

Package model

Class Summary

[Chore](#)

This class is responsible for creating a Chore object.

[ChoresAllocation](#)

This class is responsible for creating a ChoresAllocation object.

[ChoresList](#)

This class is responsible for creating an object which can store a list of Chores.

[Person](#)

This class is responsible for creating a Person object.

[Profile](#)

This class is responsible for creating a Profile object.

[ProfileData](#)

This class is responsible for creating a ProfileData object.

model

Class Chore

```
java.lang.Object
|
+--model.Chore
```

All Implemented Interfaces:

java.lang.Comparable

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Chore
extends java.lang.Object
implements java.lang.Comparable
```

This class is responsible for creating a Chore object. Implements Comparable to allow elements to be compared.

Author:

Yashwant Rathor

Fields

choreName

```
private java.lang.String choreName  
    Name of the chore.
```

choreTime

```
private int choreTime  
    Time taken to complete chore.
```

Constructors

Chore

```
public Chore(java.lang.String choreName,  
             int choreTime)  
    throws java.lang.IllegalArgumentException
```

Custom constructor. Creates a Chore object with two input parameters.

Parameters:

choreName - Name of the chore.
choreTime - Time taken to complete the chore.

Throws:

java.lang.IllegalArgumentException - If choreName is empty or null.

Methods

compareTo

```
public int compareTo(Chore other)  
    Compares two Chore objects with each other.
```

equals

```
public boolean equals(java.lang.Object obj)  
    Checks if two Chore objects have the same values.  
Overrides:  
    equals in class java.lang.Object
```

getChoreName

```
public java.lang.String getChoreName()
```

Allows the choreName value to be accessed.

Returns:

returns choreName.

getChoreTime

```
public int getChoreTime()
```

Allows the choreTime value to be accessed.

Returns:

returns the choreTime.

toString

```
public java.lang.String toString()
```

Overrides the default toString() representation of this class.

Overrides:

toString in class java.lang.Object

model

Class ChoresAllocation

```
java.lang.Object  
|  
+--model.ChoresAllocation
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

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

This class is responsible for creating a ChoresAllocation object. The object uses the TreeMap data structure to store a Chore object and an ArrayList[String].

Author:

Yashwant Rathor

Fields

tMap

```
private java.util.TreeMap tMap
    TreeMap
```

Constructors

ChoresAllocation

```
public ChoresAllocation()
```

Default Constructor. Creates an empty TreeMap.

Methods

addAllocation

```
public void addAllocation(Chore c,
                           java.util.ArrayList al)
```

Stores a Chore object and an ArrayList[String] to the TreeMap.

Parameters:

c - Chore to be added as key to the {@link tMap} field.

al - ArrayList[String] to be added as value linked to a key, to the {@link tMap} field.

getChores

```
public java.util.Collection getChores()
```

Returns a collection of TreeMap keys.

Returns:

returns all of the keys stored within the TreeMap.

getPeople

```
public java.util.Collection getPeople()
```

Returns a collection of TreeMap values.

Returns:

returns all of the values stored within the TreeMap.

returnIterator

```
public java.util.Iterator returnIterator()
```

Allows elements within the TreeMap to be iterated through.

Returns:

returns an iterator over the elements in this TreeMap's set.

toString

```
public java.lang.String toString()
```

Overrides the default toString() representation of this class.

Overrides:

toString in class java.lang.Object

model

Class ChoresList

```
java.lang.Object
|
+--model.ChoresList
```

All Implemented Interfaces:

java.lang.Iterable

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class ChoresList
extends java.lang.Object
implements java.lang.Iterable
```

This class is responsible for creating an object which can store a list of Chores. Implements Iterable to allow elements to be iterated through.

Author:

Yashwant Rathor

Fields

choresList

```
private java.util.ArrayList choresList
    List of chores.
```

Constructors

ChoresList

```
public ChoresList()
```

Default Constructor.

Methods

addChore

```
public void addChore(Chore c)
```

Adds a non-null Chore object to the list.

Parameters:

c - the Chore object to be added to {@link choresList} field.

choresListSize

```
public int choresListSize()
```

Allows the size of the choresList to be returned.

Returns:

returns the size of the list.

getChore

```
public Chore getChore(int index)
```

Allows a Chore object to be retrieved from the choresList.

Parameters:

index - the index to be accessed from the choresList.

Returns:

returns the Chore object located at the supplied index.

iterator

```
public java.util.Iterator iterator()
```

Allows elements within choresList to be iterated through.

toString

```
public java.lang.String toString()
```

Overrides the default toString() representation of this class.

Overrides:

toString in class java.lang.Object

model

Class Person

```
java.lang.Object
|
+--model.Person
```

All Implemented Interfaces:

java.lang.Comparable

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class Person
  extends java.lang.Object
  implements java.lang.Comparable
```

This class is responsible for creating a Person object. Implements Comparable to allow elements to be compared.

