SCD-Lab

Lab#2

Name: Anas-Altaf

Roll.No: 22F-3639

Java Codes:

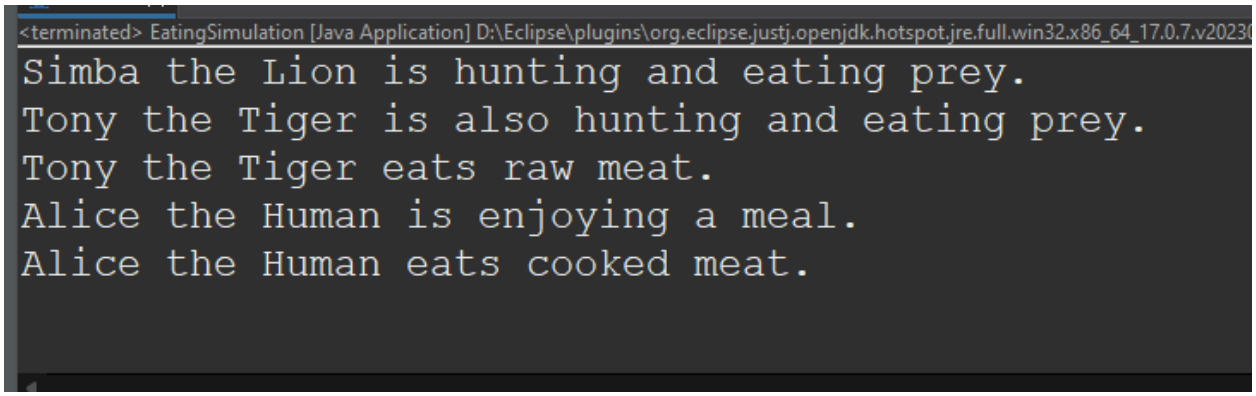
Task-1:

```
interface Eatable {
    void eats();
    void eatMeats();
}
abstract class Animal implements Eatable {
    private String name;
    Animal(String name) {
        this.name = name;
    }
    String getName() {
        return name;
    }
}
class Lion extends Animal {
    Lion(String name) {
        super(name);
    }
```

```
    public void eats() {
        System.out.println(getName() + " the
Lion eats raw meat.");
    }
    public void eatMeats() {
        System.out.println(getName() + " the
Lion is hunting and eating prey.");
    }
}
class Tiger extends Animal {
    Tiger(String name) {
        super(name);
    }
    public void eats() {
        System.out.println(getName() + " the
Tiger is also hunting and eating prey.");
    }
    public void eatMeats() {
        System.out.println(getName() + " the
Tiger eats raw meat.");
    }
}
class Human implements Eatable {
    private String name;
    Human(String name) {
        this.name = name;
    }
```

```
    public void eats() {
        System.out.println(name + " the
Human is enjoying a meal.");
    }
    public void eatMeats() {
        System.out.println(name + " the
Human eats cooked meat.");
    }
}
public class EatingSimulation {
    public static void main(String args[])
{
        Eatable[] khalqat = { new
Lion("Simba"),
                new Tiger("Tony"), new
Human("Alice"),
        };
        for (Eatable m : khalqat) {
            m.eats();
            m.eatMeats();
        }
    }
}
```

Output:



Task-2:

```
package Task_02;
public class Main {
    // For Null Pointer Exception
    private static void printLength(String
str) {
        System.out.println(str.length());
    }
    // For ArithmeticException
    private static void divideByArray(int
var1, int var2) {
        System.out.println((var1 / var2) + "
is the result of " + var1 + "/" + var2);
    }
    // For IndexOutOfBoundsException
    private static void printNullString() {
        int ar[] = { 1, 2, 3, 4, 5 };
```

```
        for (int i = 0; i <= ar.length; i++)
{
            System.out.println(ar[i]);
        }
    }
    public static void main(String[] args)
{
        try {
            // Try dividing by zero
            divideByArray(10, 0);
        } catch (ArithmeticException e) {
            System.out.println("Caught
ArithmeticException: " + e.getMessage());
        }
        try {
            // Try printing length of a null
string
            String myString = null;
            printLength(myString);
        } catch (NullPointerException e) {
            System.out.println("Caught
NullPointerException: " + e.getMessage());
        }
        try {
            // Try accessing an invalid index
in the array
            printNullString();
```

```
        } catch (IndexOutOfBoundsException
e) {
            System.out.println("Caught
IndexOutOfBoundsException: " +
e.getMessage());
        } finally {
            System.out.println("Program
Ended");
        }
    }
}
```

