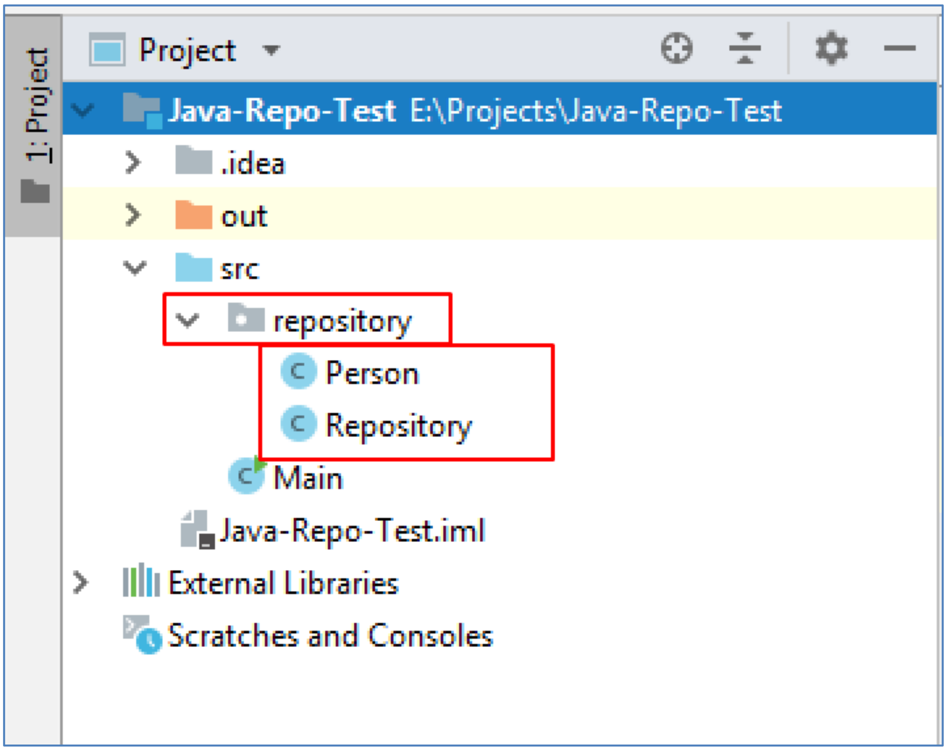
Problem 3. Repository

I. Project Structure

For this problem you should create a new package named **"repository",** which should hold inside the two classes **both Person and Repository.** The Main class can also be inside this package however it is not a must it may also be outside the package. Your project structure should look like that:



Pay attention to name the package, all the classes, their fields and methods exactly the same way they are presented in the following document. It is also important to keep the project structure as described above.

II. Person

Create Java class **`Person that has the following structure:`**

```
public class Person {
    // TODO: implement this class
}
```

1. Fields

- **name** – String
- **age** – int
- **birthDate** – String

The class **constructor** should receive all the fields parameters (**name**, **age**, **birthDate**).

2. Methods:

- Method **toString()** which returns the information about a single Person object in the following format:
  "Name: {name}"
  "Age: {age}"
  "Birthday: {birthDate}"

III. Repository

Write a Java class **Repository** that has **data field**, which stores objects of type Person with a corresponding unique **ID**, that is assigned when they are added **starting from zero**.

```
public class Repository {
    // TODO: implement this class
}
```

1. Fields

- **data** – Map<Integer, Person>

The class **constructor** should initialize the **data** with a new **Map** instance**.**

2. Methods

- Method **add(Person person)** – adds an Person to the data field with the next ID value
- Method **get(int id)** – returns the Person object stored with the given ID
- Method **update(int id, Person newPerson)** – replaces the Person stored to the coresponding ID with the new Person object. **Returns false** if the **ID doesn't exist, otherwise return true.**
- Method **delete(int id)** – deletes the Person object by the given id. **Return false if the id doesn't exist, otherwise return true.**
- Method **getCount()** – returns the number of stored Person objects.

Examples

This is an example how the **Repository** class is **intended to be used**. Make sure to comment out the parts that throw an error!

Sample code usage

```
public static void main(String[] args) {

    //Initialize the repository
    Repository repository = new Repository();

    //Initialize Person
    Person person = new Person("Pesho", 14, "13-07-2004");

    //Add two entities
    repository.add(person);

    //Initialize second Person object
    Person secondPerson = new Person("Gosho", 42, "21-09-1976");
    repository.add(secondPerson);

    System.out.println(repository.get(0).toString());
    //Name: Pesho
    //Age: 14
    //Birthday: 13-07-2004

    System.out.println(repository.get(1).toString());
    //Name: Gosho
    //Age: 42
    //Birthday: 21-09-1976


    //Update person with id 1
    repository.update(1, new Person("Success", 20, "01-01-1999"));

    System.out.println(repository.get(1).toString());
    //Name: Success
    //Age: 20
    //Birthday: 01-01-1999

    //Delete entity
    repository.delete(0);

    System.out.println(repository.getCount());
    //1
}
```

Constraints

- The ID should change **only** when we **add** a new Person.
- The ID is unique per repository – if two repositories are instantiated, each has its own counter.

Submission

- Submit **single .zip file**, containing **repository package, with the two classes inside (Person and Repository) and the Main class**, there is no specific content required inside the Main class e. g. you can do any kind of local testing of you program there. However there should be **main(String[] args)** method inside: