37:10.962Z]
"d" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.id 24516 exited with code:

D:\SQL Lite>java -cp sqlite-jdbc-3.40.0.0.jar;. ProjectJava
Connected

D:\SQL Lite>

```

Был написан простой код на подключение к базе данных и при подключении он выводит в консоль Connected, как продемонстрировано выше

```

1  import java.sql.*;
2
3  public class ProjectJava
4  {
5      public static void main (String [] args)
6      {
7          try
8          {
9              Class.forName("org.sqlite.JDBC");
10             Connection connection = DriverManager.getConnection(
11                 "jdbc:sqlite:D:\\SQL Lite\\CountryJava.db");
12             System.out.println("Connected");
13         }
14         catch (Exception e)
15         {
16             System.out.println (e.getMessage());
17         }
18     }
19 }

```

5. По заданию нужно было привести таблицу в формат 3NF базы данных, и так получились связи (так же были заполнены командой INSERT):

Страна –Местонахождение

CountryJava					
Имя таблицы:		location country			
	Имя	Тип данных	Первичный ключ	Внешний ключ	Уникальн
1	Country or area	TEXT (100)			
2	Subregion	TEXT (100)			
3	Region	TEXT (100)			

Табличный вид Форма

Отфильтр

	Country or area	Subregion	Region
1	China	Eastern Asia	Asia
2	India	Southern Asia	Asia
3	United States	Northern America	Americas
4	Indonesia	South-eastern Asia	Asia
5	Brazil	South America	Americas
6	Nigeria	Western Africa	Africa
7	Bangladesh	Southern Asia	Asia
8	Russia	Eastern Europe	Europe
9	Pakistan	Southern Asia	Asia
10	Japan	Eastern Asia	Asia
11	Mexico	Northern America	Americas
12	Iran	Western Asia	Asia
13	Germany	Western Europe	Europe
14	Philippines	South-eastern Asia	Asia
15	Turkey	Southern Europe	Europe
16	Vietnam	South-eastern Asia	Asia
17	United Kingdom	Northern Europe	Europe
18	France	Western Europe	Europe
19	Egypt	Northern Africa	Africa
20	South Korea	Eastern Asia	Asia
21	Spain	Southern Europe	Europe
22	Thailand	South-eastern Asia	Asia
23	Italy	Southern Europe	Europe
24	Poland	Eastern Europe	Europe
25	Canada	Northern America	Americas
26	Argentina	South America	Americas

Страна – Население

Структура Данные Ограничения Индексы

CountryJava

Имя таблицы: people country

	Имя	Тип данных	Первичный ключ	В
1	Country or area	TEXT (100)		
2	Internet users	INTEGER		
3	Population	INTEGER		

Табличный вид

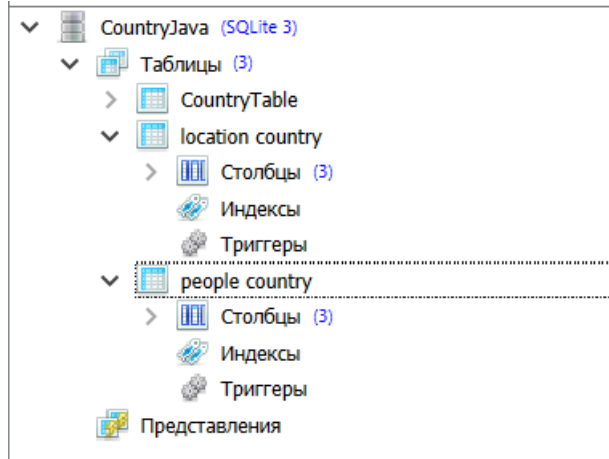
Форма



Отфильтровать

	Country or area	Internet users	Population
1	China	1,010,740,000	1,427,647,786
2	India	833,710,000	1,352,642,280
3	United States	312,320,000	327,096,265
4	Indonesia	196,000,000	267,670,543
5	Brazil	160,010,801	209,469,323
6	Nigeria	136,203,231	195,874,685
7	Bangladesh	129,180,000	166,303,498
8	Russia	124,000,000	145,734,038
9	Pakistan	118,800,000	213,756,286
10	Japan	117,400,000	127,202,192
11	Mexico	92,010,000	128,972,439
12	Iran	78,086,663	81,800,188
13	Germany	77,794,405	83,124,418
14	Philippines	73,003,313	106,651,394
15	Turkey	69,945,905	82,340,088
16	Vietnam	68,172,134	95,545,962
17	United Kingdom	65,001,016	67,141,684
18	France	59,470,000	64,990,511
19	Egypt	54,740,141	98,423,598
20	South Korea	49,421,084	51,171,706
21	Spain	42,400,756	46,692,858
22	Thailand	54,043,108	69,428,453
23	Italy	50,540,000	60,627,291
24	Poland	34,697,848	37,921,592
25	Canada	33,950,632	37,064,562
26	Argentina	33,561,876	44,361,150

6. Все данные были сохранены после создания и заполнения таблиц в бд.



7. Для того что бы сделать набор SQL-запросов к БД согласно своему варианту задания, сначала подготовил приложение, а точнее для первого задания, где нужно было построить график процентного соотношения. Так как нужно визуализировать данные я использовал JavaFX и по новой создал проект перенеся подключение к БД(Так выглядит конечный код, где были добавлены методы в Main для дальнейших запросов).

