Day 9 – Assignment

Pratik K Kamble
Employee ID : 46263548
20 – 09 – 2022

Q1: Get the name of all females in upper case who are in age more than 6. Attempt this without java8 concepts. (Note : This is the imperative way of working).

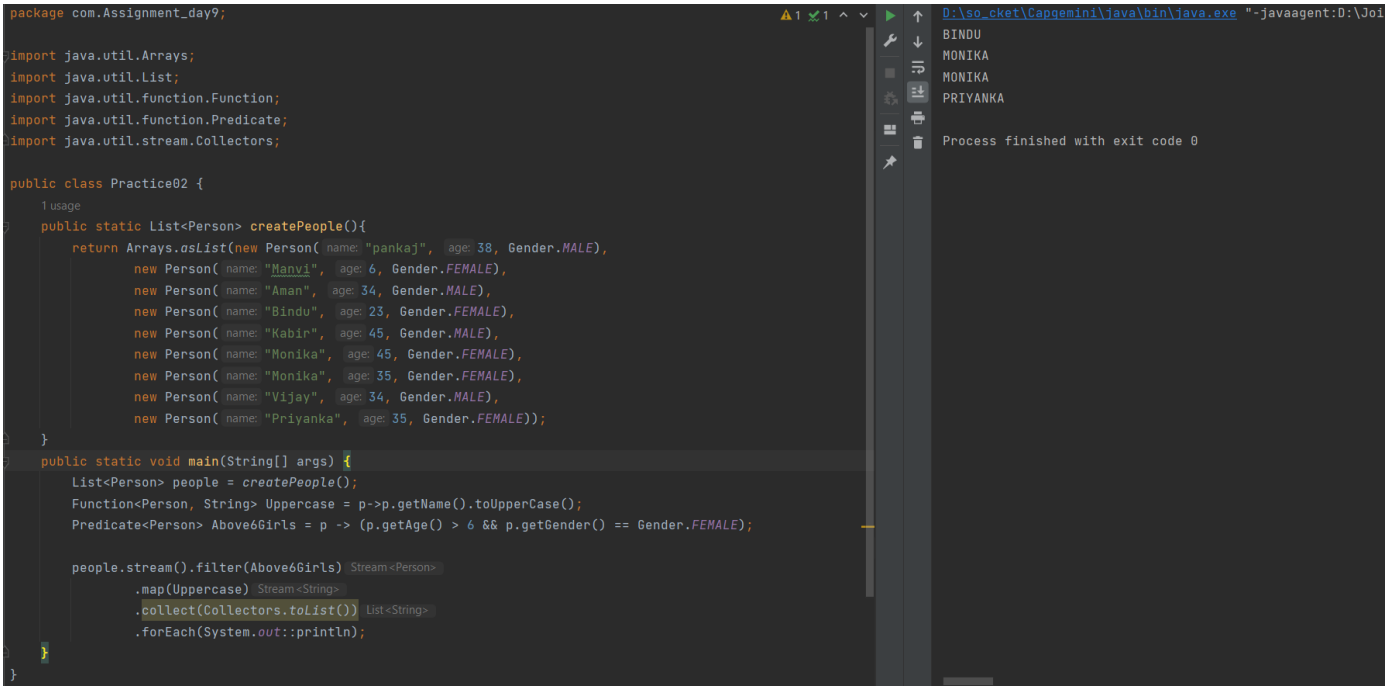
Solution –

```
package com.Assignment_day9;

import java.util.Arrays;
import java.util.List;

public class Practice01 {

    1 usage
    public static List<Person> createPeople(){
        return Arrays.asList(new Person( name: "pankaj",  age: 38, Gender.MALE),
                new Person( name: "Manvi",  age: 6, Gender.FEMALE),
                new Person( name: "Aman",  age: 34, Gender.MALE),
                new Person( name: "Bindu",  age: 23, Gender.FEMALE),
                new Person( name: "Kabir",  age: 45, Gender.MALE),
                new Person( name: "Monika",  age: 45, Gender.FEMALE),
                new Person( name: "Monika",  age: 35, Gender.FEMALE),
                new Person( name: "Vijay",  age: 34, Gender.MALE),
                new Person( name: "Priyanka",  age: 35, Gender.FEMALE));
    }

    public static void main(String[] args) {

        List<Person> people = createPeople();
        for(Person p : people)  {
            if(p.getAge() > 6 && p.getGender() == Gender.FEMALE) {
                System.out.println(p.getName());
            }
        }

    }
}
```

```
D:\so_cket\Capgemini\java\bin\java.exe "-javaagent:D:\Joining Capgemini\Plural
Bindu
Monika
Monika
Priyanka

Process finished with exit code 0
```

