DSA coursework report

Federico Vivaldo S1828951

All the classes have been tested using Junit, with appropriate class testing techniques. In each class test every single method has been tested, producing output where required.

Track class test:

In this test class there are 16 tests with only one showing some kind of output.A screenshot of a cell phone

Description automatically generated

All the test except the toString method are conducted by using the assertEquals method which compares some expected output with some actual output of each method.

The toString method uses a print console statement display a formatted version of all the variables values in a track object instance.

TrackList class test:

Different testing techniques are adopted, junit testing has still been used.

Most of the tested methods produce an output to the console, again each method is tested individually.

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

On the output side of the console, each method tested has a title to identify and divide the testing, some asserequals method are also used whenever required.

TrackBST class test:

This junit class test also produces some output to the console testing each piece of method in the class.A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

All the different test method are dived and organised to the console output with appropriate headings.

Each clss test can be runned several times, all the test classes are located in the test packages of the project.

Error prevention:

On task four several methods have been implemented to prevent accidental errors and mistakes from the user. All the methods are stored in the ReadConsole class which has the job of taking the console input and read it to perform some actions within the Menu class.

The user can enter on the console specific characters that corresponds to a menu option, each character is processed from the system as a capital letter with a to uppercase method in this way the user can enter either a small or capital letter, the system will still process it.

The characters are also checked whether they are within the range of the options, in the case a character is not an option the system will provide an error feedback suggesting the user to try again with an existing option.

A screenshot of a cell phone

Description automatically generated

For the yes and no options a special version of checkChar method is implemented in this special case the console will accept only 1 of two values Y for yes and N for no, providing appropriate feedback in case of mistake.A screenshot of a cell phone

Description automatically generated

The system also checks if the input has more than 1 characters providing appropriate feedback in case of error.

A screenshot of text

Description automatically generated

Javadoc documentation has also been provided and can be found in the distribution folder of the project under index.html file.

A screenshot of a computer screen

Description automatically generated