Output:

```
Caught ArithmeticException: / by zero
Caught NullPointerException: Cannot invoke
"String.length()" because "str" is null
1
2
3
4
5
Caught IndexOutOfBoundsException: Index 5
out of bounds for length 5
Program Ended
```

Task-3:

```
package Task_03;
import java.util.Scanner;
class NegativeValueException extends
Exception {
    public NegativeValueException(String
msg) {
        super(msg);
    }
}
public class Main {
    private static void
calculateSquareRoot(int input) throws
NegativeValueException {
        if (input < 0) {
            throw new
NegativeValueException("Can not take Square
Root of negative Number");
        } else {
            double result =
java.lang.Math.sqrt(input);
            System.out.println("Result is: "
+ result);
        }
    }
    public static void main(String args[])
{
```

```
        Scanner scanner = new
Scanner(System.in);
        try {
            System.out.println("Enter a
Number: ");
            int input = scanner.nextInt();
            calculateSquareRoot(input);
        } catch (NegativeValueException e) {
            System.out.println("Caught : " +
e.getMessage());
        } finally {
            System.out.println("Program
Ended");
            scanner.close();
        }
    }
}
```

Output:

```
Enter a Number:
-13
Caught : Can not take Square Root of
negative Number
Program Ended
```

Task-4:

```
package Task_04;
import java.util.Scanner;
interface LibraryMember {
    void borrowBook();
    void returnBook();
}
interface SportsTeamPlayer {
    void playSport();
    void attendPractice();
}
class UniversityStudent implements
LibraryMember, SportsTeamPlayer {
    private String name;
    private String studentId;
    private String favoriteSport;
    @Override
    public void playSport() {
        System.out.println(name + " plays "
+ favoriteSport);
    }
    @Override
    public void attendPractice() {
        System.out.println(name + " with Id
: " + studentId + " attends Practice.");
    }
```

```
    @Override
    public void borrowBook() {
        System.out.println(name + " with Id
: " + studentId + " borrows Book.");
    }
    @Override
    public void returnBook() {
        System.out.println(name + " with Id
: " + studentId + " returns Book.");
    }
    void takeInput() {
        Scanner scanner = new
Scanner(System.in);
        System.out.print("Enter name: ");
        this.name = scanner.nextLine();
        System.out.print("Enter student ID:
");
        this.studentId = scanner.nextLine();
        System.out.print("Enter favorite
sport: ");
        this.favoriteSport =
scanner.nextLine();
    }
    // Additional getters (optional, but
useful for testing)
    public String getName() {
        return name;
```

