EpiphanyEngine

Class Engine

java.lang.Object
    EpiphanyEngine.Engine

```
public class Engine
extends java.lang.Object
```

This class can be instantiated to perform all the operations that the program needs to perform. All the tasks would be stored in different projects of type project. Within these projects, we would have an ArrayList of Task which is used to store the instruction, the date as well as the name of the project that it belongs to. The interpreter would pass in the commands from the user and this class would be used to store, modify and update the projects that the user creates. It currently has the abilty to add, delete, search and display. Additionally, it has complete project management. Missing: Storing projects in separate files (save).

Author:

Moazzam

Field Summary

Fields

Modifier and Type	Field and Description
static java.util.ArrayList<EpiphanyEngine.Task>	**defaultProject**
static java.util.ArrayList<EpiphanyEngine.Project>	**EpiphanyMain**
static java.lang.String	**MESSAGE_ADD**
static java.lang.String	**MESSAGE_CLEAR**
static java.lang.String	**MESSAGE_CLEAR_EMPTY**
static java.lang.String	**MESSAGE_DELETE**
static java.lang.String	**MESSAGE_DELETE_INVALID**
static java.lang.String	**MESSAGE_DISPLAY**
static java.lang.String	**MESSAGE_DISPLAY_ERROR**
static java.lang.String	**MESSAGE_EXIT**
static java.lang.String	**MESSAGE_INVALID_SEARCH**
static java.lang.String	**MESSAGE_NO_ENTRY**
static java.lang.String	**MESSAGE_PROVIDE_ARGUMENT**
static java.lang.String	**MESSAGE_SORT**
static java.lang.String	**MESSAGE_SORTED**
static java.lang.String	**MESSAGE_WRONG_ENTRY**
static java.util.ArrayList<java.lang.String>	**projectNames**
static java.util.ArrayList<java.util.Date>	**testDate1**
static java.util.ArrayList<EpiphanyEngine.Task>	**testDateSort**

Constructor Summary

Constructors

Constructor and Description
Engine()

Method Summary

Modifier and Type	Method and Description
void	**deleteProject**(java.util.ArrayList<EpiphanyEngine.Task> projectName) Finds the project are removes it entirely.
void	**displayAll**() A function that helps to display all the projects including all the tasks stored within them.
void	**displayProject**(EpiphanyEngine.Project tempName) Displays all the tasks within any one of the projects.
void	**displayProjects**() Displays the names and indices of all the projects that exist.
void	**executeCommand**(CommandType userCommand) Interpreter passes in a command type object.
java.util.Date	**formatDate**(java.lang.String dateStr) This function helps to format the date.
void	**printToUser**(java.lang.String text, java.lang.String arg1, java.lang.String arg2)
java.lang.String	**removeFirstWord**(java.lang.String userCommand)
void	**sortDateInList**(java.util.ArrayList<java.util.Date> testDate) Sorts the date.
java.lang.String	**toString**(java.util.Date deadLine) Helps to format the date and return the formatted date.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

Field Detail

projectNames

public static java.util.ArrayList<java.lang.String> projectNames

defaultProject

public static java.util.ArrayList<EpiphanyEngine.Task> defaultProject

EpiphanyMain

public static java.util.ArrayList<EpiphanyEngine.Project> EpiphanyMain

testDate1

public static java.util.ArrayList<java.util.Date> testDate1

testDateSort

public static java.util.ArrayList<EpiphanyEngine.Task> testDateSort

MESSAGE_WRONG_ENTRY

public static final java.lang.String MESSAGE_WRONG_ENTRY

See Also:
Constant Field Values

MESSAGE_SORTED

public static final java.lang.String MESSAGE_SORTED

See Also:
Constant Field Values

MESSAGE_DELETE_INVALID

public static final java.lang.String MESSAGE_DELETE_INVALID

See Also:
Constant Field Values

MESSAGE_NO_ENTRY

public static final java.lang.String MESSAGE_NO_ENTRY

See Also:
Constant Field Values

MESSAGE_DELETE

public static final java.lang.String MESSAGE_DELETE

See Also:
Constant Field Values

MESSAGE_CLEAR_EMPTY

public static final java.lang.String MESSAGE_CLEAR_EMPTY

See Also:
Constant Field Values

MESSAGE_ADD

public static final java.lang.String MESSAGE_ADD

See Also:
Constant Field Values

MESSAGE_DISPLAY_ERROR

public static final java.lang.String MESSAGE_DISPLAY_ERROR

See Also:
Constant Field Values

MESSAGE_CLEAR

public static final java.lang.String MESSAGE_CLEAR

See Also:
Constant Field Values

MESSAGE_DISPLAY

public static final java.lang.String MESSAGE_DISPLAY

See Also:
Constant Field Values

MESSAGE_EXIT

`public static final java.lang.String MESSAGE_EXIT`

See Also:

MESSAGE_SORT

`public static final java.lang.String MESSAGE_SORT`

See Also:

MESSAGE_INVALID_SEARCH

`public static final java.lang.String MESSAGE_INVALID_SEARCH`

See Also:

MESSAGE_PROVIDE_ARGUMENT

`public static final java.lang.String MESSAGE_PROVIDE_ARGUMENT`

See Also:

Constructor Detail

Engine

`public Engine()`

Method Detail

displayAll

`public void displayAll()`

A function that helps to display all the projects including all the tasks stored within them.

displayProject

`public void displayProject(EpiphanyEngine.Project tempName)`

Displays all the tasks within any one of the projects.

Parameters:

`name - The name of the project that we wish to display.`

displayProjects

`public void displayProjects()`

Displays the names and indices of all the projects that exist.

deleteProject

`public void deleteProject(java.util.ArrayList<EpiphanyEngine.Task> projectName)`

Finds the project are removes it entirely.

Parameters:

projectName – is the name of the project.

---

formatDate

public java.util.Date formatDate(java.lang.String dateStr)

This function helps to format the date.

Parameters:

dateStr – String that contains the date

Returns:

formatted date.

---

sortDateInList

public void sortDateInList(java.util.ArrayList<java.util.Date> testDate)

Sorts the date.

Parameters:

testDate – An ArryList of date that needs to be sorted.

---

toString

public java.lang.String toString(java.util.Date deadLine)

Helps to format the date and return the formatted date.

Parameters:

deadLine –

Returns:

the formatted date.

---

executeCommand

public void executeCommand(CommandType userCommand)

Interpreter passes in a command type object. This method determines which type of command it is and uses the appropriate methods using the switch statements.

Parameters:

userCommand – is the command that the interpreter send in.

---

removeFirstWord

public java.lang.String removeFirstWord(java.lang.String userCommand)

---

printToUser

public void printToUser(java.lang.String text,
                        java.lang.String arg1,
                        java.lang.String arg2)