THE UNIVERSITY OF SYDNEY

INFO1113 Week 10 Tutorial

Lambdas, Streams, Anonymous Classes

Lambdas

Lambda methods within java allows us to easily define a method that can be used through a functional interface type. Prior to defining a lambda, you will need to create a functional interface (an interface with only 1 abstract method) which your variables will use. When defining a variable, you can utilise the interface type and assign it to a lambda expression.

```
interface Say {
    public void yo(String str);
}
//...
public static void main(String[] args) {
    Say s = (x) -> System.out.println(x);
    s.yo("Good Morning!");
    s.yo("Good Night");
}
```

Question 1: Even or Odd

Write a simple program that utilises a lambda expression that will check if a number is even through the method name `isEven`. You will be required to create your own interface for the test that will allow you to construct a lambda expression.

```
public class EvenOrOdd {
    public static void main(String[] args) {
        /* Your interface here*/ t = //Your method here
        System.out.println(t.isEven(4));
        System.out.println(t.isEven(5));
        /*System.out.println(t.isOdd(10));
        System.out.println(t.isOdd(21));*/
    }
}
```

Extend the interface to contain a default method `isOdd` that can utilise the defined `isEven` method as part of its solution. Uncomment the isOdd lines and check if your default method works correctly.

Question 2: Loop to stream

Given the following code, rewrite it to utilise a stream. Utilise the Predicate and Consumer classes for your stream methods.

```
List<String> strings = new ArrayList<String>();
//..Add strings
for(String s : strings) {
    if(s.length() > 5) {
        System.out.println(s);
    }
}
```

Question 3: Rollercoaster

Using the stream object with a List, retrieve a list of people that are able to be admitted onto the roller coaster ride. The roller coaster ride has an age and height requirement. For someone to be admitted onto the ride, they will need to be older than 6 and 140cm.

You can utilise the following class

```
class Person {
    private String name;
    private int age;
    private double height;

    public Person(String name, int age, double height) {
        this.name = name;
        this.age = age;
        this.height = height;
    }

    //Add getters
}
```

Question 4: Comparator

Using the following class `Song`,Write an anonymous class/lambda that will allow you to sort the elements in a list of Songs with a certain property.

Implement the different sorting methods.

- Sort the list of songs using title
- Sort the list of songs using author

- Sort the list of songs using duration

```
class Song {
    private String title;
    private String author;
    private double duration;

    public Song(String title, String author, double duration) {
        this.title = title;
        this.author = author;
        this.duration = duration;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public double getDuration() {
        return duration;
    }
}
```

Question 5: Lambda Comparator

After finishing the previous exercise, rewrite your comparator objects with lambda expression for your program. The comparator interface is a functional interface that requires implementing just one method.

Question 6: Library

Write a library program that will allow you to add, remove, retrieve and borrow books.

Your address book program has the following commands:

```
FIND <title> - Outputs the book title, author, how many available
ADD <title> <author> <copies> - Add a new book, check if the name exists
DEL <title> - Removes a book from the library
BORROW <title> - Checks out a book from the library
```

You can use the following scaffold for your program:

```
import java.util.List;
import java.util.ArrayList;

class Book {
    //Define the fields for the Book
    public Book(String name, String author, int noCopies) {
        //Initialise them through the constructor
    }
    //Add getters and setters
}

public class Library {
    public static void main(String[] args) {
        List<Book> books = new ArrayList<Book>();
        books.add(new Book("To Kill A Mockingbird", "Harper Lee", 2));
        books.add(new Book("1984", "George Orwell", 3));
        books.add(new Book("A Brave New World", "Aldous Huxley", 1));
        books.add(new Book("The English Patient", "Michael Ondaatje", 2));
        books.add(new Book("The English Patient", "Michael Ondaatje", 2));
    }
}
```

Question 7: Assessed Task - Quiz 2

Remember you are required to complete the quiz next week. Go to Canvas page for this unit and click on Quizzes to find out the quiz and the due date. This is a marked assessment.