## Class Summary

|  |  |  |
| --- | --- | --- |
| Enum, interface, class, abstract class | Name of the type | Description of use or purpose of this type |
| Class | Main | The “Driver” class of this program, it controls the initialization and termination of the program while providing a GUI to the users and uses the GUI as a way to build interaction between user and underlying data structures. |
| Class | FoodData | The database of all the food and other trees that store the nutrients references of all the food. |
| Interface | FoodDataADT | The interface of the food data. It specifies basic operations that FoodData class needs to implement such as filtering food items and loading files. |
| Class | FoodItem | Store all the information of one food item. |
| Class | BPTree | The data structure we use to store reference of food item that are in order by the nutrition amount. |
| Interface | BPTreeADT | The interface of the BPTree. It specifies basic operations that a BPTree should accomplish: insert and rangeSearch. |
| Class | GUI | GUI object for the program. It provides user specified options and display outcomes. |
| Class | MealList | Stores the list of food items added or to be added to the meal plan and includes summary methods. |

## Class Diagram

Design Document Requirements

Your design document must include:

* Class Summary - a table with a row for each class, interface, or enum
  + column 1 - indicates if the type is an interface, class, or enum
  + column 2 - contains the name of the type
  + column 3 - contains a brief description of the purpose or use of that type in your project  
      
    EXAMPLE:

|  |  |  |
| --- | --- | --- |
| ***enum, interface, class, abstract class*** | ***Name of the type*** | ***Description of use or purpose of this type*** |
| interface | DataStructureADT | defines required operations for my data structure |
| class | DataStructure implements DataStructureADT | implements DataStructureADT and ... |
| ... | ... | ... |

* Class Diagrams - include a table for each data type that provides the following
  + For each class in your project
    - Create a table (table) ***or*** [***UML Class diagram (Links to an external site.)Links to an external site.***](https://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/index.html)showing the public fields and methods of that type:
    - For each public field, describe its purpose and valid range of values
    - For each public constructor, describe its parameters
    - For each public method of the class, add a row with the following columns:
    - column 1 - the return type
    - column 2 - the method name
    - column 3 - parameter list
    - column 4 - brief description of the method
* Object Diagram - a sketch or list of the object instances that exist when program first is first launched
  + [**Unified Modeling Language Object Diagram example** (Links to an external site.)Links to an external site.](https://www.geeksforgeeks.org/unified-modeling-language-uml-object-diagrams/)
  + show what instances (objects) exist when the program's main GUI page has been created
  + show relationships between objects with edges and labels describing the connections.
  + you are not required to additional objects that will be created when program is being used  
    (just show those that exist at start)
  + label your figure and explain any abbreviations or symbols you use

### GUI Layout Sketch - an image saved in your design document

* a sketch or graphic image showing your proposed graphic user interface (GUI) at start of program.

A close up of text on a white background

Description automatically generated