CSE443 – Object Oriented Analysis and Design – Midterm

Gökçe Nur Erer – 171044079

Question 1:

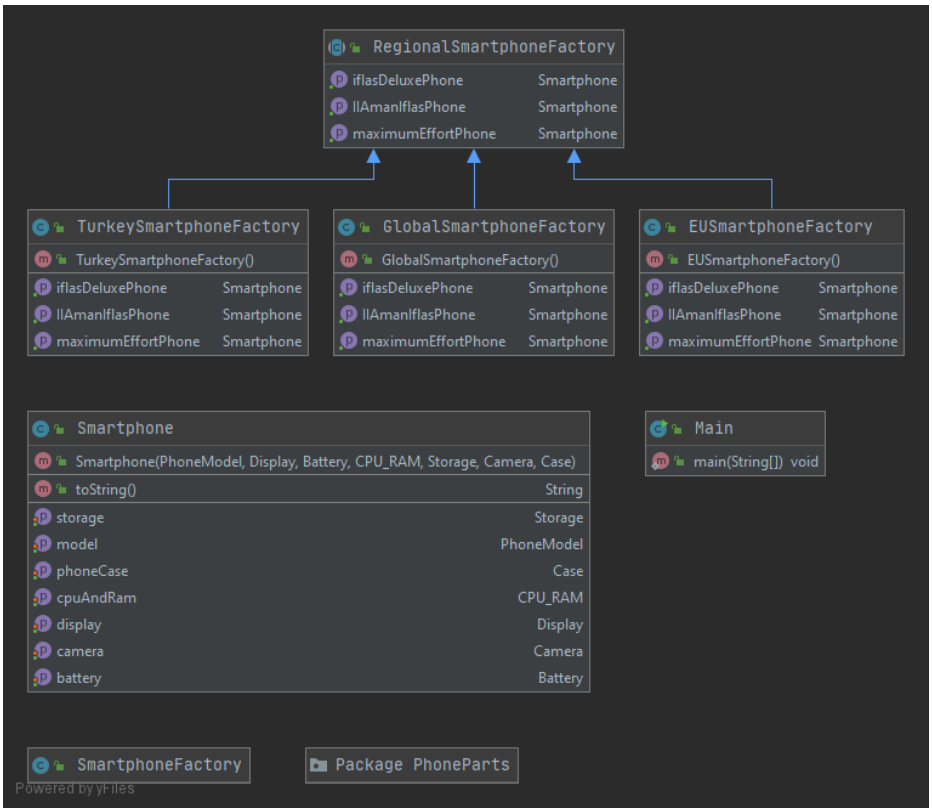
For this question Abstract Factory pattern is used as it was said in the problem description.

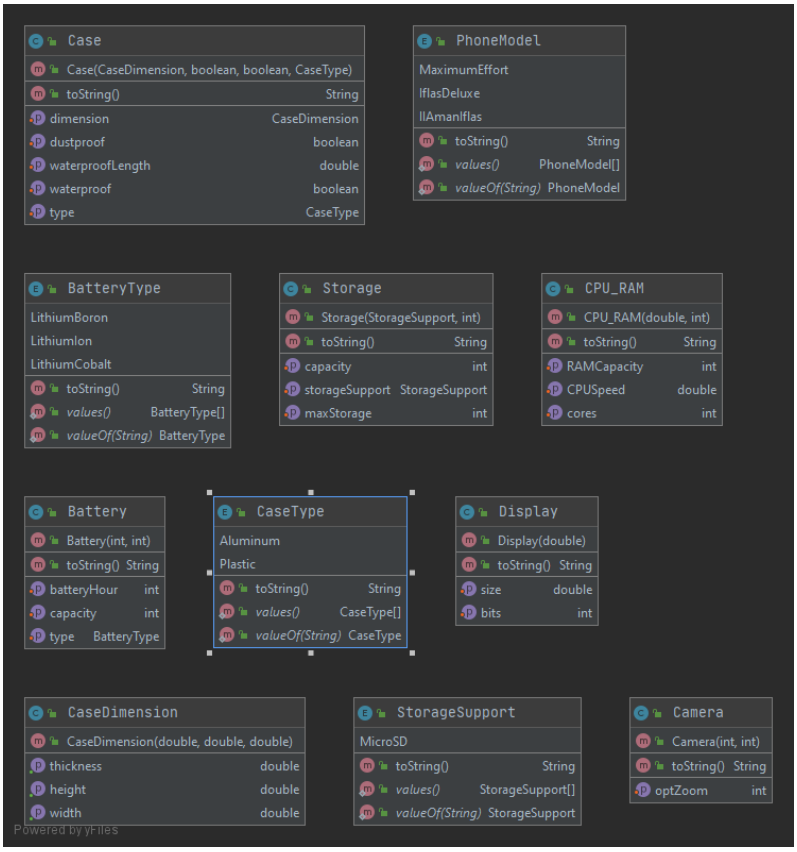
To implement this design pattern the following classes and packages are created:

