**Лабораторная работа №12**

Использование поведенческих шаблонов

**Цель работы**

Изучить поведенческие шаблоны.

**Ход работы**

**Задание.** Разработать программное обеспечение, содержащее поведенческие шаблоны.

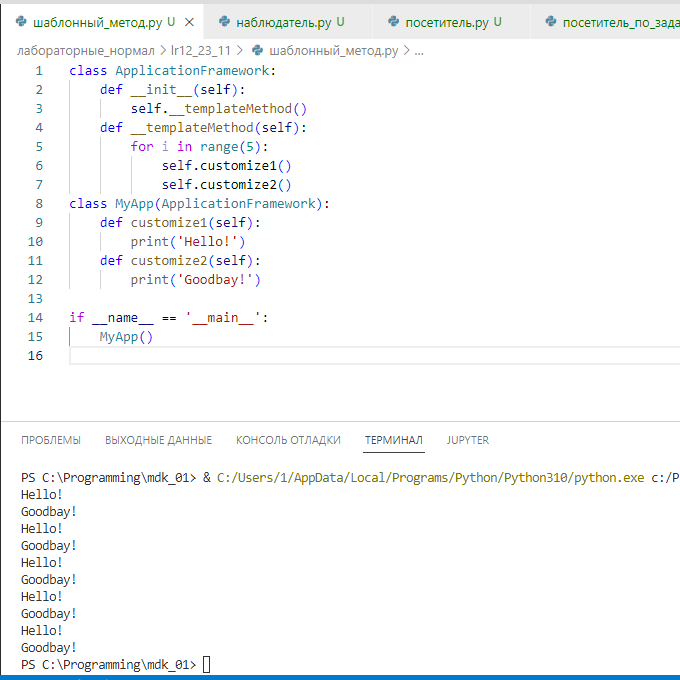


Рисунок 1 – Реализация шаблонного метода

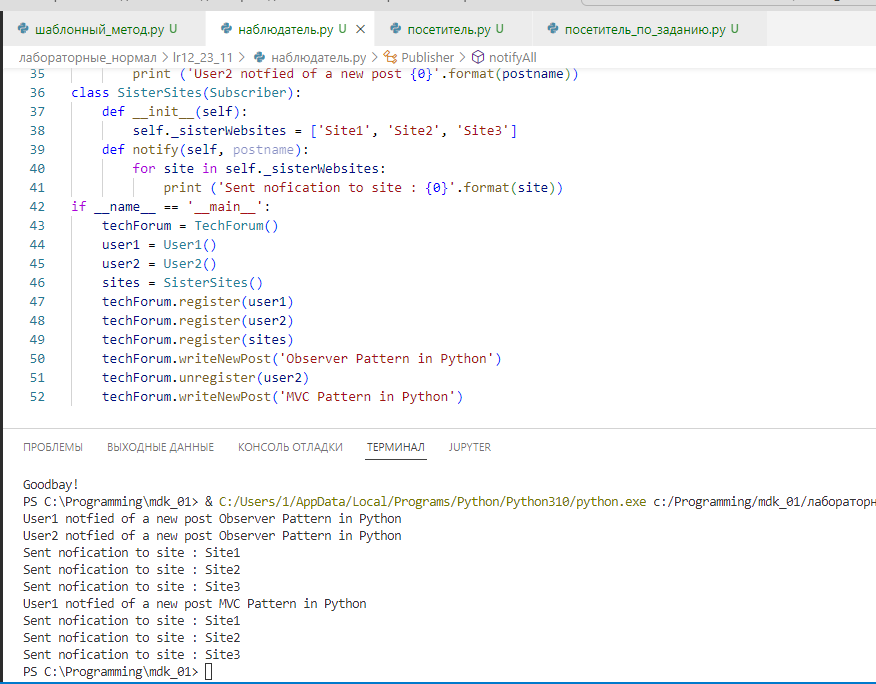


Рисунок 2 – Реализация «Наблюдателя»

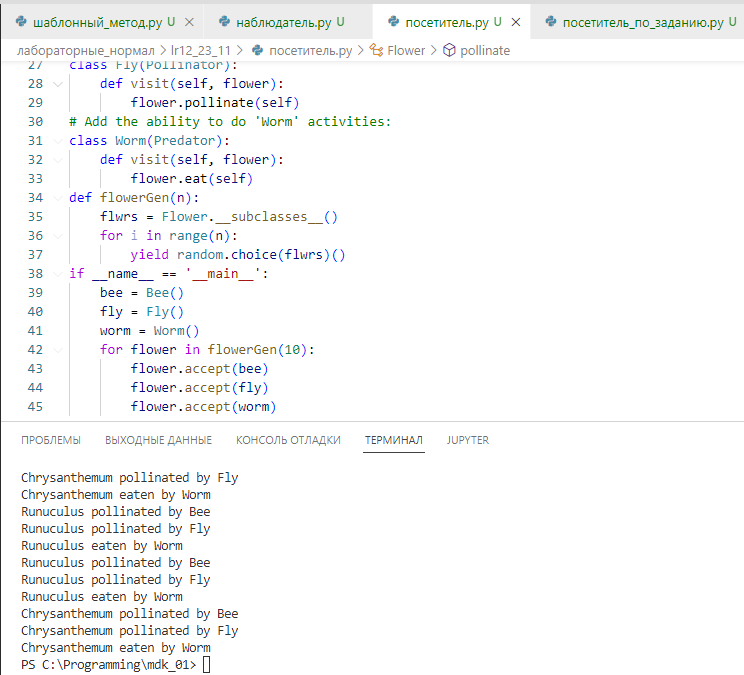


Рисунок 3 – Реализация «Посетителя»

