Report

- Main goal:

Improve and automatize the current electricity billing system. Get rid of manual reading and billing process

- Design patterns used:

1) **FACTORY METHOD PATTERN** - Users of our system fall in 3 main categories: residential, commercial and industrial consumers, therefore there are different types of accounts for each of them. Factory method is used to create a particular type of consumer.
2) **COMPOSITE METHOD PATTERN** - Our system includes monthly and yearly payment methods and the latter is used when there is a debt for the user. So a year consists of months and it is a group of objects that is treated the same way as a single instance of the months.
3) **ITERATOR METHOD PATTERN** - Such a system always includes lists. In our case it is bills and consumer lists. Each of them requires different types of iteration so this method comes in handy as it allows us to take out iterators and add new ways to iterate through lists.

- UML diagram description:

Diagram includes all 3 used patterns. For **factory method** there is *ConsumerFactory* class which is the father for *ResidentialFactory, CommercialFactory* and *IndustrialFactory* classes.

MonthlyBill and *YearlyBill* classes relate to *Bill* as a child, and *YearlyBill* uses *BillCollection* to store bills.

UML also shows 2 iterator methods which are *BillListIterator* and *ConsumerListIterator.* Both of them are implementing *iterator* interface

Aside from pattern usage, there are classes that correspond directly to usage of application and user interaction with system classes. Everything has its own methods and data types

- Source code description:

Majority of classes correspond to their particular design pattern and are created to support functionality of our app. Except for them, there several other classes which are meaningful for our project

Our *Application* class is a simple tool to run our system and start to use it. It includes main methods for interaction between user and system.

Consumer class is fundamental for every new user which contains common information between all types of users.

Bill class creates a second indistinguishable part of our system. This is simple class containing information about consumed energy

Project was created by: Bobiev Nodir, Davlatkhodja Magzumov, Aibek Bakirov, Valikhan Ilyasov and Kaiyrly Alimzhan