```
public class Main extends Application
{
    Connection connection;
    static Map<String,Double> percentUsersSubregion1 = new HashMap <String,Double>();
    static Map<String,List<String>> SubregionCountry = new HashMap<String,List<String>>();
    public static void main(String[] args)
    {
        Main program = new Main();
        if (program.connect())
        {
            Map<String,Integer> subregionMap = getAllSubregion();//подготовка данных для 1го задания
            percentUsersSubregion1 = GetPercentUsersSubregion(subregionMap);//подготовка данных для 1го задания
            SmallestNumberOfRegisteredInWesternEurope();// 2е задание
            percentageOfInternetUsersInCountries();// 3е задание
            Launch(args);// вызов таблицы
        }
    }

    boolean connect()
    {
        try
        {
            Class.forName("org.sqlite.JDBC");
            Connection connection = DriverManager.getConnection(
                "jdbc:sqlite:D:\\SQL Lite\\CountryJava.db");
            System.out.println("Connected");
            return true;
        }
        catch (Exception e)
        {
            System.out.println (e.getMessage());
            return false;
        }
    }
}
```

Задание в себе имеет 3 пункта, перечислены фотографией ниже.

10. Country	Постройте график процентного соотношения пользователей в интернете от всего населения по субрегионам.	Выведите название страны с наименьшим кол-вом зарегистрированных в ин-ете пользователей в Восточной Европе.	Выведите в консоль название страны процент зарегистрированных в интернете пользователей которой находится в промежутке от 75% до 85%
-------------	---	---	--

Для первого задания нужно было подготовить словарь с названием субрегиона и его процента соотношения населения. Что бы подготовить данный словарь нужно было взять данные из двух таблиц [location country] из которой подготовить связь страна – субрегион и таблицы [people country] из которой нужно было взять данные по странам и при помощи данных населения найти процент населения который

Сразу хотелось бы добавить, если бы не было условия создать таблицу по 3 форме, решалось бы все в два запроса, а так как таблицы 2 и запроса получилось 3

Первый шаг, который я сделал это собрал все страны, подсчитал их кол-во и записал в свой субрегион свое кол-во.

Второй шаг для решения задачи, я собрал лист стран в свои субрегионы (то есть записал в словарь где ключ субрегион и значение лист стран) .

```
public static Map<String,Integer> getAllSubregion() /// 1 часть решения 1 задачи.
{
    try
    {
        Connection connection = DriverManager.getConnection("jdbc:sqlite:D:\\SQL Lite\\CountryJava.db");// подключение к бд
        Statement statement = connection.createStatement();

        ResultSet resultSet = statement.executeQuery("SELECT Subregion FROM [location country]");// создание запроса на получение всех субрегионов

        Map<String,Integer> dictionarySubregions = new HashMap<String,Integer>(); // будет ниже реализован подсчет стран входящий в субрегионы
        ArrayList<String> places = new ArrayList<String>();// в листе будет запись всех данных из запроса

        while (resultSet.next())
        {
            String subregion = resultSet.getString("Subregion");// преобразование данных столбика Subregion в строку
            places.add(subregion);// добавление в лист
        }

