BIM492 Design Patterns

Term Project Details

The patterns we used at our project are:

1. Factory Method Pattern for the creation of the recipes
2. Command Pattern for modification of the recipes
3. Strategy Pattern to search for recipes
4. Observer Pattern to rate recipes

Factory Method Pattern

We have created a class file named *RecipeFactory* which includes a method named *createRecipe* to create recipes suddenly. Also, we have created a class file named *Recipe* to keep recipes' specifications, getter and setter methods, and constructors.

Command Pattern

We have created an interface file named *Command* which includes *execute* and *undo* methods. Also, we have created a *ModifyRecipeCommand* class file which implements the *Command* interface and overrides its methods, and includes variables like *recipe*, *oldName*, *newName*, *oldIngredients*, *newIngredients*, etc. Also, we have created a class file named *RecipeModificationInvoker* which includes *executeCommand* and *undoLastCommand* methods, constructor, and *commandHistory* variable.

Strategy Pattern

We have created an interface file named *SearchStrategy* which includes a method named *search*. Also, created three class files and these are *SearchByCategoriesStrategy*, *SearchByIngredientsStrategy*, *SearchByTagsStrategy*. These three files implement the *SearchStrategy* interface and override its *search* method. Lastly, we have created a class file named *RecipeSearcher* to set strategy via the *setSearchStrategy* method and to search recipes via the *searchRecipes* method. This file also includes a variable named *searchStrategy* to keep the search strategy.

Observer Pattern

Firstly, we have created an interface file named *RecipeRatingObserver* which includes update method to update a recipe's rating. Then, we have created two class files named *RecipeRatingDisplay* and *RecipeRating*. *RecipeRatingDisplay* implements the *RecipeRatingObserver* interface and overrides its update method and includes a variable named *recipeRating* and a constructor. The other class *RecipeRating* includes two variables named observers and rating. Also, this class includes four methods named *addObserver*, *removeObserver*, *setRating*, and *notifyObservers*. *setRating* method takes recipe and rating which user entered and calculates new average rating for the recipe.

Metehan Kürşat Güzel – 53164674006

Doğukan Turgut – 27445149668

İsmail Burak Arıkan – 61864071384