

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТУ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра інформаційних систем та мереж

**Лабораторна робота № 3**  
з дисципліни  
«Системний аналіз»

**Виконав:**  
студент групи КН-208  
Келемен С. Й.  
**Викладач:**  
Бойко Н. І.

Львів – 2019 р.

# Шаблон Спостерігач

## Код

```
"""
Визначає зв'язок один до багатьох між об'єктами таким чином, що коли
один об'єкт змінює стан, всі залежні сповіщуються і оновлюються автоматично.
"""

import abc

class Thing:

    def __init__(self, name, price, number):
        self.name = name
        self.price = price
        self.number = number

    def inc_price(self):
        self.price += 2

    def dec_price(self):
        self.price -= 2

    def inc_num(self):
        self.number += 10

    def dec_num(self):
        self.number -= 10

class Subject:
    """
    Know its observers. Any number of Observer objects may observe a
    subject.
    Send a notification to its observers when its state changes.
    """

    def __init__(self):
        self._observers = set()
        self._subject_state = None

    def attach(self, observer):
        observer._subject = self
        self._observers.add(observer)

    def detach(self, observer):
        observer._subject = None
        self._observers.discard(observer)

    def _notify(self):
        for observer in self._observers:
```

```

        observer.update(self._subject_state)

    @property
    def subject_state(self):
        return self._subject_state

    @subject_state.setter
    def subject_state(self, arg):
        self._subject_state = arg
        self._notify()

class Observer(metaclass=abc.ABCMeta):
    """
    Define an updating interface for objects that should be notified of
    changes in a subject.
    """

    def __init__(self):
        self._subject = None
        self._observer_state = None

    @abc.abstractmethod
    def update(self, arg):
        pass

class User(Observer):
    """
    Implement the Observer updating interface to keep its state
    consistent with the subject's.
    Store state that should stay consistent with the subject's.
    """

    def update(self, arg):
        self._observer_state = arg

def main():
    print('\n1 - Додати користувача\n2 - Видалити користувача\n3 - Підвищити\n'
          'ціну\n4 - Знизити ціну\n5 - Збільшити кількість\n6 - Зменшити кількість\n7 -\n'
          'Переглянути стани користувачів\n0 - Вийти')
    concrete_observer = []
    users = 0
    tng = Thing('Хліб', 15, 100)
    subject = Subject()
    while (exit != True):
        choice = int(input())
        if choice == 1:
            concrete_observer.append(User())
            subject.attach(concrete_observer[users])
            print("Кількість користувачів: " + str(len(subject._observers)))

```

```

        users += 1
    elif choice == 2:
        subject.detach(concrete_observer[users-1])
        del concrete_observer[users-1]
        print("Кількість користувачів: " + str(len(subject._observers)))
        users -= 1
    elif choice == 3:
        tng.inc_price()
        print("Ціна виросла до " + str(tng.price))
    elif choice == 4:
        tng.dec_price()
        print("Ціна знизилась до " + str(tng.price))
    elif choice == 5:
        tng.inc_num()
        print("Кількість виросла до " + str(tng.number))
    elif choice == 6:
        tng.dec_num()
        print("Кількість знизилась до " + str(tng.number))
    elif choice == 7:
        for indx, i in enumerate(concrete_observer):
            print(f"Користувач №{indx+1}: ({i._observer_state.name},
{i._observer_state.price}, {i._observer_state.number})")
        elif choice == 0:
            break
    subject.subject_state = tng
    subject._notify()

if __name__ == "__main__":
    main()

```

## Результат роботи

```
1 - Додати користувача
2 - Видалити користувача
3 - Підвищити ціну
4 - Знизити ціну
5 - Збільшити кількість
6 - Зменшити кількість
7 - Переглянути стани користувачів
0 - Вийти

1
Кількість користувачів: 1
1
Кількість користувачів: 2
3
Ціна виросла до 17
3
Ціна виросла до 19
5
Кількість виросла до 110
5
Кількість виросла до 120
7
Користувач №1: (Хліб, 19, 120)
Користувач №2: (Хліб, 19, 120)
2
Кількість користувачів: 1
7
Користувач №1: (Хліб, 19, 120)
0

Process finished with exit code 0
```

## Висновок

У цій лабораторній роботі я ознайомився з патерном Спостерігач і навчився його застосовувати.