        Map<String, Integer> frequency = places.stream().collect(Collectors.toMap( e -> e, e -> 1, Integer::sum));// подсчет всех стран
        frequency.forEach((k, v) -> dictionarySubregions.put(k, v)); ///System.out.println(dictionarySubregions);// запись субрег. и кол-ва в нем стран
        return dictionarySubregions;
    }

    catch (Exception e) // если будет ошибка в запросе или подключении то выведет что не так.
    {
        System.out.println(e.getMessage());
        return null;
    }
}
```

```

// после заполнения словаря субрегион - ли стран начинаю подсчитывать кол-во населения и пользователей сети
for (int i = 0; i < subregionMap.keySet().size(); i++)// пробегаюсь по всем субрегионам
{
    ResultSet resultSet = statement.executeQuery("SELECT \"Internet users\\\", Population FROM [people country] \"
        + \"WHERE Population IS NOT NULL and Population <= \" AND \"Country or area\" IN \"
        + SubregionCountry.get(SubregionCountry.keySet().toArray()[i])
        .toString().replace(\"[\", \"(\").replace(\"]\", \")\").replace(\"\", \"\", \"\", \"\") );
    // запрос имеет вид - выбрать пользователей интернета + популяцию страны ГДЕ популяция не пустая(в таблице было 3-4 пустых значения)
    // и где страны состоят в списке (далее список стран по субрегиону).
    //Для придания формы нужного для запроса я использовал ком репейс удаляя и заменяя кавычки и запятые на круглые скобки и разбиение значения по ""
    //То есть для запроса нужна была форма ("страна", "страна", "страна", ...)

    var user = 0; // счетчик пользователей интернета
    var popul = 0; // счетчик населения
    while (resultSet.next())
    {
        var Internet_users = Integer.parseInt(resultSet.getString("Internet users").replace(",", "")); // запись населения и пользователей интернета по столбцам
        var Population = Integer.parseInt(resultSet.getString("Population").replace(",", "")); // и так же идет удаление запятых, то есть из формы 111,111,111 в 111111111
        user += Internet_users; //подсчет всего кол-ва
        popul += Population; //подсчет всего кол-ва

    }
    percentUsersSubregion.put(SubregionCountry.keySet().toArray()[i].toString(),
        Double.parseDouble(String.format("%.3g%n", //вместо Эки можно поставить сколько нужно знаков
            (double) user * 100.0 / (double) popul) // формула для подсчета процента населения и изверов
            .replace(",", "."))); // замена , на . для того что бы перевести стринг в дэбл.
}
return percentUsersSubregion;
}

catch (Exception e) // если какая то ошибка опять же то будет отсылать что не так или запрос или данные не в том формате.
{
    System.out.println(e.getMessage());
    return null;
}

```


Финальная часть первого задания, используя получившийся словарь, где ключ – субрегион, а значение процент пользователей в интернете. Заношу его в график.

```
@Override public void start(Stage stage) // не углублялся как он работает, взято из JavaJx
{
    Scene scene = new Scene(new Group());
    stage.setTitle("График процентного соотношения."); // название
    stage.setWidth(900); // размер в длину
    stage.setHeight(800); // в ширину

    ObservableList<PieChart.Data> pieChartData = FXCollections.observableArrayList(); // коллекция данных которые в будущем войдут в график