Q2: Get the name of all females in upper case who are in age more than 6. Attempt this by using filter, map and collect method of stream api. Use Predicate and Function functional interface as the argument of filter and map methods. (Note : This is the declarative way of working.)

Solution –

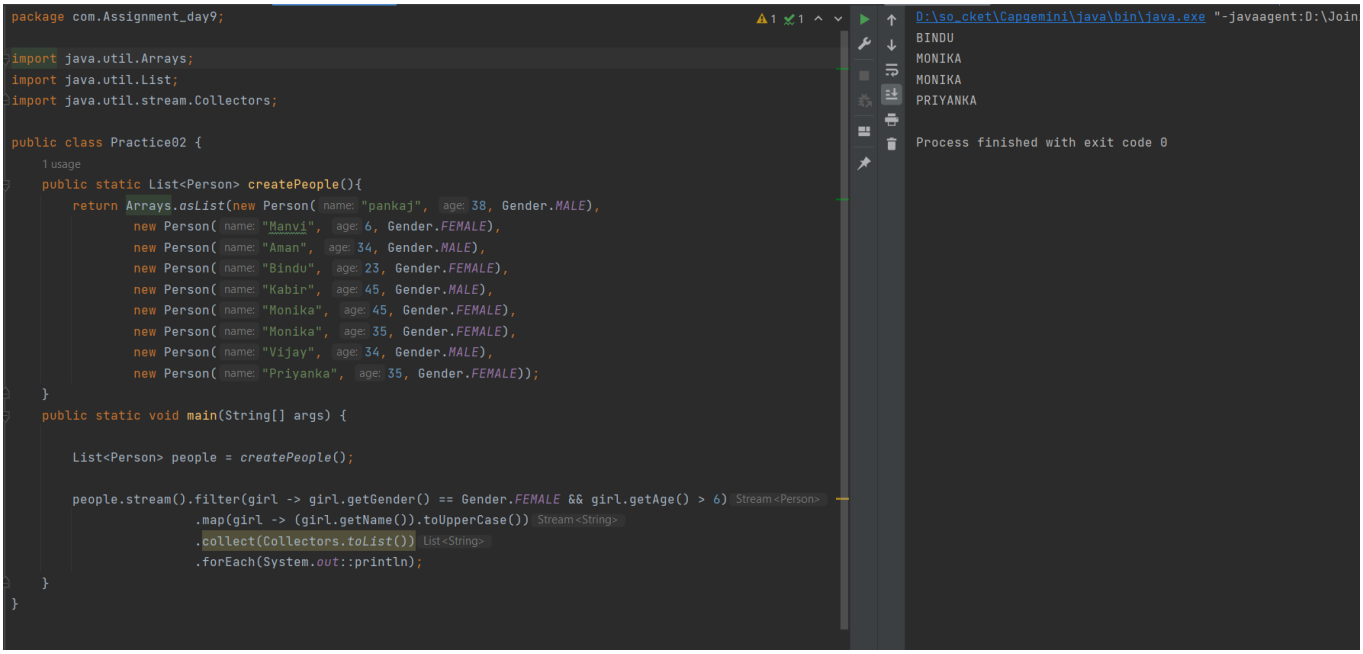


```
package com.Assignment_day9;

import java.util.Arrays;
import java.util.List;
import java.util.function.Function;
import java.util.function.Predicate;
import java.util.stream.Collectors;

public class Practice02 {
    1 usage
    public static List<Person> createPeople(){
        return Arrays.asList(new Person( name: "pankaj",  age: 38, Gender.MALE),
                new Person( name: "Manvi",  age: 6, Gender.FEMALE),
                new Person( name: "Aman",  age: 34, Gender.MALE),
                new Person( name: "Bindu",  age: 23, Gender.FEMALE),
                new Person( name: "Kabir",  age: 45, Gender.MALE),
                new Person( name: "Monika",  age: 45, Gender.FEMALE),
                new Person( name: "Monika",  age: 35, Gender.FEMALE),
                new Person( name: "Vijay",  age: 34, Gender.MALE),
                new Person( name: "Priyanka",  age: 35, Gender.FEMALE));
    }
    public static void main(String[] args) {
        List<Person> people = createPeople();
        Function<Person, String> Uppercase = p->p.getName().toUpperCase();
        Predicate<Person> Above6Girls = p -> (p.getAge() > 6 && p.getGender() == Gender.FEMALE);

        people.stream().filter(Above6Girls) Stream<Person>
                .map(Uppercase) Stream<String>
                .collect(Collectors.toList()) List<String>
                .forEach(System.out::println);
    }
}
```

```
D:\so_cket\Capgemini\java\bin\java.exe "-javaagent:D:\Joi
BINDU
MONIKA
MONIKA
PRIYANKA

Process finished with exit code 0
```

