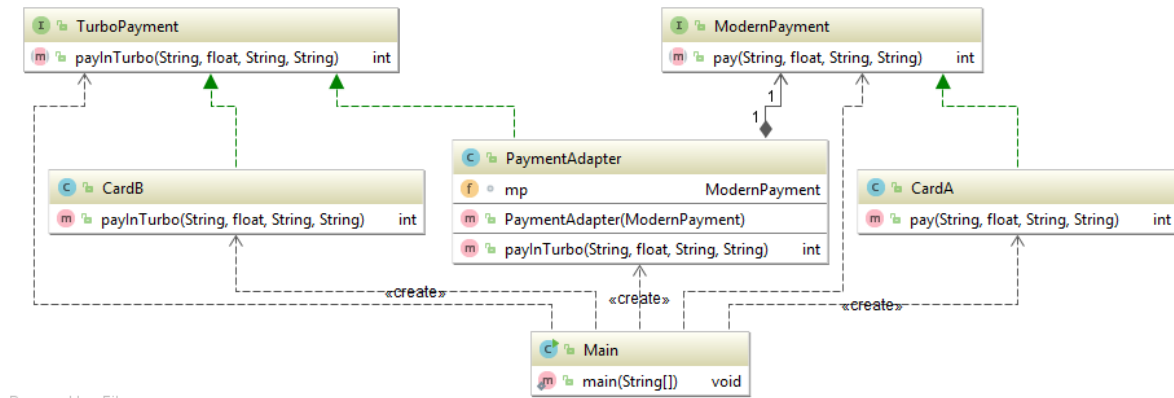


CSE443
OBJECT ORIENTED ANALYSIS AND
DESIGN
HW3
DOCUMENT

HAKKI ERDEM DUMAN
151044005

QUESTION 1

In this question, adapter pattern is used to adapt Turbo Payment, which is the old payment method, to the Modern Payment, which is the newer one.

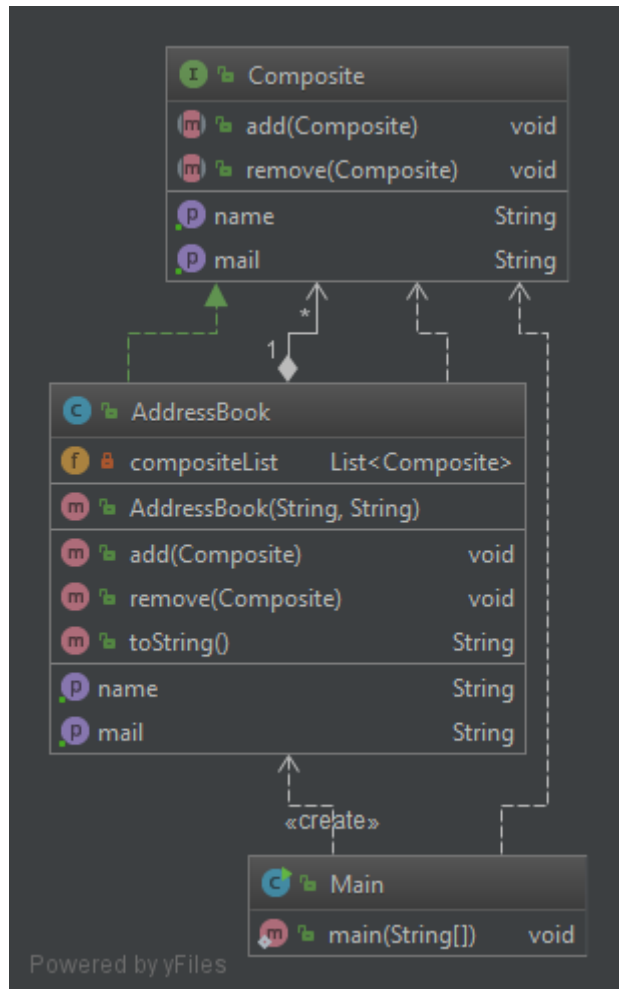


CardA class implements ModernPayment interface and CardB class implements TurboPayment interface. Since TurboPayment interface and CardB class can't be changed, PaymentAdapter class implements the interface that will be adapted to newer system, which is "TurboPayment" one. PaymentAdapter gets a ModernPayment object and calls its payment method inside of "payInTurbo" method. That's the way we adapt payInTurbo to new system.