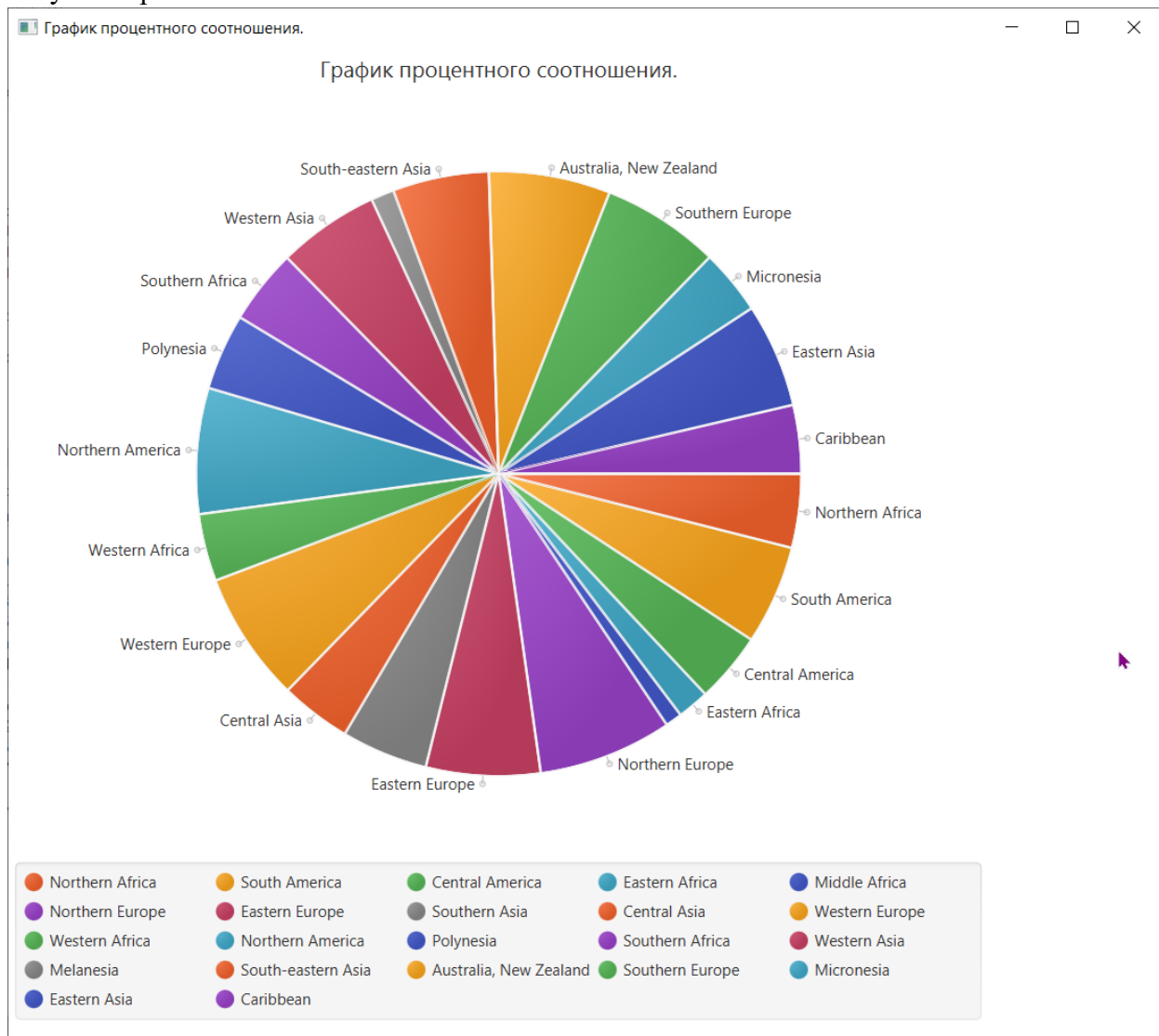
    for (int i = 0; i < percentUsersSubregion1.keySet().size(); i++) // пробегаюсь по всем субрегионам
    {
        pieChartData.add(new PieChart.Data(percentUsersSubregion1.keySet().toArray()[i].toString(), // записываю название субрегиона
            percentUsersSubregion1.get(percentUsersSubregion1.keySet().toArray()[i].toString()))); // значение субрегиона
    }

    final PieChart chart = new PieChart(pieChartData); // тут объявление графика
    chart.setMinWidth(750); // размеры графика в ширину
    chart.setMinHeight(750); // в высоту
    chart.setLabelLineLength(6); // на сколько далеко будут названия от графика
    chart.setTitle("График процентного соотношения.");

    for (final PieChart.Data data : chart.getData()) // ивент которые будет выводить данные субрегиона по которому нажмете
    {
        data.getNode().addEventHandler(MouseEvent.MOUSE_PRESSED, // добавление ивента
            new EventHandler<MouseEvent>() {
                @Override public void handle(MouseEvent e) {
                    System.out.println("-----");
                    System.out.println(String.valueOf("In the subregion " + data.getName()
                        + " " + data.getPieValue() + "% the population uses the Internet")); // будет выводить - В субрегионе "название" "процент" % от пользователей интернета
                }
            }
        );
    }