- **Smartphone:** This class represents a smartphone which has a battery, display, cpu, ram, storage unit, camera and a case.
- **SmartphoneFactory:** This class represents a smartphone factory which creates smartphones with their base requirements.
- **RegionalSmartphoneFactory:** This class is an abstract class that represents a regional smartphone factory which can add region specific specifications to the smartphones. It keeps a SmartphoneFactory instance so that it can be used to create the phone base.
- **TurkeySmartphoneFactory:** This class represents Turkey factory of smartphones. It implements the abstract methods in the RegionalSmartphoneFactory according to Turkey's specifications.
- **EUSmartphoneFactory:** This class represents EU factory of smartphones. It implements the abstract methods in the RegionalSmartphoneFactory according to EU's specifications.
- **GlobalSmartphoneFactory:** This class represents global factory of smartphones. It implements the abstract methods in the RegionalSmartphoneFactory according to global specifications.

- **PhoneParts:**
    - Battery : Represents the battery in a smartphone.
    - BatteryType : Represents the type of a battery.
    - Camera : Represents the camera in a smartphone.
    - Case: Represents the case in a smartphone.
    - CaseDimension: Represents dimensions (width, height and thickness) of a case.
    - CaseType: Represents the type of a case.
    - CPU_RAM: Represents the CPU&RAM in a smartphone.
    - Display: Represents the CPU&RAM in the smartphone.
    - PhoneModel: Represents the model of the smartphone.
    - Storage: Represents the storage unit of the smartphone.
    - StorageSupport: Represents the support type of the storage unit.

Class Diagrams





Outputs

For Turkey:

```
---------- IFLAS TECHNOLOGIES TURKEY ----------

Creating "Maximum Effort Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = Maximum Effort
Display: Size = 5.5 inches , Bit Size = 32 bits
Battery: Battery Hour = 27h , Battery Capacity = 3600mAh , Battery Type = Lithium-Boron
CPU & RAM: CPU Speed = 2.8GHz , RAMCapacity = 8GB , Cores = 8 cores
Storage: Storage Support = MicroSD support , Capacity = 64GB , Max Storage = 128GB
Camera: Front Resolution = 12Mp , Rear Resolution = 8Mp , Opt Zoom = 4x
Case: Dimension = 151.0x73.0x7.7mm , Dustproof = true , Waterproof = true , Waterproof Length = 2.0m , Type = Aluminum
```

```
Creating "Iflas Deluxe Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = Iflas Deluxe
Display: Size = 5.3 inches , Bit Size = 32 bits
Battery: Battery Hour = 20h , Battery Capacity = 2800mAh , Battery Type = Lithium-Boron
CPU & RAM: CPU Speed = 2.2GHz , RAMCapacity = 6GB , Cores = 8 cores
Storage: Storage Support = MicroSD support , Capacity = 32GB , Max Storage = 128GB
Camera: Front Resolution = 12Mp , Rear Resolution = 5Mp , Opt Zoom = 4x
Case: Dimension = 149.0x73.0x7.7mm , Dustproof = false , Waterproof = true , Waterproof Length = 2.0m , Type = Aluminum
```

```
Creating "I-I-Aman Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = I-I-Aman-Iflas
Display: Size = 4.5 inches , Bit Size = 32 bits
Battery: Battery Hour = 16h , Battery Capacity = 2000mAh , Battery Type = Lithium-Boron
CPU & RAM: CPU Speed = 2.2GHz , RAMCapacity = 4GB , Cores = 8 cores
Storage: Storage Support = MicroSD support , Capacity = 16GB , Max Storage = 128GB
Camera: Front Resolution = 8Mp , Rear Resolution = 5Mp , Opt Zoom = 4x
Case: Dimension = 143.0x69.0x7.3mm , Dustproof = false , Waterproof = true , Waterproof Length = 2.0m , Type = Plastic
```

