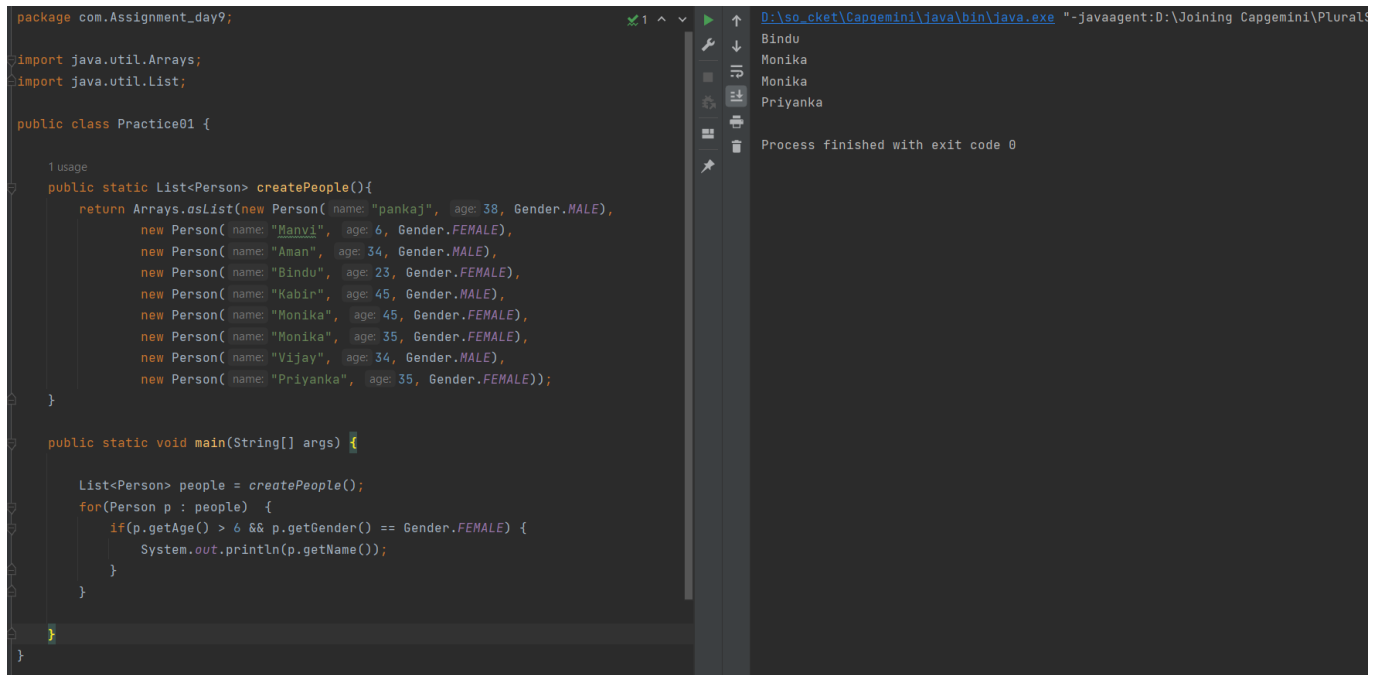


Day 9 – Assignment

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 –



The screenshot shows an IDE with a Java file named `Practice01.java` in the package `com.Assignment_day9`. The code defines a `Person` class with attributes `name`, `age`, and `gender`. It includes a `createPeople()` method that returns a list of `Person` objects and a `main` method that iterates through the list, filtering for females aged 6 and above, and printing their names in uppercase.

```
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());
            }
        }
    }
}
```

The output window on the right shows the execution of the program, displaying the names of the females: Bindu, Monika, Monika, and Priyanka. The 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 {

    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: "Kabin", 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())
            .forEach(System.out::println);
    }
}
```

0:\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:\socket\Capgemini\java\bin\java.exe "-javaagent:D:\Join
BINDU
MONIKA
MONIKA
PRIYANKA
Process finished with exit code 0

Use the below approach for this exercise.

- 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.

```
package com.Assignment_day9;

import java.io.File;
import java.io.FileFilter;
import java.io.FileReader;

public class FileFilterDemoUsingLambda {
    public static void main(String[] args) {
        FileFilter filefilter = file -> file.getName().endsWith(".txt");

        File dir = new File( pathname: "src/com/Assignment_day9");
        File[] files = dir.listFiles(filefilter);

        for(File fil:files) {
            System.out.println(fil);
        }
    }
}
```

Process finished with exit code 0

D:\socket\Capgemini\java\bin\java.exe -javaagent:D:\Jo...

src\com\Assignment_day9\a.txt
src\com\Assignment_day9\b.txt
src\com\Assignment_day9\c.txt
src\com\Assignment_day9\d.txt
src\com\Assignment_day9\e.txt