    ((Group) scene.getRoot()).getChildren().add(chart); // добавление значений
    stage.setScene(scene); // окно?
    stage.show(); // показ
}
}
```

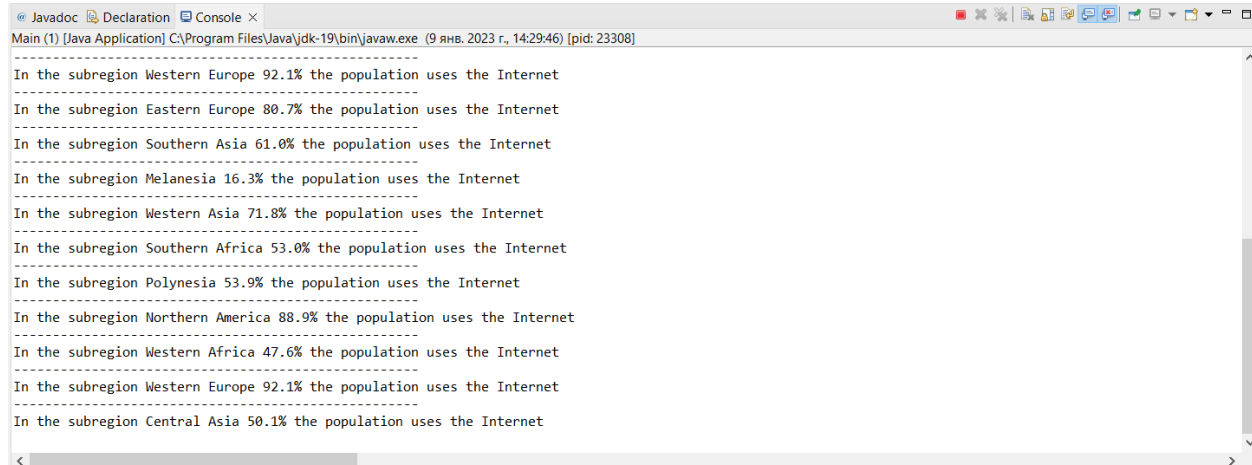
Результат работы -



Так же если нажать на часть графика, например на северную африку, выдаст в консоли

In the subregion Northern Africa 52.2% the population uses the Internet

Ну или если сделать много таких запросов -



```
-----  
In the subregion Western Europe 92.1% the population uses the Internet  
-----  
In the subregion Eastern Europe 80.7% the population uses the Internet  
-----  
In the subregion Southern Asia 61.0% the population uses the Internet  
-----  
In the subregion Melanesia 16.3% the population uses the Internet  
-----  
In the subregion Western Asia 71.8% the population uses the Internet  
-----  
In the subregion Southern Africa 53.0% the population uses the Internet  
-----  
In the subregion Polynesia 53.9% the population uses the Internet  
-----  
In the subregion Northern America 88.9% the population uses the Internet  
-----  
In the subregion Western Africa 47.6% the population uses the Internet  
-----  
In the subregion Western Europe 92.1% the population uses the Internet  
-----  
In the subregion Central Asia 50.1% the population uses the Internet  
-----
```

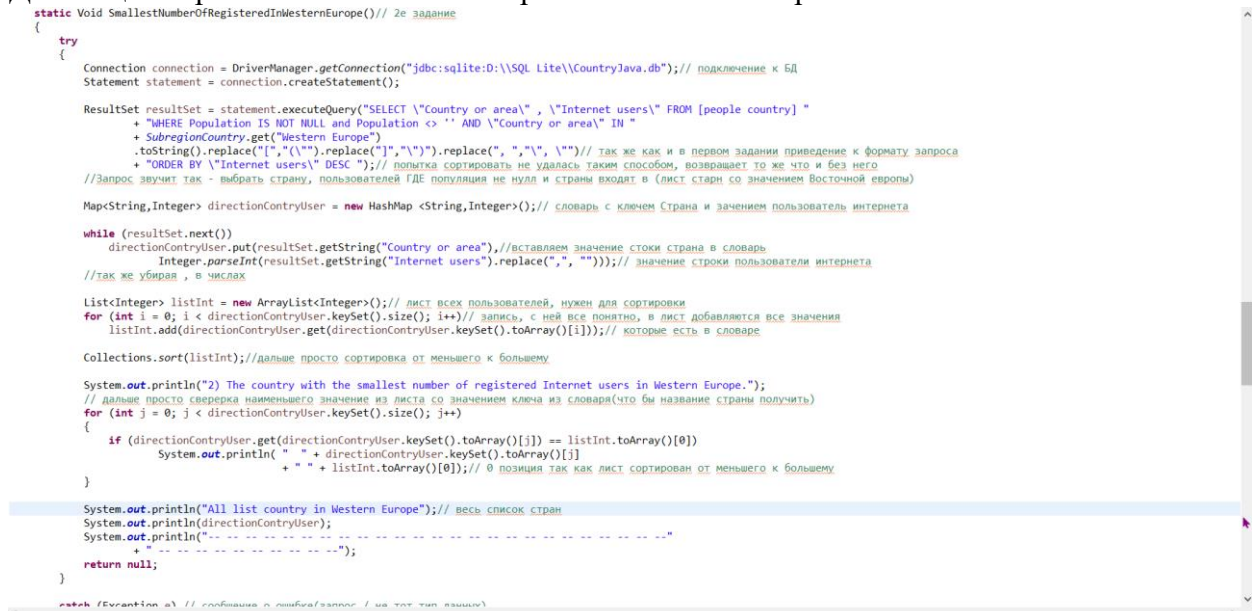
Для второго задания, которое звучит так –

“Выведите название страны с наименьшим кол-вом зарегистрированных в ин-ете пользователей в Восточной Европе.”.

Работа проходила с таблицей [people country].

Для него я взял готовый словарь Субрегион – лист стран и по нему нашел значения для каждой страны и записал в новый словарь Страна - пользователи. После чего все значения пользователей записал в лист для сортировки от меньшего к большему. Ну и просто сравнил первое значение в листе со значением ключа, вывел на экран.

Для общей картины вывел весь словарь со значениями стран – пользователь