In program output, first set is printed by ModernPayment way. Second one is printed by TurboPayment way and third set is printed after the old system is adapted to new one.

QUESTION 2

In this question, composite pattern is used to keep a list of a class inside of the same class.



AddressBook class implements Composite interface and keeps a list of the same interface inside of itself. When “add” method of this class is called, the user, who is given by parameter is added to that list. “Remove” method deletes the parameter from list. And “toString” of this method prints mail address and name of the user. If the list of object is filled, also prints their information.

In “main” method, every object is created as a personal account. Whenever another user is added to a personal account, that personal account becomes a

group account. That’s the way I interpret this part of homework.

Harry Potter characters are used for example. Gryffindor was a personal account until some users are added into it.

QUESTION 3

In this question, iterator is used to iterate a given array.

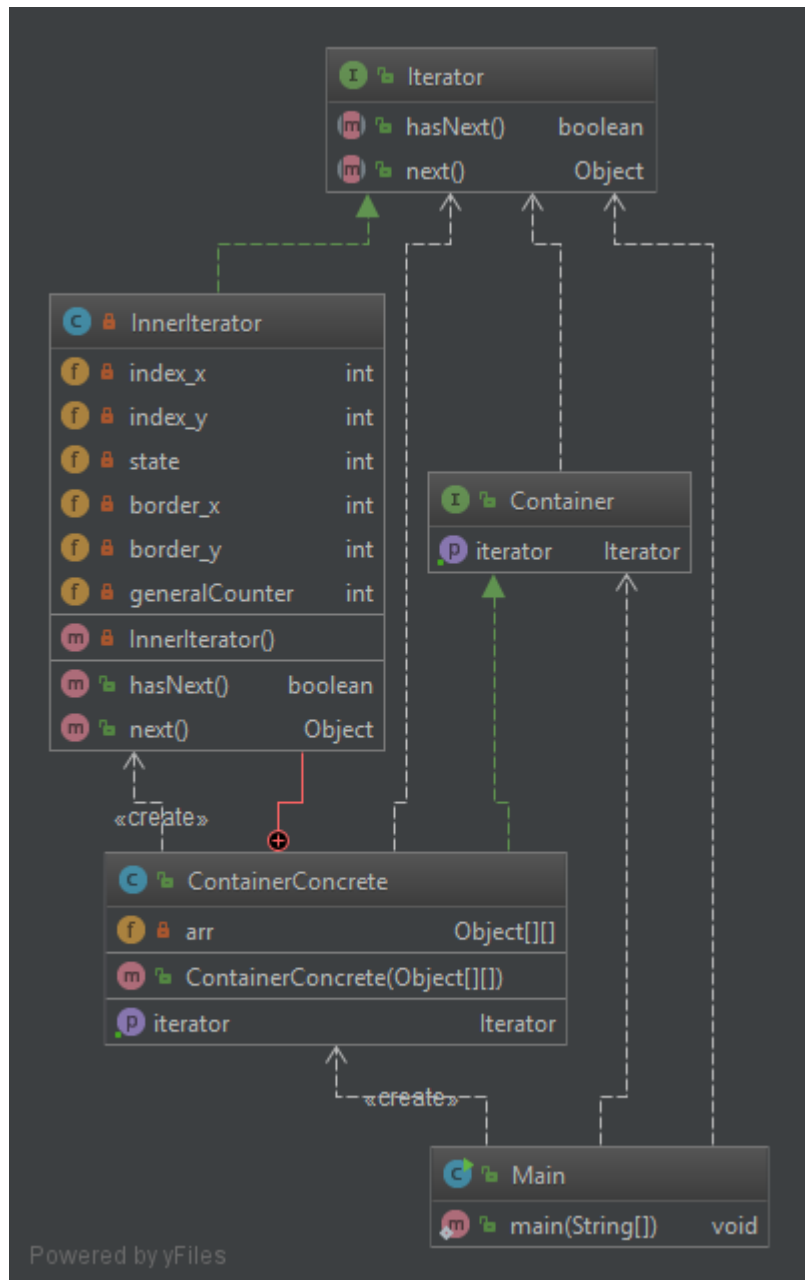
Container interface is created to be implemented by ContainerConcrete class. This class has a method called “getIterator” that returns an iterator, and has an inner class, which is iterator’s itself.