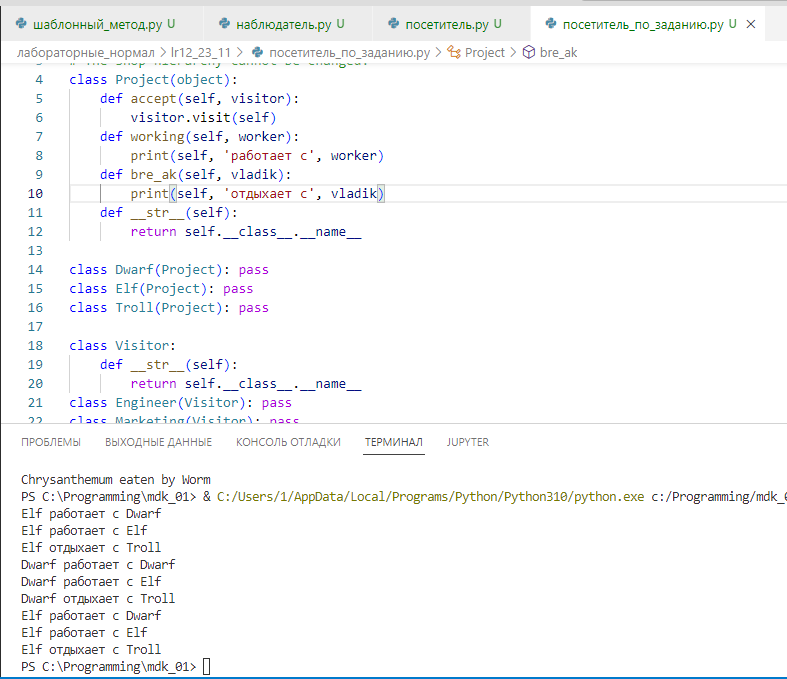


Рисунок 4 – Реализация «Посетителя» в соответствии с заданием

**Контрольные вопросы**

*1.Виды поведенческих шаблонов?*

Существуют следующие виды поведенческих шаблонов:

* Цепочка обязанностей (Chain of Responsibility);
* Команда (Command);
* Итератор (Iterator);
* Посредник (Mediator);
* Снимок (Memento)
* Наблюдатель (Observer);
* Состояние (State);
* Стратегия (Strategy);
* Шаблонный метод (Template method);
* Посетитель (Visitor).

*2.Функции поведенческих шаблонов?*

|  |  |
| --- | --- |
| Вид порождающего шаблона | Функция порождающего шаблона |
| Цепочка обязанностей (Chain of Responsibility) | Позволяет передавать запросы последовательно по цепочке обработчиков. Каждый последующий обработчик решает, может ли он обработать запрос сам и стоит ли передавать запрос дальше по цепи. |
| Команда (Command) | Превращает запросы в объекты, позволяя передавать их как аргументы при вызове методов, ставить запросы в очередь, логировать их, а также поддерживать отмену операций |
| Итератор (Iterator) | Даёт возможность последовательно обходить элементы составных объектов, не раскрывая их внутреннего  представления |
| Посредник (Mediator) | Позволяет уменьшить связанность множества классов между собой, благодаря перемещению этих связей в один класс-посредник. |
| Снимок (Memento) | Позволяет делать снимки состояния объектов, не раскрывая подробностей их реализации. Затем снимки можно использовать, чтобы восстановить прошлое состояние объектов. |
| Наблюдатель (Visitor) | Создаёт механизм подписки, позволяющий одним объектам следить и реагировать на события, происходящие в других объектах |
| Состояние (State) | Позволяет объектам менять поведение в зависимости от своего состояния. Извне создаётся впечатление, что изменился класс объекта |
| Стратегия (Strategy) | Определяет семейство схожих алгоритмов и помещает каждый из них в собственный класс. После чего, алгоритмы можно взаимозаменять прямо во время исполнения программы |
| Шаблонный метод (Template method) | Определяет скелет алгоритма, перекладывая ответственность за некоторые его шаги на подклассы. Паттерн позволяет подклассам переопределять шаги алгоритма, не меняя его общей структуры |
| Посетитель (Visitor) | Позволяет создавать новые операции, не меняя классы объектов, над которыми эти операции могут выполняться |

**Вывод**

На данной лабораторной работе я изучил поведенческие шаблоны.