```
static void SmallestNumberOfRegisteredInWesternEurope() // 2е задание  
{  
    try  
    {  
        Connection connection = DriverManager.getConnection("jdbc:sqlite:D:\\SQL Lite\\CountryJava.db");// подключение к БД  
        Statement statement = connection.createStatement();  
  
        ResultSet resultSet = statement.executeQuery("SELECT \"Country or area\", \"Internet users\" FROM [people country] "  
            + "WHERE Population IS NOT NULL and Population <> '' AND \"Country or area\" IN "  
            + "SubregionCountry.get(\"Western Europe\")  
            .toString().replace("[", "(").replace("]", ")").replace(",", ", ").replace("\\", "\\\\");// так же как и в первом задании приведение к формату запроса  
            + "ORDER BY \"Internet users\" DESC "); // попытка сортировать не удалась таким способом, возвращает то же что и без него  
        //Запрос звучит так - выбрать страну, пользователей где популяция не нуль и страны входят в (лист стран со значением Восточной европы)  
  
        Map<String,Integer> directionContryUser = new HashMap<String,Integer>();// словарь с ключем Страна и значением пользователь интернета  
  
        while (resultSet.next())  
        {  
            directionContryUser.put(resultSet.getString("Country or area"),resultSet.getString("Internet users").replace(",","")); //вставляем значение строки страна в словарь  
            //так же убирая , в число  
        }  
  
        List<Integer> listInt = new ArrayList<Integer>();// лист всех пользователей, нужен для сортировки  
        for (int i = 0; i < directionContryUser.keySet().size(); i++)// запись, с ней все понятно, в лист добавляются все значения  
        {  
            listInt.add(directionContryUser.get(directionContryUser.keySet().toArray()[i])); // которые есть в словаре  
        }  
  
        Collections.sort(listInt);//дальше просто сортировка от меньшего к большему  
  
        System.out.println("2) The country with the smallest number of registered Internet users in Western Europe.");  
        // дальше просто сверка наименьшего значение из листа со значением ключа из словаря(что бы название страны получить)  
        for (int j = 0; j < directionContryUser.keySet().size(); j++)  
        {  
            if (directionContryUser.get(directionContryUser.keySet().toArray()[j]) == listInt.toArray()[0])  
            {  
                System.out.println(" " + directionContryUser.keySet().toArray()[j] + " " + listInt.toArray()[0]); // 0 позиция так как лист сортирован от меньшего к большему  
            }  
        }  
  
