1. package assignment;

import java.util.\*;

/\*\*

\*

\* @author Darius Kirui

\*/

public class Assignment {

public static void main(String[] args) {

String phrase1 = "Anagram";

String phrase2 = "Nag a Ran";

/\*\*\* remove spaces from the strings \*\*\*/

phrase1 = phrase1.replaceAll("\\s", "");

phrase2 = phrase2.replaceAll("\\s", "");

/\*\* Convert phrases to lower case \*\*\*/

phrase1 = phrase1.toLowerCase();

phrase2 = phrase2.toLowerCase();

//compare the lengths

if(phrase1.length() != phrase2.length()){

System.out.println("No Anagram Phrases");

}

else{

//convert phrases to array list of characters

//Phrase 1

ArrayList<Character> chars1 = new ArrayList<Character>();

for (char c1 : phrase1.toCharArray()) {

chars1.add(c1);

}

//phrase2

ArrayList<Character> chars2 = new ArrayList<Character>();

for (char c2 : phrase2.toCharArray()) {

chars2.add(c2);

}

//count instances of items

//Compare the arraylists

String anagramoutput = "";

for(int i=0;i<chars1.size();i++){

if(chars1.contains(chars2.get(i))){

int occurence = Collections.frequency(chars1, chars1.get(i));

int occurence2 = Collections.frequency(chars2, chars1.get(i));

if(occurence == occurence2){

anagramoutput = "Anagram Phrases";

}

else{

anagramoutput = "No Anagram Phrases";

break;

}

}else{

anagramoutput = "No Anagram Phrases";

break;

}

}

System.out.println(anagramoutput);

}

}

}

1. **Exceptions**

Checked Exceptions - Exceptions checked at compile time. If some code within a method

throws a checked exception, then the method must either handle the exception using

throw keyword, try and catch clauses etc.

examples of real-word usage:

-checking if the database server is available

-checking if a file exists in FileReader

-Checking SQL Language Statements (DML,DCL,DDL,TCL)

-etc

Unchecked Exceptions - Exceptions that are not checked at compile time. It means if your program

is throwing an unchecked exception and even if you didn’t handle/declare that exception, the program

won’t give a compilation error. Most of the times these exception occurs due to the bad data provided

by user during the user-program interaction. It is up to the programmer to judge the conditions in

advance, that can cause such exceptions and handle them appropriately. Try and Catch statements

can be used to return the exception. All Unchecked exceptions are direct sub classes of RuntimeException

class

examples in real-world;

-NullPointerException

-ArrayIndexOutOfBoundsException

-ArithmeticException

-IllegalArgumentException

-NumberFormatException

1. **First Case prints false and the second case prints true**

In java Integer type is an object.

In java 1.5 and above, Integer objects with the values -128 to 127 are immutable (that is, for one particular integer value, say 100, all integer objects instantiated through your program with the value 100 points to the exact same object because it is between the range of -128 to 127). For both to be true one requires to use Integer.equals method when comparing integer objects.

e.g.

System.out.println(a.equals(b));

4. MVC in Javascript

Model - The model is responsible for managing application data. It responds to the request from view and to the instructions from controller to update itself.

|  |
| --- |
| (function() { |
|  |  |
|  | "use strict"; |
|  |  |
|  | var App = angular.module("App.services",[]); |
|  |  |
|  | App.service('Model', function() { |
|  | this.getLinks = function() { |
|  | return [{ |
|  | id: 1, |
|  | title: 'Item A' |
|  | }, { |
|  | id: 2, |
|  | title: 'Item B' |
|  | }, { |
|  | id: 3, |
|  | title: 'Item C' |
|  | }, { |
|  | id: 4, |
|  | title: 'Item D' |
|  | }, { |
|  | id: 5, |
|  | title: 'Item E' |
|  | }, { |
|  | id: 6, |
|  | title: 'Item F' |
|  | } |
|  | ]}; |
|  | }); |
|  |  |
|  | }()); |

View - A presentation of data in a particular format, triggered by the controller's decision to present the data.

<h1>VIEW 2</h1>

Controller - The controller responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

|  |
| --- |
| (function() { |
|  |  |
|  | "use strict"; |
|  |  |
|  | var App = angular.module("App.controllers", []); |
|  |  |
|  | App.controller("MyCtrl1", ["$scope", function ($scope, UtilSrvc){ |
|  | $scope.aVariable = 'anExampleValueWithinScope'; |
|  | $scope.valueFromService = UtilSrvc.helloWorld("User"); |
|  | }]); |
|  |  |
|  | App.controller("MyCtrl2", ["$scope", function($scope){ |
|  | // if you have many controllers, it's better to separate them into files |
|  | }]); |
|  |  |
|  | }()); |

5. Real World examples of when to use ArrayList and LinkedList

Arraylist – This is used where an item in the list can be accessed randomly by use of an index/position

* School Records
* Online shopping Cart
* Medical Records

Linkedlist – This is used where items are links to each other e.g. first item is link to the second item.

* Music Playlist
* Queue
* Operating System Memory Allocation
* Pagination
* Browser history

6. Benefits of using spring:

* Spring MVC is a framework for java web applications with the following in-build features

1. Special form tags validation out of the box
2. Stateful events

* It exposes restful services
* It provides secure web applications through Spring Security
* Provides Spring Object Relational Mapping for communicating with database. This is a way to represent tables into objects and vice-versa.

**Spring MVC Subprojects**

* **Spring Cloud Security** - Provides support for load-balanced OAuth2 rest client and authentication header relays in a Zuul proxy.
* **Spring Cloud for Amazon Web Services -** Easy integration with hosted Amazon Web Services. It offers a convenient way to interact with AWS provided services using well-known Spring idioms and APIs, such as the messaging or caching API. Developers can build their application around the hosted services without having to care about infrastructure or maintenance.
* **Spring Data JDBC –** JDBC-based repositories
* **Spring Data Rest -** Exports Spring Data repositories as hypermedia-driven RESTful resources.

7.

8. Dependency Injection

Dependency Injection is a Design Pattern of objects creating a dependency or asking a factory object to make one for them, you pass the needed dependencies in to the constructor or via property setters, and you make it somebody else's problem (an object further up the dependency graph, or a dependency injector that builds the dependency graph).

**Injecting Bean in Spring**

You inject bean in spring using value annotation

@Value("${propName}")

private String str;

9. Aspects in Spring

package intellisoft;

@Aspect

public class saveannotation {

/\*\* Check parameters method\*\*\*/

public checkParameters(declare parameters){

//code for checking parameters

//call save method

}

public save(){

//database connection and database exceptions

}

/\*\*\* save method \*\*\*/

}

10. Benefits of Hibernate framework

- Supports Inheritance, Association and Collection

- All Exceptions are un-checked

- It’s database independent

- It supports caching mechanism

- Supports annotations