For EU:

```
---------- IFLAS TECHNOLOGIES EU ----------

Creating "Maximum Effort Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = Maximum Effort
Display: Size = 5.5 inches , Bit Size = 24 bits
Battery: Battery Hour = 27h , Battery Capacity = 3600mAh , Battery Type = Lithium-Ion
CPU & RAM: CPU Speed = 2.8GHz , RAMCapacity = 8GB , Cores = 4 cores
Storage: Storage Support = MicroSD support , Capacity = 64GB , Max Storage = 64GB
Camera: Front Resolution = 12Mp , Rear Resolution = 8Mp , Opt Zoom = 3x
Case: Dimension = 151.0x73.0x7.7mm , Dustproof = true , Waterproof = true , Waterproof Length = 1.0m , Type = Aluminum
```

```
Creating "Iflas Deluxe Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = Iflas Deluxe
Display: Size = 5.3 inches , Bit Size = 24 bits
Battery: Battery Hour = 20h , Battery Capacity = 2800mAh , Battery Type = Lithium-Ion
CPU & RAM: CPU Speed = 2.2GHz , RAMCapacity = 6GB , Cores = 4 cores
Storage: Storage Support = MicroSD support , Capacity = 32GB , Max Storage = 64GB
Camera: Front Resolution = 12Mp , Rear Resolution = 5Mp , Opt Zoom = 3x
Case: Dimension = 149.0x73.0x7.7mm , Dustproof = false , Waterproof = true , Waterproof Length = 1.0m , Type = Aluminum
```

```
Creating "I-I-Aman Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = I-I-Aman-Iflas
Display: Size = 4.5 inches , Bit Size = 24 bits
Battery: Battery Hour = 16h , Battery Capacity = 2000mAh , Battery Type = Lithium-Ion
CPU & RAM: CPU Speed = 2.2GHz , RAMCapacity = 4GB , Cores = 4 cores
Storage: Storage Support = MicroSD support , Capacity = 16GB , Max Storage = 64GB
Camera: Front Resolution = 8Mp , Rear Resolution = 5Mp , Opt Zoom = 3x
Case: Dimension = 143.0x69.0x7.3mm , Dustproof = false , Waterproof = true , Waterproof Length = 1.0m , Type = Plastic
```

For Global:

```
---------- IFLAS TECHNOLOGIES GLOBAL ----------

Creating "Maximum Effort Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = Maximum Effort
Display: Size = 5.5 inches , Bit Size = 24 bits
Battery: Battery Hour = 27h , Battery Capacity = 3600mAh , Battery Type = Lithium-Cobalt
CPU & RAM: CPU Speed = 2.8GHz , RAMCapacity = 8GB , Cores = 2 cores
Storage: Storage Support = MicroSD support , Capacity = 64GB , Max Storage = 32GB
Camera: Front Resolution = 12Mp , Rear Resolution = 8Mp , Opt Zoom = 2x
Case: Dimension = 151.0x73.0x7.7mm , Dustproof = true , Waterproof = true , Waterproof Length = 0.5m , Type = Aluminum
```

```
Creating "Iflas Deluxe Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = Iflas Deluxe
Display: Size = 5.3 inches , Bit Size = 24 bits
Battery: Battery Hour = 20h , Battery Capacity = 2800mAh , Battery Type = Lithium-Cobalt
CPU & RAM: CPU Speed = 2.2GHz , RAMCapacity = 6GB , Cores = 2 cores
Storage: Storage Support = MicroSD support , Capacity = 32GB , Max Storage = 32GB
Camera: Front Resolution = 12Mp , Rear Resolution = 5Mp , Opt Zoom = 2x
Case: Dimension = 149.0x73.0x7.7mm , Dustproof = false , Waterproof = true , Waterproof Length = 0.5m , Type = Aluminum
```

```
Creating "I-I-Aman Model Phone"...

Attaching display...
Attaching battery...
Attaching CPU & RAM...
Attaching storage...
Attaching camera...
Enclosing phone case...
Model = I-I-Aman-Iflas
Display: Size = 4.5 inches , Bit Size = 24 bits
Battery: Battery Hour = 16h , Battery Capacity = 2000mAh , Battery Type = Lithium-Cobalt
CPU & RAM: CPU Speed = 2.2GHz , RAMCapacity = 4GB , Cores = 2 cores
Storage: Storage Support = MicroSD support , Capacity = 16GB , Max Storage = 32GB
Camera: Front Resolution = 8Mp , Rear Resolution = 5Mp , Opt Zoom = 2x
Case: Dimension = 143.0x69.0x7.3mm , Dustproof = false , Waterproof = true , Waterproof Length = 0.5m , Type = Plastic
```