        System.out.println("All list country in Western Europe");// весь список стран  
        System.out.println(directionContryUser);  
        System.out.println("-----");  
        return null;  
    }  
    catch (Exception e) // обработка ошибок(запрос / не тот тип данных)  
    {  
    }  
}
```

Результат -

```
2) The country with the smallest number of registered Internet users in Western Europe.  
Liechtenstein 37201  
All list country in Western Europe  
{Austria=7681957, Netherlands=15877494, Belgium=10021242, Monaco=37553, Luxembourg=570794, Liechtenstein=37201, France=59470000, Switzerland=7942864, Germany=
```

Для решения третьего задания использовалась таблица [people country]

По сути то же решение что и для второй задачи, только тут нет готовых значений.

Брались все значения стран и их населения с пользователями. Находился сразу процент пользователей. Записывался в лист для сортировки, в дальнейшем сравнивался со словарем и выводился результат. Так как не было четкого условия какие значения нужно выводить я вывел первые 3 от малого значения. Но в коде это можно легко поменять, просто заменив число на размер ключей и он выведет весь список.

```
static void percentageOfInternetUsersInCountries()
{
    try
    {
        Connection connection = DriverManager.getConnection("jdbc:sqlite:D:\\SQL Lite\\CountryJava.db");// подключение к бд
        Statement statement = connection.createStatement();

        ResultSet resultSet = statement.executeQuery("SELECT \"Country or area\" , \"Internet users\", Population FROM [people country] "
            + "WHERE Population IS NOT NULL and Population <= 1000000000");
        // запрос выглядит так - Выбрать страну, пользователей, популяцию ГДЕ популяция без нуля

        List<Double> listDouble = new ArrayList<Double>();// лист для процентов
        Map<String, Double> dictionaryPercent = new HashMap<String, Double>();// словарь с ключем Страна - значение процент

        while (resultSet.next())
        {
            String country = resultSet.getString("Country or area");//запись из колонки страна
            Integer users = Integer.parseInt(resultSet.getString("Internet users").replace(",","."));//запись из колонки пользователей
            Integer population = Integer.parseInt(resultSet.getString("Population").replace(",","."));// запись из колонки популяция
            Double percent = Double.parseDouble(String.format("%.3g", (double) users * 100.0 / (double) population).replace(",",".")); // итоговый процент пользователей
            if(percent >= 75.0 && percent <= 85.0)// условие задачи
            {
                listDouble.add(percent);// добавление в лист , для сортировки
                dictionaryPercent.put(country, percent);// добавление в словарь значений
            }
        }
        Collections.sort(listDouble);// сортировка, для того что бы было значение <= 85.0 на малое нужно перевернуть лист

        System.out.println("3) The percentage of registered Internet users of which is in the range from 75% to 85% (the first three)"); // я брал от малого к большому

        for (int i = 0; i < 3; i++)// взял 3 первых, можно поменять значение, например на размер ключей и выведет весь список
        {
            for (int j = 0; j < dictionaryPercent.keySet().size(); j++)// сверка листа и значений в ключе словаря для вывода
            {
                int number = i + 1;
                if (dictionaryPercent.get(dictionaryPercent.keySet().toArray()[j]) == listDouble.toArray()[i])
                {
                    System.out.println("    " + number + " "
                        + dictionaryPercent.keySet().toArray()[j]
                        + " " + listDouble.toArray()[i] + "%");
                }
            }
        }

        System.out.println("All list");
        System.out.println(dictionaryPercent);
        System.out.println("-----");
        return null;
    }
    catch (Exception e) // ловит ошибки с типом и запросом
    {
        System.out.println(e.getMessage());
        return null;
    }
}
```

Результат работы (так же как и в 2 задании я решил вывести все значения)-

```
3) The percentage of registered Internet users of which is in the range from 75% to 85% (the first three)
1) Greece 75.3%
2) Argentina 75.7%
3) Kazakhstan 75.9%
All list
{Singapore=83.7, Cyprus=80.1, Malaysia=80.4, Thailand=77.8, Kazakhstan=75.9, Oman=77.0, New Caledonia=80.9, Malta=78.5, Saudi Arabia=80.3, Greece=75.3, Cayman=75.0, ...}
```

8. И 9 этап был расписан в 7.