

Київський національний університет імені Тараса Шевченка

КРЕНЕВИЧ А.П.

Методичні вказівки
до лабораторних занять із дисципліни

"Шаблони проектування програмного забезпечення"

для студентів механіко-математичного факультету

Київ – 2023

Рецензенти:

доктор фізико-математичних наук, доцент Ростислав Ямненко
кандидат фізико-математичних наук, доцент Юрій Бойко

Рекомендовано до друку вченою радою механіко-математичного
факультету
(протокол № __ від __ _____ 2023 року)

Крєневич А.П.

Методичні вказівки до лабораторних занять із дисципліни "Шаблони проектування програмного забезпечення" для студентів механіко-математичного факультету – К.: ВПЦ "Київський Університет", 2023. – 43 с.

Посібник містить перелік завдань для аудиторної та самостійної роботи з дисципліни «Шаблони проектування програмного забезпечення», що викладається студентам механіко-математичного факультету Київського національного університету імені Тараса Шевченка.

Для студентів механіко-математичного факультету та викладачів, які проводять заняття з курсу "Шаблони проектування програмного забезпечення".

ЗМІСТ

ВСТУП	5
1. UML та його застосування до зображень діаграм класів.....	6
2. Фабричний метод	9
3. Абстрактна фабрика.....	12
4. Будівельник	14
5. Прототип	16
6. Одинак	17
7. Адаптер	18
8. Міст.....	20
9. Компонувальник	22
10. Декоратор.....	24
11. Фасад.....	26
12. Легковаговик.	27
13. Замісник.....	29
14. Ланцюжок обов'язків.....	31
15. Команда.....	32
16. Ітератор.....	33
17. Посередник.....	34
18. Знімок.....	35
19. Спостерігач.....	36
20.Стан.....	37
21.Стратегія.....	39
22.Шаблонний метод.....	40
23. Відвідувач.....	41

Список літератури та додаткових джерел	43
--	----

ВСТУП

Посібник містить перелік завдань для аудиторної та самостійної роботи з дисципліни «Шаблони проектування програмного забезпечення», що викладається студентам механіко-математичного факультету Київського національного університету імені Тараса Шевченка.

Посібник призначений для роботи в парі з репозиторієм, що знаходиться на сторінці автора за посиланням

<https://github.com/krenevych/design-patterns-java>

Структура репозиторію повністю узгоджена зі структурою цього посібника. Всі завдання лежать у папці Java кореневої папки. При цьому завдання кожної лабораторної роботи знаходяться у окремій папці, назва якої починається з префіксу labXX, де XX – номер лабораторної роботи відповідно до цього посібника. Далі, кожне завдання лабораторної роботи лежить в окремій папці, та містить у назві номер задачі відповідно до цього посібника.

Постановка задач розміщені у гілці main репозиторію, у той час на гілці solutions запропоновано розв'язки аудиторних задач.

1. UML та його застосування до зображень діаграм класів.

1.1. Контрольні запитання

- 1.1.1. Як позначається клас на UML діаграмі класів? З яких частин складається позначення класу?
- 1.1.2. Яким чином позначаються модифікатори доступу на діаграмах класів?
- 1.1.3. У чому відмінність позначення інтерфейсу на діаграмі класів від позначення саме класу?
- 1.1.4. Як позначається наслідування та реалізація на діаграмах класів?
- 1.1.5. Що таке залежність у контексті зв'язків між класами та як вона позначається на UML діаграмі класів?
- 1.1.6. Що таке асоціація, у чому її відмінність від залежності? Як асоціація позначається на діаграмах класів.
- 1.1.7. Що таке потужність асоціації та які види потужності бувають?
- 1.1.8. Який тип зв'язків називається агрегацією? Як агрегація позначається на діаграмах класів?
- 1.1.9. Який тип зв'язків називається композицією? