

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 4**

з дисципліни “ Об'єктно-орієнтовне програмування ”

тема “Реалізація поведінкових шаблонів проектування”

|  |  |  |
| --- | --- | --- |
| Виконав  студентка II курсу  групи КП-72  Городченко Анна Володимирівна  (*прізвище, ім’я, по батькові*)  варіант № 6 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна  (*прізвище, ім’я, по батькові*) |

Київ 2019

**Мета роботи**

Ознаойомлення з основними характеристиками шаблонів «Стратегія», «Шаблонний метод» та «Стан», запам’ятовування поширених ситуацій, коли використання цих шаблонів є доцільним, набуття вмінь та навичок реалізації шаблонів під час створення програмного коду.

**Постановка завдання**

1. За допомогою шаблона проектування реалізувати підкажчик щодо приймання ліків хворими за чітким графіком: в залежності від дня тижня хворий повинен приймати різні ліки, наприклад:

а) в понеділок зранку - ліки No1, ввечері – ліки No4;

б) у вівторок зранку – ліки No2, ввечері - No4;

в) у середу зранку - ліки No3, ввечері - No4;

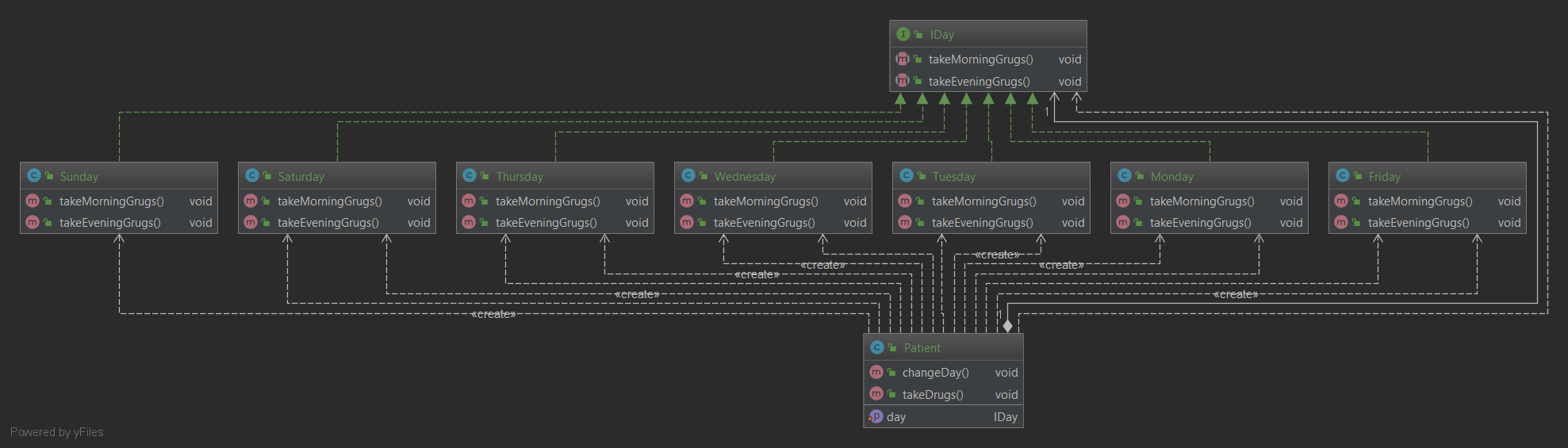
г) у четвер зранку – ліки No1, ввечері - No4;

д) в п’ятницю зранку – ліки No2, ввечері - No4;

е) в суботу зранку – ліки No3, ввечері - No4;

ж) у неділю перерва у прийомі ліків.

2. За допомогою шаблону проектування реалізувати різні алгоритми подання документів для отримання візи такими категоріями людей: службовці, приватні підприємці, діти. Як відомо, для кожної групи потрібні різні документи. Наприклад, ПП має подавати копію звіту про прибуток за останні 2 квартали з податкової інспекції; діти – дитячий проїзний документ тощо.Завдання 1

 Для виконання першого завдання було вирішено обрати шаблон проектування під назвою “State”. Він був використаний тому, що об’єкт може змінювати свій стан.