Q3: Get the name of all females in upper case who are in age more than 6. Attempt this by using filter, map and collect method of stream api, now in this instead of Predicate and Function functional interfaces, use Lambda expression. (More concise way of writing code using Java8)

Solution –



```
package com.Assignment_day9;

import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

public class Practice02 {
    1 usage
    public static List<Person> createPeople(){
        return Arrays.asList(new Person( name: "pankaj",  age: 38, Gender.MALE),
                new Person( name: "Manvi",  age: 6, Gender.FEMALE),
                new Person( name: "Aman",  age: 34, Gender.MALE),
                new Person( name: "Bindu",  age: 23, Gender.FEMALE),
                new Person( name: "Kabir",  age: 45, Gender.MALE),
                new Person( name: "Monika",  age: 45, Gender.FEMALE),
                new Person( name: "Monika",  age: 35, Gender.FEMALE),
                new Person( name: "Vijay",  age: 34, Gender.MALE),
                new Person( name: "Priyanka",  age: 35, Gender.FEMALE));
    }
    public static void main(String[] args) {

        List<Person> people = createPeople();

        people.stream().filter(girl -> girl.getGender() == Gender.FEMALE && girl.getAge() > 6) Stream<Person>
                       .map(girl -> (girl.getName()).toUpperCase()) Stream<String>
                       .collect(Collectors.toList()) List<String>
                       .forEach(System.out::println);
    }
}
```

```
D:\so_cket\Capgemini\java\bin\java.exe "-javaagent:D:\Join
BINDU
MONIKA
MONIKA
PRIYANKA

Process finished with exit code 0
```

Q4: You need to create a program which will check the extension of the files available in specified folder, and display the name of the filtered files.
Use the below approach for this exercise.
Approach# :With Lambda Function approach

- Step1: Create **FileFilterDemoUsingLambda** Class with main method
- Step2: Create a Reference of FileFilter.
- Step3: Reference of FileFilter should hold the lambda function that will hold the logic for FileFilter
- Step4: Using File IO display the filtered files from Files folder
- Step5: You must be creating the files.txt file in your app. Which must be holding the name of the files with some extension.

Solution –