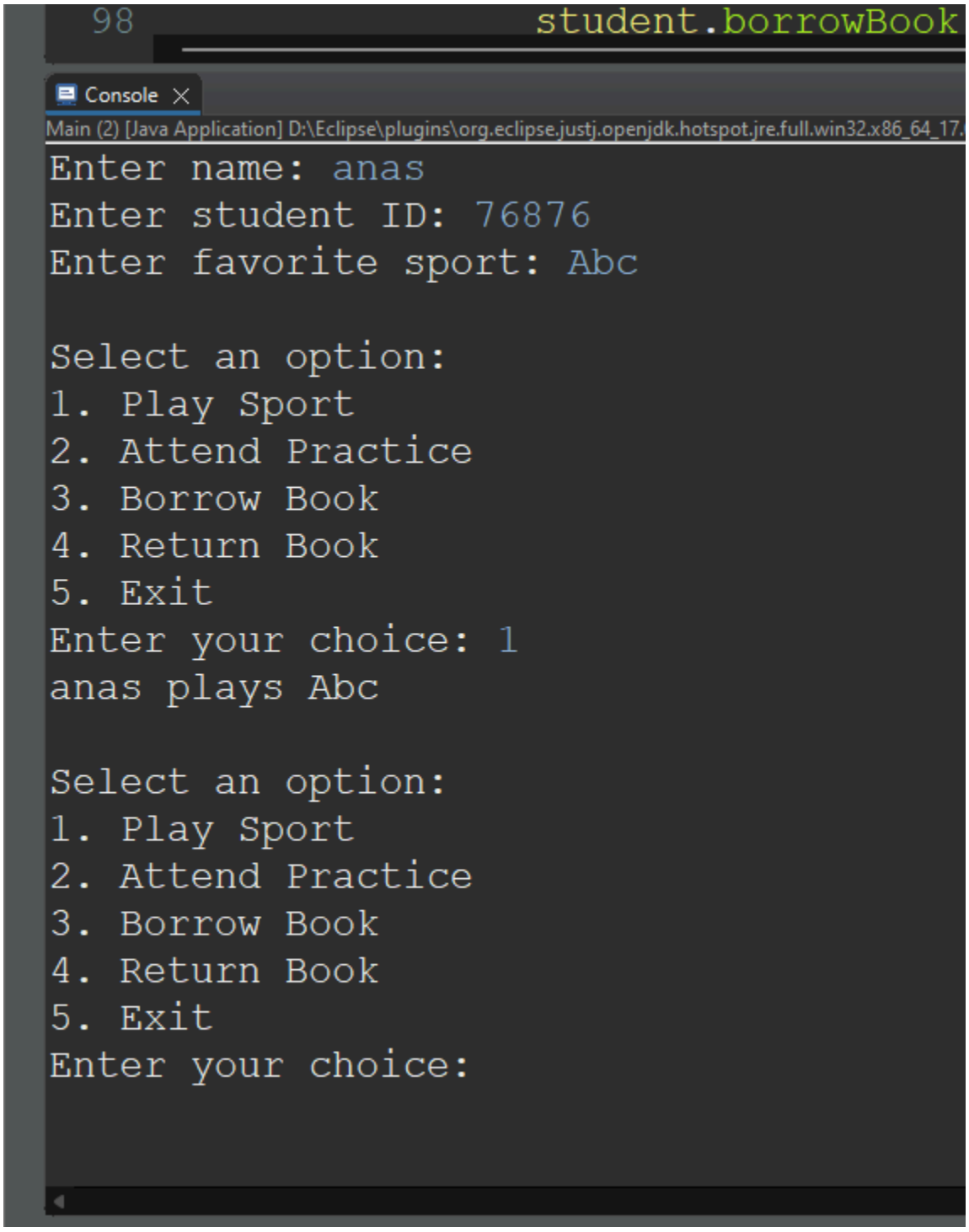
```
    }
    public String getStudentId() {
        return studentId;
    }
    public String getFavoriteSport() {
        return favoriteSport;
    }
}
public class Main {
    public static void main(String[] args)
{
        Scanner scanner = new
Scanner(System.in);
        UniversityStudent student = new
UniversityStudent();
        // Get student details
        student.takeInput();
        int choice;
        do {
            System.out.println("\nSelect an
option:");
            System.out.println("1. Play
Sport");
            System.out.println("2. Attend
Practice");
            System.out.println("3. Borrow
Book");
```

```
            System.out.println("4. Return
Book");
            System.out.println("5. Exit");
            System.out.print("Enter your
choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume
newline
            switch (choice) {
            case 1:
                student.playSport();
                break;
            case 2:
                student.attendPractice();
                break;
            case 3:
                student.borrowBook();
                break;
            case 4:
                student.returnBook();
                break;
            case 5:

System.out.println("Exiting...");
                break;
            default:
```

```
                System.out.println("Invalid
choice. Please enter a number between 1 and
5.");
            }
        } while (choice != 5);
        scanner.close();
    }
}
```

output:


98 student.borrowBook
Console
Main (2) [Java Application] D:\Eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.
Enter name: anas
Enter student ID: 76876
Enter favorite sport: Abc
Select an option:
1. Play Sport
2. Attend Practice
3. Borrow Book
4. Return Book
5. Exit
Enter your choice: 1
anas plays Abc
Select an option:
1. Play Sport
2. Attend Practice
3. Borrow Book
4. Return Book
5. Exit
Enter your choice: