

CS3024

Generated by Doxygen 1.8.6

Sun Nov 23 2014 14:01:03

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

DateMan	??
FileAccess	??
Food	??
GDA	??
Main	??
Meal	??
MealPlanner	??
Person	??

Chapter 2

Class Documentation

2.1 DateMan Class Reference

Public Member Functions

- String **getExpDate** (int day, int month, int year, int daysFromToday)
- String **getExpDate** (int daysFromToday)

2.1.1 Detailed Description

A representation of a date.

2.1.2 Member Function Documentation

2.1.2.1 String DateMan.getExpDate (int *day*, int *month*, int *year*, int *daysFromToday*)

Returns the date of expiration of the **Food** (p. ??).

Parameters

<i>day</i>	The day of the month.
<i>month</i>	The month of the year
<i>year</i>	The year.
<i>daysFromToday</i>	

Returns

The expiration date.

2.1.2.2 String DateMan.getExpDate (int *daysFromToday*)

Returns the date of expiration of the **Food** (p. ??).

Parameters

<i>daysFromToday</i>	The expiration date from today.
----------------------	---------------------------------

Returns

The expiration date.

The documentation for this class was generated from the following file:

- DateMan.java

2.2 FileAccess Class Reference

Public Member Functions

- String **getRecordById** (String id)
- String **getIdFromRec** (String rec)
- String **getNameFromRec** (String rec)
- int **getRoomD** (String rec)
- int **getFridgeD** (String rec)
- int **getFreezerD** (String rec)

Public Attributes

- int **headerL** = 0

2.2.1 Detailed Description

Author

fps

A class that will access the **Food** (p. ??) information.

2.2.2 Member Function Documentation

2.2.2.1 int FileAccess.getFreezerD (String rec)

Calculates the **Food** (p. ??) use by date when stored in the freezer.

Parameters

<i>rec</i>	
------------	--

Returns

The **Food** (p. ??) use by date in the freezer.

2.2.2.2 int FileAccess.getFridgeD (String rec)

Calculates the **Food** (p. ??) use by date when stored in the fridge.

Parameters

<i>rec</i>	
------------	--

Returns

The **Food** (p. ??) use by date in the fridge.

2.2.2.3 String FileAccess.getIdFromRec (String *rec*)

Get's the ID from the record.

Parameters

<i>rec</i>	The current record.
------------	---------------------

Returns

the ID.

2.2.2.4 String FileAccess.getNameFromRec (String *rec*)

Get's the name from the given record.

Parameters

<i>rec</i>	The record wanting to be returned.
------------	------------------------------------

Returns

The name of the record.

2.2.2.5 String FileAccess.getRecordById (String *id*)

Finds the record in the raf with a unique ID.

Parameters

<i>id</i>	
-----------	--

Returns

the Record found.

2.2.2.6 int FileAccess.getRoomD (String *rec*)

Calculates the **Food** (p. ??) use by date when stored at room temperature.

Parameters

<i>rec</i>	
------------	--

Returns

The **Food** (p. ??) use by date at room temperature.

2.2.3 Member Data Documentation

2.2.3.1 int FileAccess.headerL = 0

Using an .raf file storing **Food** (p. ??) this class will find a food and return information about it's use by date.

The documentation for this class was generated from the following file:

- FileAccess.java

2.3 Food Class Reference

Public Member Functions

- **Food** (String key, double unts, int amnt, double pric, double cal, double prot, double crbs, double sgrs, double ft, double sats, double fbr, double slt)
- String **toString** ()
- String **getName** ()

2.3.1 Detailed Description

Author

ruaraidh

A representation of a **Food** (p. ??).

2.3.2 Constructor & Destructor Documentation

2.3.2.1 Food.Food (String key, double unts, int amnt, double pric, double cal, double prot, double crbs, double sgrs, double ft, double sats, double fbr, double slt)

Constructs a food with all the necessary attributes.

Parameters

<i>key</i>	That access the food in the database.
<i>unts</i>	Units of food maybe kgs or litres. TODO: Adapt depending on food.
<i>amnt</i>	Amount of food, example if there is 4 cans of tomatoes then amount is 4.
<i>pric</i>	The price of the food.
<i>cal</i>	The calories in the food.
<i>prot</i>	The protein in the food.
<i>crbs</i>	The carbohydrates in the food.
<i>sgrs</i>	The sugars in the food.
<i>ft</i>	The fat in the food.
<i>sats</i>	The saturated fat in a food.
<i>fbr</i>	The fibre in the food.
<i>slt</i>	The salt in the food.

2.3.3 Member Function Documentation

2.3.3.1 String Food.getName ()

Get's the name of the food.

Returns

Name of the food.

2.3.3.2 String Food.toString ()

Represents the **Food** (p. ??) as a String.

Returns

The attributes of the food as a String.

The documentation for this class was generated from the following file:

- Food.java

2.4 GDA Class Reference

Public Member Functions

- **GDA** (**Person** p)
- String **toString** ()
- double **findTDEE** (**Person** p)
- double **toGrams** (double x)
- double **toKcal** (double x)

2.4.1 Detailed Description**Author**

ruaraidh

A representation of what nutrients a user needs each day.

2.4.2 Constructor & Destructor Documentation**2.4.2.1 GDA.GDA (Person p)**

Constructs the **GDA** (p. ??) model for a user. Calculates each as a ratio of calories and converts to grams.

Parameters

<i>p</i>	takes in a user.
----------	------------------

2.4.3 Member Function Documentation**2.4.3.1 double GDA.findTDEE (Person p)**

Calculates the user's Body Mass Ration (B.M.R). This is a calculation that works with attributes from **Person** (p. ??) class. After B.M.R is calculate then depending on the user's exercise level the calories consumed in a day is calculated.

Parameters

p	Takes in a person.
-----	--------------------

Returns

The amount of calories a **Person** (p. ??) consumes in a day.

2.4.3.2 double GDA.toGrams (double x)

Formats the ratios from calories to convert them to grams then round them to 2 decimal places.

Parameters

x	The ratio of calories that needs to be formatted.
-----	---

Returns

The formatted version.

2.4.3.3 double GDA.toKcal (double x)

Formats the calories to round to 2 decimal places.

Parameters

x	
-----	--

Returns

The rounded result.

2.4.3.4 String GDA.toString ()

Represents the **GDA** (p. ??) as a string.

The documentation for this class was generated from the following file:

- GDA.java

2.5 Main Class Reference

Static Public Member Functions

- static void **main** (String[] args)

2.5.1 Detailed Description**Author**

ruaraidh

Main (p. ??) class that brings all the other classes together.

2.5.2 Member Function Documentation

2.5.2.1 `static void Main.main (String[] args) [static]`

Will create a **Person** (p. ??) object, **Food** (p. ??) and the create a **MealPlanner** (p. ??) and add the **Food** (p. ??).
TODO: add a **MealPlanner** (p. ??) to the **Person** (p. ??) that will associate with.

Parameters

<i>args</i>	
-------------	--

The documentation for this class was generated from the following file:

- Main.java

2.6 Meal Class Reference

Public Member Functions

- **Meal** (String name, ArrayList< **Food** > ingredients)
- String **toString** ()
- void **add** (**Food** f)
- String **getName** ()

2.6.1 Detailed Description

Author

ruaraidh Represents a **Meal** (p. ??)

2.6.2 Constructor & Destructor Documentation

2.6.2.1 Meal.Meal (String name, ArrayList< Food > ingredients)

Constructs a default **Meal** (p. ??).

Parameters

<i>name</i>	The Meal (p. ??) name.
<i>ingredients</i>	The Food (p. ??) needed to make the Meal (p. ??).

2.6.3 Member Function Documentation

2.6.3.1 void Meal.add (Food f)

Allows a **Food** (p. ??) to be added as an ingredient.

Parameters

<i>f</i>	
----------	--

2.6.3.2 String Meal.getName ()

Gets the name of the **Meal** (p. ??).

Returns

The name of the **Meal** (p. ??).

2.6.3.3 String Meal.toString ()

Displays the **Meal** (p. ??) as a String

The documentation for this class was generated from the following file:

- Meal.java

2.7 MealPlanner Class Reference

Public Member Functions

- **MealPlanner** ()
- void **add** (**Meal** name, String day, String time)
- String **toString** ()
- int **writeToDay** (String day)
- int **writeToTime** (String time)
- String **getTime** (int i)
- String **getDay** (int i)

2.7.1 Detailed Description

Author

ruaraidh

Creates an empty 2D grid to allow a weekly meal plan to be created.

2.7.2 Constructor & Destructor Documentation

2.7.2.1 MealPlanner.MealPlanner ()

Constructs a default empty planner.

2.7.3 Member Function Documentation

2.7.3.1 void MealPlanner.add (Meal name, String day, String time)

Allows a **Meal** (p. ??) to be added.

Parameters

<i>name</i>	The Meal (p. ??) name.
<i>day</i>	The day of the week it will be added to.
<i>time</i>	The time of the day to add the meal to.