Author:

Yashwant Rathor

Fields

ID

```
private int ID
  An identification number unique to each person.
```

firstName

```
private java.lang.String firstName
  A person's forename.
```

lastName

```
private java.lang.String lastName  
    A person's surname.
```

nickName

```
private java.lang.String nickName  
    A person's nickname.
```

Constructors

Person

```
public Person(java.lang.String firstName,  
              java.lang.String lastName,  
              int ID)  
    throws java.lang.IllegalArgumentException
```

Custom Constructor with no nickname field parameter.

Parameters:

firstName - Forename of a person.
lastName - Surname of a person.
ID - Identification number of a person.

Throws:

java.lang.IllegalArgumentException - If firstName or lastName parameters are empty or null.

Person

```
public Person(java.lang.String firstName,  
              java.lang.String lastName,  
              java.lang.String nickName,  
              int ID)  
    throws java.lang.IllegalArgumentException
```

Custom Constructor 2 with nickname field parameter

Parameters:

firstName - Forename of a person.
lastName - Surname of a person
nickName - Nickname of a person.
ID - Identification number of a person.

Throws:

java.lang.IllegalArgumentException - If firstName, lastName or nickName parameters are empty or null.

Methods

compareTo

```
public int compareTo(Person other)
```

Compares two Person objects with each other.

equals

```
public boolean equals(java.lang.Object obj)
```

Checks if two Person objects have the same values.

Overrides:

equals in class java.lang.Object

getFirstName

```
public java.lang.String getFirstName()
```

Ensures that the firstName is returned with the correct formatting.

Returns:

returns firstName with every word capitalised.

getFullName

```
public java.lang.String getFullName()
```

Allows a full name String to be generated from a person's forename, surname and nickname.

Returns:

a combined String which represents a person's full name.

getID

```
public int getID()
```

Allows a person's ID number to be accessed.

Returns:

returns the unique ID value.

getLastName

```
public java.lang.String getLastName()
```

Ensures that the lastName is returned with the correct formatting.

Returns:

returns lastName with every word capitalised.

getNickName

```
public java.lang.String getNickName()
```

Ensures that the nickName is returned with the correct formatting.

Returns:

returns nickName with every word capitalised.

toString

```
public java.lang.String toString()
```

Overrides the default toString() representation of this class.

Overrides:

toString in class java.lang.Object

model

Class Profile

```
java.lang.Object
|
+--model.Profile
```

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

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

This class is responsible for creating a Profile object.

Author:

Yashwant Rathor

Fields

averageChoreTime

```
private int averageChoreTime
```

An int value to store the average time needed to complete a chore.

chores

```
private ChoresList chores
```

A ChoresList object.

currentDateLine

```
private java.lang.String currentDateLine
```

A String value to represent the current Date line for a text file.

filePath

```
private java.lang.String filePath
```

A String value to store the filepath to where the text file will be created and written to.

formattedAverageTime

```
private java.lang.String formattedAverageTime
```

A String value to store the formatted average time.

formattedTotalTime

```
private java.lang.String formattedTotalTime
```

A String value to store the formatted combined time.

noOfChores

```
private int noOfChores
```

An int value to store the number of chores for each Profile.

person

```
private Person person
```

A Person object.

separator

```
private java.lang.String separator
```

A String value to represent a (-) division of a dynamic length, calculated from titleLine's length.

separator2

```
private java.lang.String separator2
```

A String value to represent a (-) division of a dynamic length, calculated from formatted time String lengths.

titleLine

```
private java.lang.String titleLine
```

A String value to represent the title line for a text file.

totalChoreTime

```
private int totalChoreTime
```

An int value to store the combined time needed to complete all chores.

totalLine

```
private java.lang.String totalLine
```

A String value to represent the total line for a text file.

useNickName

```
private java.lang.Boolean useNickName
```

A Boolean value for whether a nickname is to be used or not.

wcDateLine

```
private java.lang.String wcDateLine
```

A String value to represent the week commencing Date line for a text file.

Constructors

Profile

```
public Profile(Person person,  
              java.lang.Boolean useNickName)  
    throws java.lang.NullPointerException
```

Custom Constructor.

Parameters:

person - A Person object to be assigned to person.

useNickName - A Boolean value to be assigned to useNickName.

Throws:

java.lang.NullPointerException - If person or useNickname are null.

Methods

addChore

```
public void addChore(Chore c)
```

Adds a non-null Chore object to the list.

Parameters:

c - the Chore object to be added to {@link chores} field.

getAllChores

```
public ChoresList getAllChores()
```

Allows a list of chores to be accessed.

Returns:

returns a list of all chores for a profile.

getAverageChoreTime

```
public int getAverageChoreTime()
```

Allows the average chore time to be accessed.

Returns:

returns the averageChoreTime value.

getAverageChoreTimeOutput

```
public java.lang.String getAverageChoreTimeOutput()
```

Allows the average time required to complete a chore to be accessed.

Returns:

returns the formattedAverageTime, which is the averageChoreTime in hours and minutes.

getCurrentDateLine

```
public java.lang.String getCurrentDateLine()
```

Allows a profile's current Date line to be accessed.