**Текст коду**

|  |
| --- |
| package State |
| public interface IDay {  void takeMorningGrugs();  void takeEveningGrugs(); }  public class Patient {  private IDay day;   public void setDay(IDay day) {  this.day = day;  }   public void changeDay() {  if (day instanceof Monday) setDay(new Tuesday());  else if (day instanceof Tuesday) setDay(new Wednesday());  else if (day instanceof Wednesday) setDay(new Thursday());  else if (day instanceof Thursday) setDay(new Friday());  else if (day instanceof Friday) setDay(new Saturday());  else if (day instanceof Saturday) setDay(new Sunday());  else setDay(new Monday());  }  public void takeDrugs(){  day.takeMorningGrugs();  day.takeEveningGrugs();  }  } |
| public class Monday implements IDay {  @Override  public void takeMorningGrugs() {  System.*out*.println("Monday : Medicine #1 is used in the morning...");  }   @Override  public void takeEveningGrugs() {  System.*out*.println("Monday : Medicine #4 is used in the evening...");  } } |
| public class Sunday implements IDay {  @Override  public void takeMorningGrugs() {  System.*out*.println("Sunday : It is break!");  }   @Override  public void takeEveningGrugs() {  System.*out*.println("Sunday : It is break!");  }  } |

**Приклади виконання програми**

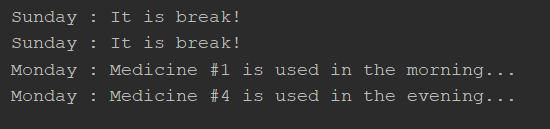
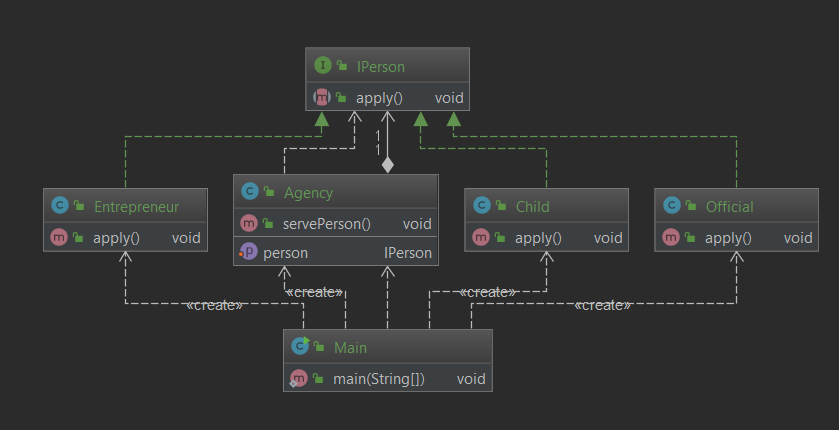


Рис. 1. Результат виконання програми

**Висновки**

Отже, ми використали клас, який змінює логіку залежно від стану об’єкту.

Завдання 2

Для виконання другого завдання було використано шаблон проектування “Strategy”. Це було зроблено тому що нам треба було змінювати поведінку системи, в залежності від конкретних умов.

**Текст програми**

|  |
| --- |
| package Strategy |
| public class Agency {  private IPerson person;   public void setPerson(IPerson person){  this.person = person;  }   public void servePerson(){  System.*out*.println("\nServing person : ");  person.apply();  System.*out*.println("Applied!!!\n");  }  } |
| public class Main {  public static void main(String[] args) {  Agency agency = new Agency();   agency.setPerson(new Official());  agency.servePerson();   agency.setPerson(new Child());  agency.servePerson();   agency.setPerson(new Entrepreneur());  agency.servePerson();   agency.setPerson(new Child());  agency.servePerson();   } } |
| public class Official implements IPerson {  @Override  public void apply() {  System.*out*.println("Official applies documents about military degree for visa ...");  } } |

**Результати виконання програми**

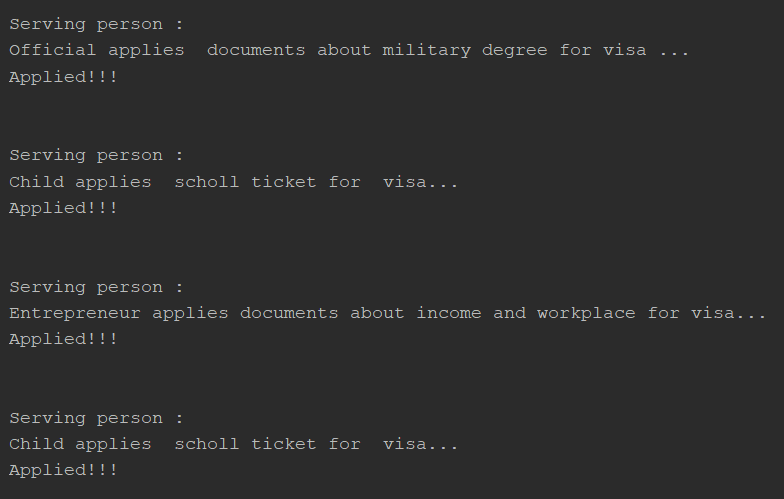


Рис. 2. Результат виконання програми

**Висновки**

Отже, за допомогою шаблону “Strategy” було забезпечено різну логіку класів в залежності від різних умов.