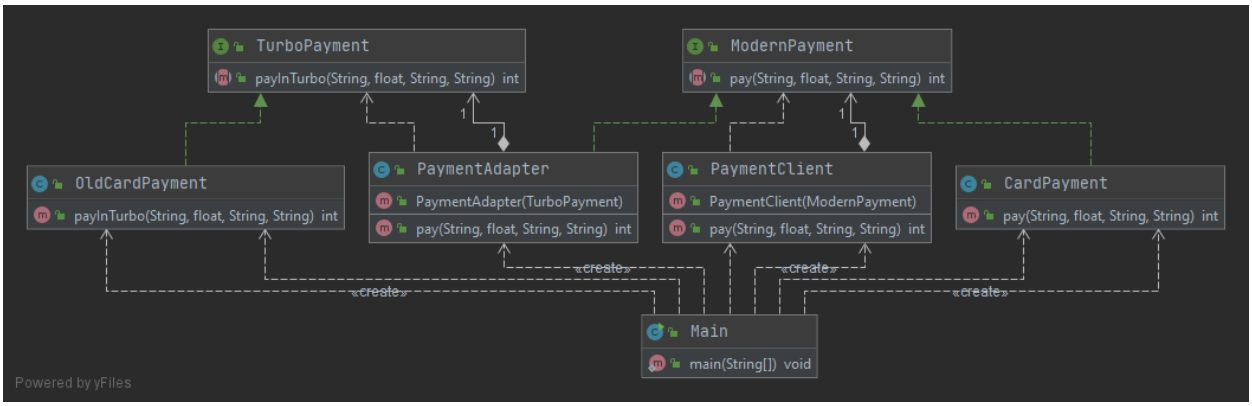
Question 2:

For this question, the Adapter pattern is used. Because Adapter Pattern enables an old implementation to adapt to a newer implementation without too much work.

To implement this pattern this classes and interfaces are created:

- ModernPayment : This interface represents the newer payment method.
- TurboPayment: This interface represents the older payment method.
- PaymentAdapter: This class is the class that adapts the TurboPayment's payment method to ModernPayment's payment method. It holds a TurboPayment instance to call its payment method.
- PaymentClient: This class represents the client who will use the pay method.
- OldCardPayment: This class is an implementation of the TurboPayment interface.
- CardPayment: This class is an implementation of the ModernPayment interface.