2.7.3.2 String MealPlanner.getDay (int i)

Since the grid will have it's indexes as integers there needs to be a function so that when the grid is printed as a string it can convert the integer back to a string.

Parameters

<i>i</i>	
----------	--

Returns

The day of the week the meal is located on the planner.

2.7.3.3 String MealPlanner.getTime (int *i*)

Since the grid will have it's indexes as integers there needs to be a function so that when the grid is printed as a string it can convert the integer back to a string.

Parameters

<i>i</i>	
----------	--

Returns

The time of day the meal is located on the planner.

2.7.3.4 String MealPlanner.toString ()

Represents the **Meal** (p. ??) plan as a string.

2.7.3.5 int MealPlanner.writeToDay (String *day*)

The day will be passed in as a String but in order to write it to the grid it needs to be an integer. This function will convert the string to the correct number.

Parameters

<i>day</i>	
------------	--

Returns

The index of the grid to add.

2.7.3.6 int MealPlanner.writeToTime (String *time*)

The time will be passed in as a String but in order to write it to the grid it needs to be an integer. This function will convert the string to the correct number.

Parameters

<i>time</i>	
-------------	--

Returns

The index of the grid to add.

The documentation for this class was generated from the following file:

- MealPlanner.java

2.8 Person Class Reference

Public Member Functions

- **Person** (String name, int age, double height, double weight, char gender, int exercise)
- String **toString** ()
- String **getName** ()
- int **getAge** ()
- double **getHeight** ()
- double **getWeight** ()
- char **getGender** ()
- int **getExercise** ()
- void **setName** (String name)
- void **setAge** (int age)
- void **setHeight** (double height)
- void **setWeight** (double weight)
- void **setGender** (char gender)
- void **setExercise** (int exercise)

2.8.1 Detailed Description

Author

ruaraidh Each person object represent attributes of a user.

2.8.2 Constructor & Destructor Documentation

2.8.2.1 Person.Person (String *name*, int *age*, double *height*, double *weight*, char *gender*, int *exercise*)

Creates a **Person** (p. ??) with the desired attributes.

Parameters

<i>name</i>	user's name.
<i>age</i>	user's age.
<i>height</i>	user's height in cm.
<i>weight</i>	user's weight in kg.
<i>gender</i>	user's gender M being male and F being female.
<i>exercise</i>	user's exercise per week: 0 = Desk job with little exercise 1 = 1-3hrs/week of light exercise 2 = 3-5hrs/week of moderate exercise 3 = 5-6hrs/week of strenuous exercise 4 = 7-21hrs/week of strenuous exercise/work

2.8.3 Member Function Documentation

2.8.3.1 int Person.getAge ()

Get the user's age.

Returns

User's age.

2.8.3.2 `int Person.getExercise ()`

Get the user's name.

Returns

User's exercise per week.

2.8.3.3 `char Person.getGender ()`

Get the user's gender.

Returns

User's gender.

2.8.3.4 `double Person.getHeight ()`

Get the user's height in cm.

Returns

User's height in cm.

2.8.3.5 `String Person.getName ()`

Get the user's name.

Returns

User's name.

2.8.3.6 `double Person.getWeight ()`

Get the user's weight in kg.

Returns

User's weight in kg.

2.8.3.7 `void Person.setAge (int age)`

Sets the user's age. If the user wishes to change it.

Parameters

<i>age</i>	The user's new age.
------------	---------------------

2.8.3.8 `void Person.setExercise (int exercise)`

Sets the user's exercise level. If the user wishes to change it.

Parameters

<i>exercise</i>	The user's new exercise level.
-----------------	--------------------------------

2.8.3.9 void Person.setGender (char *gender*)

Sets the user's gender. If the user wishes to change it.

Parameters

<i>gender</i>	The user's new gender.
---------------	------------------------

2.8.3.10 void Person.setHeight (double *height*)

Sets the user's height. If the user wishes to change it.

Parameters

<i>height</i>	The user's new height.
---------------	------------------------

2.8.3.11 void Person.setName (String *name*)

Sets the user's name. If the user wishes to change it.

Parameters

<i>name</i>	The user's new name.
-------------	----------------------

2.8.3.12 void Person.setWeight (double *weight*)

Sets the user's weight. If the user wishes to change it.

Parameters

<i>weight</i>	The user's new weight.
---------------	------------------------

2.8.3.13 String Person.toString ()

A toString function to display a person object.

The documentation for this class was generated from the following file:

- Person.java