Group Expense Tracker System

Creational Design Pattern - Singleton

Pentru buna funcționare a clasei “Currency” avem nevoie doar de o singura instanță a acestei clase. De asemenea, vom aveam nevoie de Singleton și la conexiunea bazei de date.

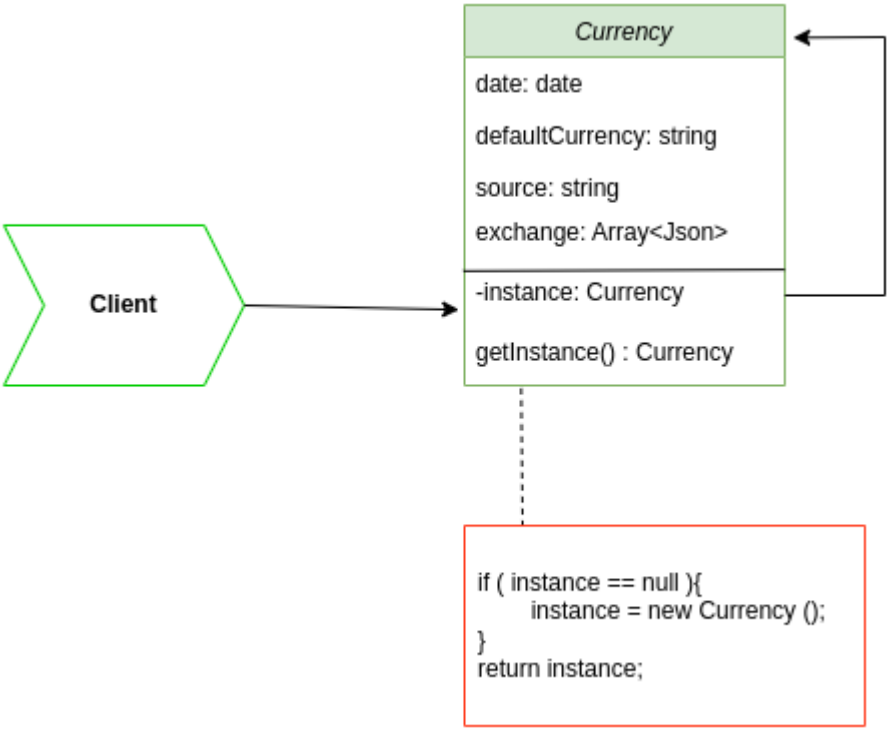
Problema:

Oriunde vom utiliza clasa “Currency” în aplicație, vom aveam nevoie doar de o singura instanță. Aceasta clasa oferă posibilitatea de a genera și a prelua cursul valutar dintr-o zi specifica. Astfel, request-ul către un anumit API se va realiza prin funcțiile clasei “Currency”. Aceeași problema o intampinam și la utilizarea clasei “Database” pentru conexiunea la baza de date.

Soluție:

Utilizăm Design Pattern Singleton pentru a avea o singura instanta generala, publica in toata aplicatia. (valabil pentru ambele clase)

Diagrama (similar și pentru “Database”):



Pseudocod clasa “Database”:

```
class Database is

    private static field instance: Database

    private constructor Database() is
        // Some initialization code, such as the actual
        // connection to a database server.
        // ...

    // The static method that controls access to the singleton
    // instance.
    public static method getInstance() is
        if (Database.instance == null) then

            Database.instance = new Database()

        return Database.instance


    public method query(sql) is
        //login for queries

class Application is
    method main() is
        Database foo = Database.getInstance()
        foo.query("SELECT ...")
        // ...
        Database bar = Database.getInstance()
        bar.query("SELECT ...")
        // The variable `bar` will contain the same object as
        // the variable `foo`.
```

Pseudocod clasa “Currency”:

```
class Currency is

    private static field instance : Currency

    Date date;

    String defaultCurrency;

    String source;

    Array <Json> exchange;


    private constructor Currency() is

        // constructor

    public static method getInstance() is

        if (Currency.instance == null)

            Currency.instance = new Currency()

        return Currency.instance

    public convert ( price, currencyType )

        return price * getExchangeRate( currencyType );

    public getExchangeRate ( currencyType = null )  // optional currency specify

        if ( currencyType)
            return callApiCurrencyRate ('https://www.fastforex.io/'. currencyType)
        else

            return this.exchange = callApiCurrencyRate ('https://www.fastforex.io/')
```

Diagrama Structurala Clase de Obiecte