Class Diagram



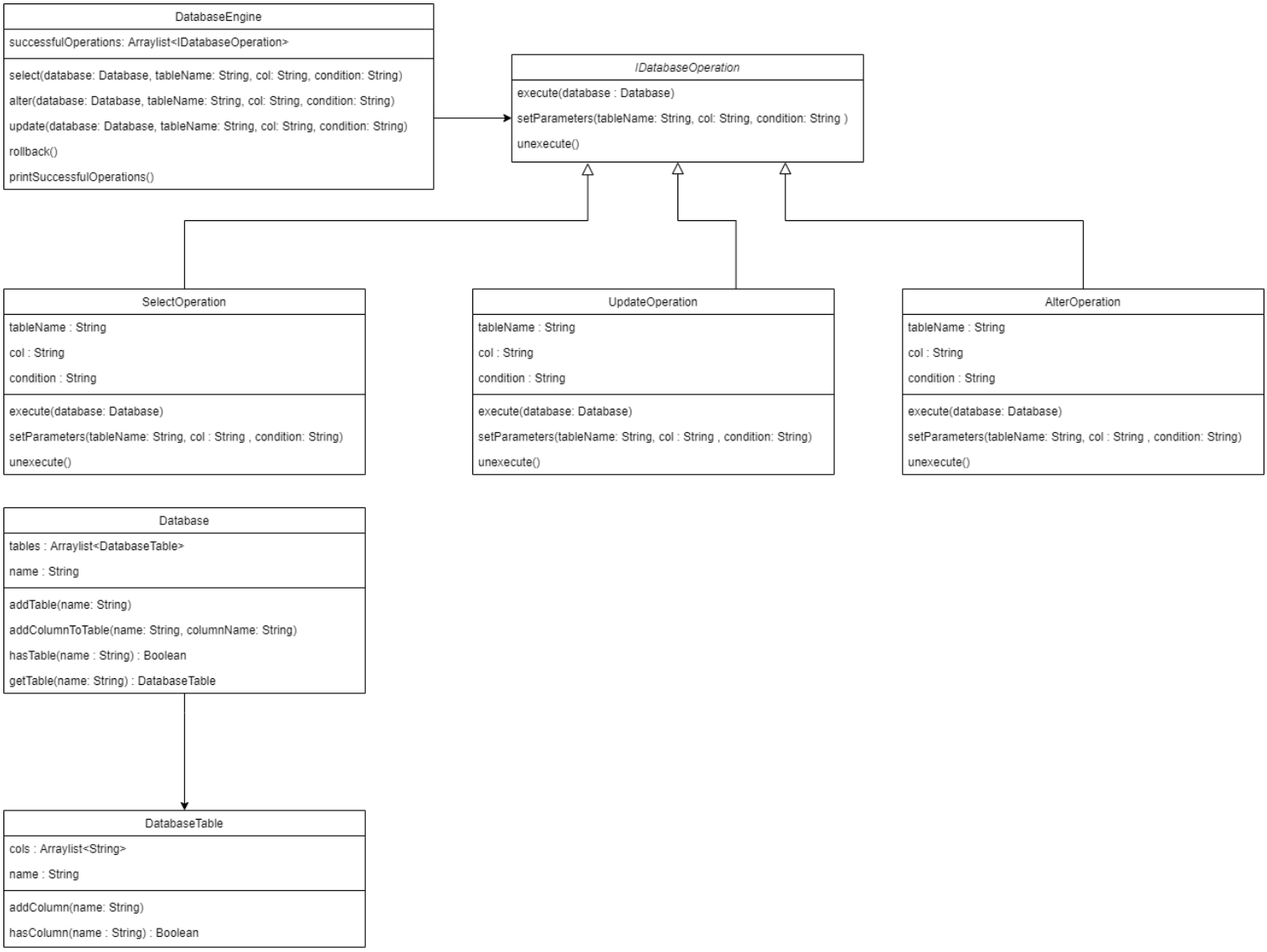
Outputs

```
---Client using the implementation of Turbo Payment for payment, adapter is used for compatibility---
Payment by using the method in Turbo Payment interface...
---Client using the implementation of Modern Payment for payment---
Payment by using the method in Modern Payment interface...
```

Question 3:

a) For this question, Command Pattern is used. Because command pattern defines certain commans and enables them to be executed and unexecuted.

To implement this pattern this class diagram is used:



b) For this question the classes above are implemented. For the rollback of the transactions, first a failing condition is decided which is table not existing in the given database or a column not existing in the given table. (This is not a complete implementation of a database, for checking a column or a table only the names are checked.) Then to rollback all the successful operations, DatabaseEngine class holds a IOperation list which holds the successful operations. When a operation fails by traversing the successful operation list from the end to the start, the operations are unexecuted by using their unexecute method one by one. After every successful operation is unexecuted the successful operations list is emptied.

Question 4:

In this question, Template Method pattern is used as it is defined in the problem description.

