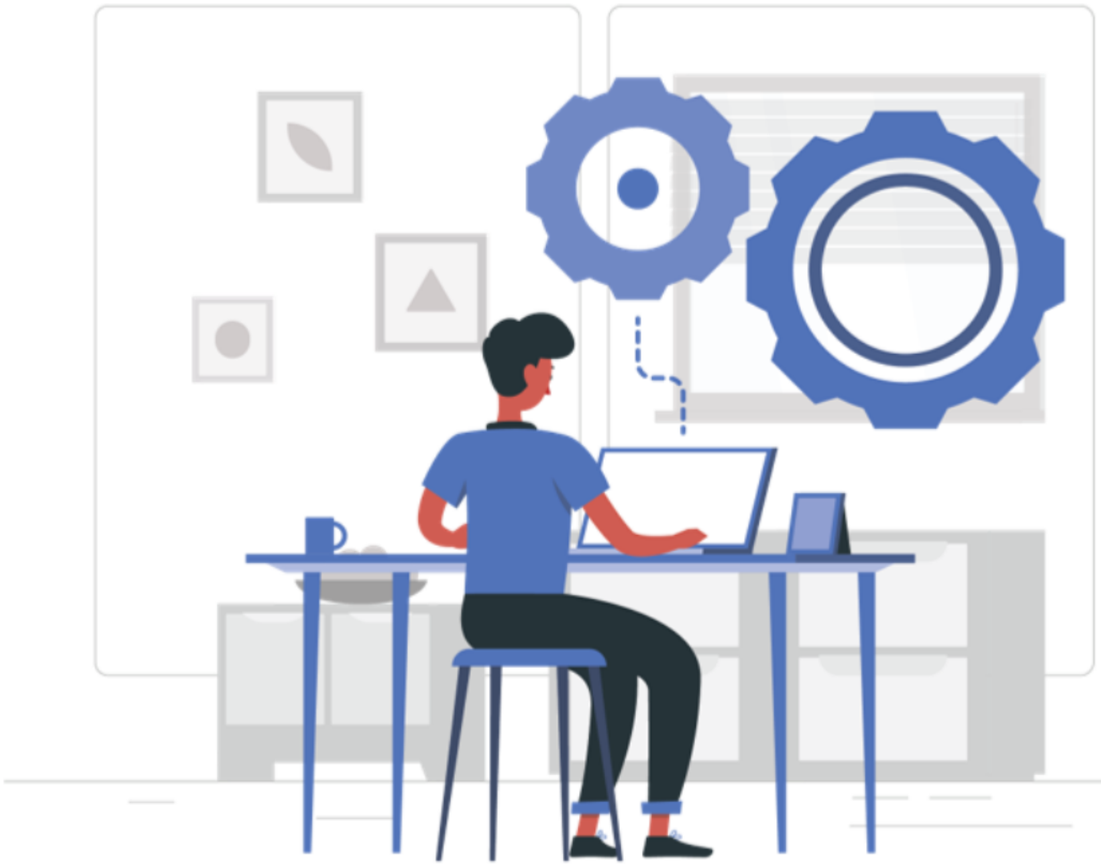

Challenge

Store and Manipulate Objects Using Ordered Collections

Implementation Environment

- The practice or challenge must be done in the IntelliJ IDE.
- Click here to install IntelliJ
- You must have access to GitLab.
- Install git to be able to clone and push code to the repository.
- You must be familiar with forking and cloning a git repository.

Mobile Service

Imagine that you wish to purchase a mobile phone with specifications including the brand name, cost, screen size, battery life, storage space, and camera pixel count. You have a list of the latest models available. Create a program that performs the following activities:

1. Find phones of a particular brand.
2. Find phones that cost $500 and above.
3. Find phones that have a camera specification of 12 MP or more.

CHALLENGE

Instructions for the Challenge

- Click on the boilerplate.
- Fork the boilerplate using the fork button 

- Select your namespace to fork the project.
- Clone the project into your local system.
- Open the project in the IntelliJ IDE.
- Work on the solution.
- Execute the test cases given in the test folder.
- Push the solution to git.

Tasks

- Define the `Mobile` class and its attributes like, `brandName, cost, screenSize, batteryLife,` etc.
- Generate Getter/Setter for all the attributes.
- Override the `toString()` method in the `Mobile` class.
- In the `MobileStore` class:
  - Create a list of type `Mobile`.
  - Read the `mobile.csv` file line by line, store the data in the respective attribute of the Mobile class by calling the setter method, and return the list of Mobile. Write the logic in the given method below:

```
public List<Mobile> readMobileData(String fileName)
```

Tasks (cont'd)

- Given a brand name as input, return the list of all mobiles of that brand. Write the logic in the method given below.

```
public List<Mobile> findPhoneByBrand(String brandName)
```

- Iterate through the list of mobile objects and return the phone that costs more than $500. Write the logic in the method given below and return the list.

```
public List<Mobile> findPhoneCostMoreThan$500()
```

Tasks (cont'd)

- Iterate through the list of mobile objects and return the phone that has a pixel count greater than 12MP. Write the logic in the method given below and return the list.

```
public List<Mobile> findPhonePixelMoreThan12MP()
```

Note: The `mobile.csv` file must be read only once, and the result must be stored in a List.