# Zircon T**eam Project OOP**

## Documentation of “My Fridge WPF Application”

### Team “Zircon”

We are team “Zircon”. We started as five members team - username: anilak(Калина Лазарова), username: velialarm(Венелин Кехайов), username: andrew.slavchev (Andrew Slavchev), username: viktorDimitrov (Виктор Димитров), username: hinkah (Хинка Хинева), but actually only two of us took part of the creation of the application: anilak, and hinkah.

## The URL of SVN repository

Our SVN repository is: [**http**://team-zircon.googlecode.com/svn/trunk/ team-zircon-read-only](http://team-zircon.googlecode.com/svn/trunk/%20team-zircon-read-only%20)

### 3. Project Explanation

#### Application Purpose

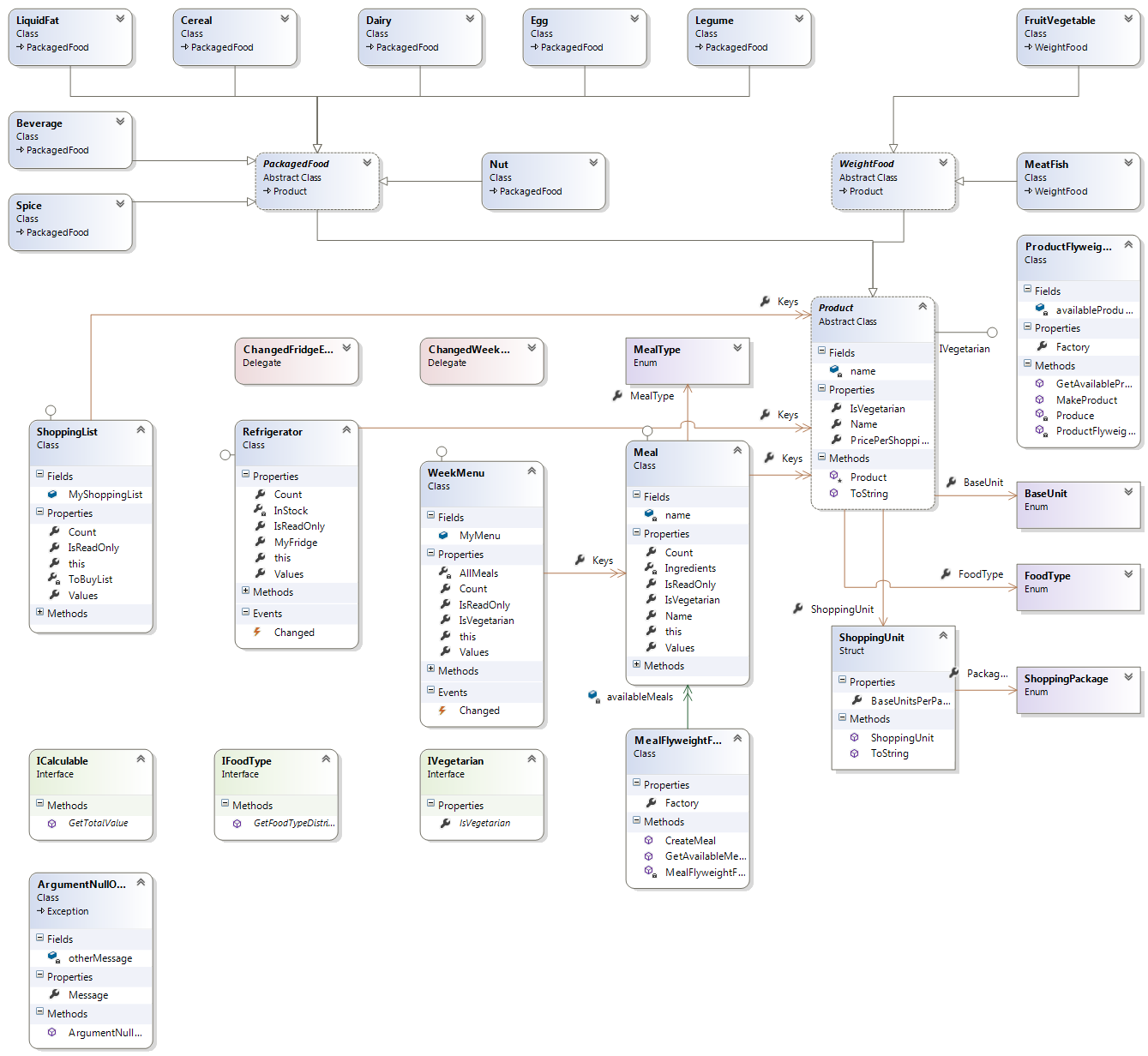
Our project is called “My Fridge”. The application main purpose is to assist you in composing your weekly shopping list.

The application offers an opportunity to compose a week menu by choosing from a list of possible meals. Each meal could be ordered more than once. It also is possible to switch between only vegetarian meals mode and all meals mode. When the week menu is composed the program gives information about the distribution of different food types in the composed menu (protein, fruit and vegetables, dairy and so on).

After composing the week menu then the user could optionally add or change the quantity of the products that are already available (referred to as change fridge quantity).

The final step is generating the week shopping list by the program. The shopping list consists of only these products which are not available in our fridge or their quantity is insufficient. It also offers for buying not the insufficient quantity but a whole shopping pack. The shopping list consists of products, needed quantity of shopping packs and their price in BGN. The shopping list has a total amount of BGN and could also be printed on paper.

#### Application Structure



The application logic is implemented by 20 classes. Three of them are abstract classes. At the top is abstract class **Product** which is inherited by two abstract classes – **WeightProducts** and **PackagedProducts**. Then these two classes are inherited by the concrete products classes: **FruitVegetable, MeatFish, Cereal, Dairy, Nut, LiquidFat, Spice, Legume, Egg and Beverage** **class**.

There are three interfaces that are defined and used in the program:

1. **ICalculable**, which allows us to price the meals and the products in the shopping list;
2. **IVegetarian**, which allows us to decide which product and meal is vegetarian;
3. **IFoodType**, which allows us to calculate the food type distribution in meals and in the week menu;

**Class Product** uses the **enum BaseUnit** for the base unit of the product, **enum FoodType** for food types and **ShoppingUnit struct**, which uses the **enum ShoppingPackage**. The prosess of creation of products is quite complicated and these products could be reused when once created which was the reason for creation of the **class ProductFlyweightFactory.** This class when asked for a product creates the products if it does not exist or retrieves it from a Dictionary which keeps the previously created products. This was implementation of Flyweight design pattern. The same model is used for the **class Meal** and **MealFlyweightFactory class.**

The Singleton design pattern is used for three of the application classes: **Refrigerator class**, **WeekMenu class** and **ShoppingList class**, because we have only one fridge, week menu and shopping list. They three are also Dictionaries which implement IDictionary interface. **Refrigerator class** keeps list of all products and quantities in the fridge.  **ShoppingList class** keeps all products and quantities which have to be bought and **WeekMenu class** keeps all the meals which are included in the week menu.

Two events are used in the application (in **class Refrigerator: Changed event** and in **class WeekMenu: Changed event**), which inform the subscribers for a change in either the quantities in the fridge or the week menu. The subscriber for these events is **ShoppingList class** which generates new shopping list each time one of these events is triggered.

The solution defines and uses one exception **class ArgumentNullOrWhiteSpace.**

For the user interface is used XAML and WPF. In the main window there is a menu at the top and three buttons for composing menu, generating shopping list and changing the content of the fridge. Each of these operations opens an additional window, in which the operation is performed. It is also possible to print to file or printer the generated shopping list from the “Generate Shopping List” window.