This iterator has 2 methods called “hasNext” and “next”. hasNext method checks whether there is an element in array to be iterated. If there is no element, it returns false and that provides the program an end. “next” method returns the

following element of given array, according to its iterating rule. In this program, elements are iterated clockwise.

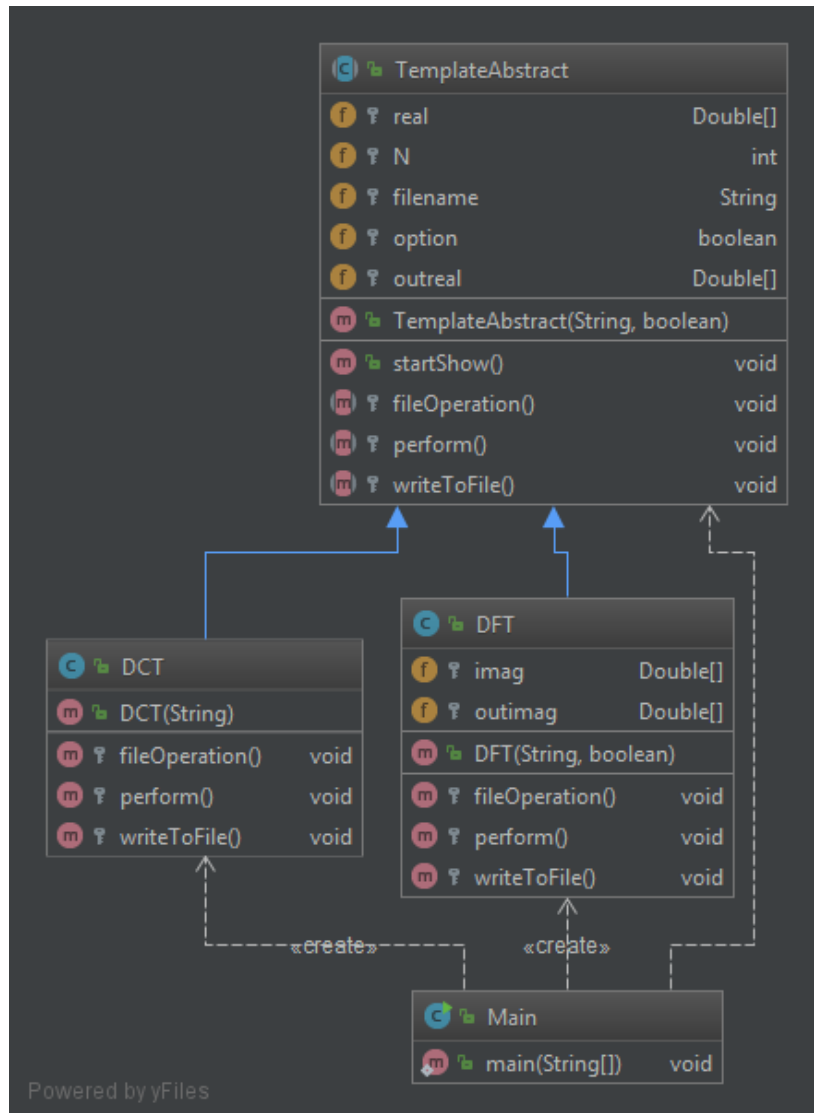
Array can be changed by using “array.txt”. It doesn’t have to be a square matrix, but it should be a rectangle at least.

Elements of the array are separated by commas.



QUESTION 4

In this question, Template Method design pattern is used to choose the method we use.



TemplateAbstract class is an abstract class. Its methods are not implemented except one. “startShow” method is actually the only method that is implemented by itself to call the methods, which are required to make calculation, one by one. If an object is created by DFT class, DFT classes’ methods will be used. Same thing is in use for DCT class. The trick part is constructor of these two implementor classes. DFT has 2

arguments but DCT has one because DFT has a chance to display process time. Since there is an option for DFT, DCT’s constructor, that matches super class, always passes its second parameter as “false”.

In “values.txt”, values before commas are “real” numbers and values after commas are “imaginary” numbers.

There is no imaginary part for inputs of DCT. DCT only uses the values before commas from “values.txt” because of that.

Solution of DCT or DFT is written inside “out.txt”.