To implement this pattern the following classes are created:

- **OneDimensionalTransform:** This abstract class represents a total transform with reading from file, calculating the transform, writing to file. This class has an abstract method applyMethod which gets a string value array as an input. Also this class has a method called executeTransform which is the complete transform with file I/O and transformation.
- **DiscreteCosineTransform:** This class implements the abstract method applyMethod in OneDimensionalTransform to perform Discrete Cosine Transform on the given values.
  PS: When applying DCT to the values the first element in the result is multiplied by $1/\sqrt{2}$ and every element is multiplied by $\sqrt{2}/N$ to make the matrix orthogonal as it was mentioned in the definition in the Wikipedia.
- **DiscreteFourierTransform:** This class implements the abstract method applyMethod in OneDimensionalTransform to perform Discrete Fourier Transform on given values.
  PS: When evaluating the values in the input array. The assumption is that if the input is [1,2,3, …] the actual complex numbers are 1+i, 2+2i, 3+3i and so on.
  Also the result values are represented as 1+i or 2-2i notation in the output file.

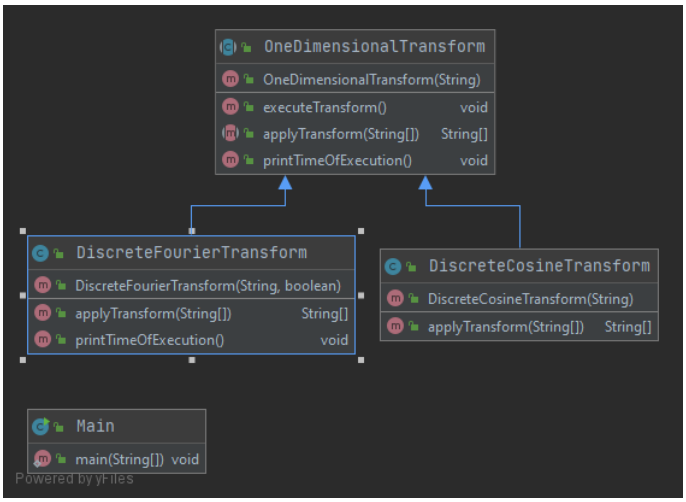
PS: The program for the solution of this question gets a filename, solution method and a parameter that decides if the DFT will show the execution time or not <u>in command line arguments</u>. In short the usage is:

<program name> <filename> <transform-method> <if:transformMethod == DFT shouldPrintExecutionTime>

Some examples can be:

- 171044079Q4 inputFile.txt DFT shouldPrintExecutionTime
- 171044079Q4 inputFile.txt DCT
- 171044079Q4 inputFile.txt DFT

Class Diagram



Outputs

With command line arguments : inputFile.txt and DCT

Output:

```
NO
```

outputFile.txt: (numbers are separated with tabs)

```
 46.96  -25.61  -0.00   -2.82   -0.00   -1.00   0.00    -0.49   -0.00   -0.29   -0.00   -0.18
-0.00  -0.11  -0.00  -0.07  -0.00  -0.04  -0.00  -0.01
```

With command line arguments : inputFile.txt DFT shouldPrintExecutionTime

Output:

```
16580500 nanoseconds
```

outputFile.txt: (numbers are separated with tabs)

```
210.00+0.00i    -10.00+63.14i  -10.00+30.78i  -10.00+19.63i  -10.00+13.76i  -10.00+10.00i
-10.00+7.27i   -10.00+5.10i   -10.00+3.25i   -10.00+1.58i   -10.00-0.00i   -10.00-1.58i
-10.00-3.25i   -10.00-5.10i   -10.00-7.27i   -10.00-10.00i  -10.00-13.76i  -10.00-19.63i
-10.00-30.78i  -10.00-63.14i
```

With command line arguments: inputFile.txt DFT

```
NO
```

outputFile.txt: (numbers are separated with tabs)

```
210.00+0.00i    -10.00+63.14i  -10.00+30.78i  -10.00+19.63i  -10.00+13.76i  -10.00+10.00i
-10.00+7.27i   -10.00+5.10i   -10.00+3.25i   -10.00+1.58i   -10.00-0.00i   -10.00-1.58i
-10.00-3.25i   -10.00-5.10i   -10.00-7.27i   -10.00-10.00i  -10.00-13.76i  -10.00-19.63i
-10.00-30.78i  -10.00-63.14i
```