

Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційних систем та технологій

**Лабораторна робота № 5**

з дисципліни «Теорія розробки ПЗ»

Тема: ШАБЛОНИ «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF  
RESPONSIBILITY», «PROTOTYPE»

Виконав:  
студент групи ІА-14

Онуфрійчук Антон  
Валерійович

Дата здачі \_\_\_\_\_

Захищено з балом \_\_\_\_\_

Перевірів: М'який Михайло  
Юрійович

# Тема: ШАБЛОНИ «ADAPTER», «BUILDER», «COMMAND», «CHAIN OF RESPONSIBILITY», «PROTOTYPE»

## Хід роботи

### Завдання.

1. Ознайомитися з короткими теоретичними відомостями.
2. Проаналізуйте тему та намалюйте схему прецеденту, що відповідає обраній темі лабораторії.

**..15 E-mail клієнт (singleton, builder, decorator, template method, interpreter, SOA)**

Поштовий клієнт повинен нагадувати функціонал поштових програм Mozilla Thunderbird, The Bat і т.д. Він повинен сприймати і коректно обробляти

pop3/smtp/imap протоколи, мати функції автонастройки основних поштових провайдерів для України (gmail, ukr.net, i.ua), розділяти повідомлення на папки/категорії/важливість, зберігати чернетки незавершених повідомлень, прикріплювати і обробляти прикріплені файли.

### Структура класу

```
public class Message {  
    2 usages  
    private String message;  
    1 usage  
    private String from;  
    1 usage  
    private String to;  
  
    1 usage  
    private Message() {  
    }  
  
    no usages  
    public Message(String message) { this.message = message; }  
  
    1 usage  
    public static Builder builder() { return new Builder(); }  
  
    6 usages  
    public static class Builder {  
        5 usages  
        private Message message;  
  
        1 usage  
        public Builder() { this.message = new Message(); }  
  
        1 usage  
        public Builder withFrom(String from) {  
            this.message.from = from;  
            return this;  
        }  
    }  
}
```

```

1 usage
public Builder withTo(String to) {
    this.message.to = to;
    return this;
}

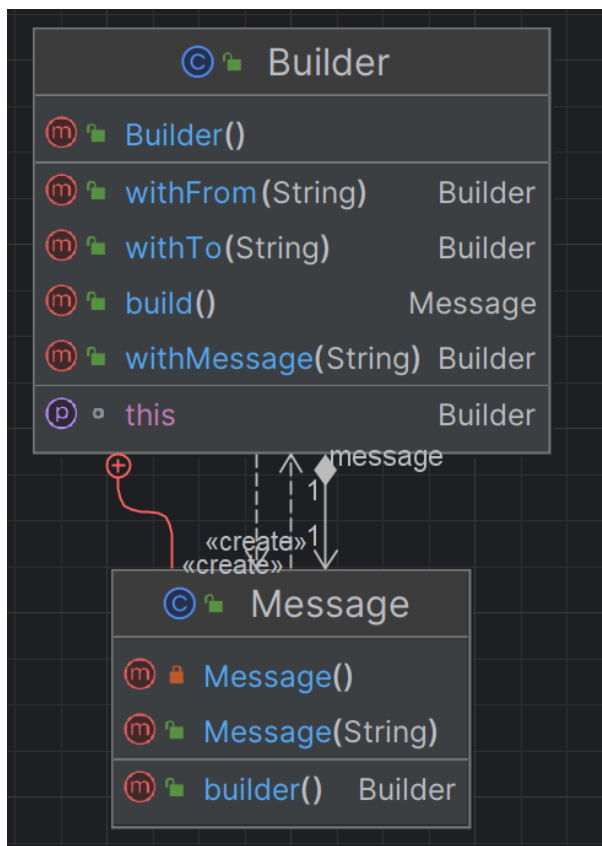
1 usage
public Builder withMessage(String message) {
    this.message.message = message;
    return this;
}

1 usage
public Message build() { return message; }

no usages
Builder getThis() {
    return this;
}
}
}

```

## Діаграма класу



**Висновок:** При виконанні лабораторної роботи реалізував патерн BUILDER.