Returns:

returns the String for the currentDateLine to be inputed in profile's text file.

getFilePath

```
public java.lang.String getFilePath()
```

Allows a profile's filepath to be accessed.

Returns:

returns the String of the filepath to store data for a profile.

getNoOfChores

```
public int getNoOfChores()
```

Allows the number of chores allocated to a profile to be accessed.

Returns:

returns the number of chores for a profile.

getPerson

```
public Person getPerson()
```

Allows person's details to be accessed.

Returns:

returns a Person object.

getSeparator

```
public java.lang.String getSeparator()
```

Allows the separator String value to be accessed.

Returns:

returns a String with a length dependent on the length of the titleLine.

getSeparator2

```
public java.lang.String getSeparator2()
```

Allows the separator2 String value to be accessed.

Returns:

returns a String with a length dependent on the length of the formatted time Strings.

getTitle

```
public java.lang.String getTitle()
```

Allows a profile's title line to be accessed.

Returns:

returns the String for the titleLine to be inputed in the profile's text file.

getTotalChoreTime

```
public int getTotalChoreTime()
```

Allows the combined total chore time to be accessed.

Returns:

returns the totalChoreTime value.

getTotalChoreTimeOutput

```
public java.lang.String getTotalChoreTimeOutput()
```

Allows the total time required to complete all chores to be accessed.

Returns:

returns the formattedTotalTime, which is the totalChoreTime in hours and minutes.

getTotalLine

```
public java.lang.String getTotalLine()
```

Allows a profile's total line to be accessed.

Returns:

returns the String for the totalLine to be inputed in profile's text file.

getWCDateLine

```
public java.lang.String getWCDateLine()
```

Allows a profile's week commencing Date line to be accessed.

Returns:

returns the String for the wcDateLine to be inputed in profile's text file.

setAverageChoreTime

```
public void setAverageChoreTime(int totalChoreTime,  
                                int noOfChores)
```

Calculates and stores the value of the average chore time.

Parameters:

totalChoreTime - the combined total chore time value.
noOfChores - the number of chores value.

setAverageTimeOutput

```
public void setAverageTimeOutput(int averageChoreTime)
```

Calculates and stores the String value to be outputted for the averageChoreTime.

Parameters:

averageChoreTime - the average chore time value.

setNoOfChores

```
public void setNoOfChores(int i)
```

Stores the numbers of chores for a profile.

Parameters:

i - number of chores to be assigned to {@link noOfChores} field.

setTotalChoreTime

```
public void setTotalChoreTime(ChoresList cl)
```

Calculates and stores the value of the combined total chore time.

Parameters:

cl - the list of chores to be searched through to set a value for totalTime

setTotalChoreTimeOutput

```
public void setTotalChoreTimeOutput(int totalTime)
```

Calculates and stores the String value to be outputted for the totalTime.

Parameters:

totalChoreTime - the combined total chore time value.

toString

```
public java.lang.String toString()
```

Overrides the default toString() representation of this class.

Overrides:

toString in class java.lang.Object

model

Class ProfileData

```
java.lang.Object
|
+--model.ProfileData
```

All Implemented Interfaces:

java.lang.Iterable

< [Fields](#) > < [Constructors](#) > < [Methods](#) >

```
public class ProfileData
extends java.lang.Object
implements java.lang.Iterable
```

This class is responsible for creating a ProfileData object. Implements Iterable to allow elements to be iterated through.

Author:

Yashwant Rathor

Fields

choresAllocation

```
private ChoresAllocation choresAllocation
```

A ChoresAllocation object.

profiles

```
private java.util.ArrayList profiles
```

An object that contains an ArrayList of profiles.

Constructors

ProfileData

```
public ProfileData()
```

Default Constructor.

Methods

addProfile

```
public void addProfile(Profile p)
```

Allows the profiles list to be populated with a single profile.

Parameters:

p - Profile object to be added to the {@link profiles} field.

getChoresAllocation

```
public ChoresAllocation getChoresAllocation()
```

Allows the allocation of chores to be accessed.

Returns:

returns the choresAllocation object.

getProfile

```
public Profile getProfile(int index)
```

Allows a profile to be accessed using a index position.

Parameters:

index - The index of the Profile object to be accessed.

Returns:

returns the Profile object located at the index supplied in the parameter.

getProfiles

```
public java.util.ArrayList getProfiles()
```

Allows all profiles stored within the ArrayList to be accessed.

Returns:

returns all Profile objects stored within the ArrayList.

iterator

```
public java.util.Iterator iterator()
```

Allows elements within profiles to be iterated through.

removeDuplicates

```
public void removeDuplicates(java.util.ArrayList list)
```

Removes any duplicate Profile objects from the ArrayList.

Parameters:

list - filtered ArrayList of Profile objects to the {@link profiles} field..

setChoresAllocation

```
public void setChoresAllocation(ChoresAllocation c)
```

Populates the allocation of chores.

Parameters:

c - ChoresAllocation object to be assigned to the {@link choresAllocation} field.

toString

```
public java.lang.String toString()
```

Overrides the default toString() representation of this class.

Overrides:

toString in class java.lang.Object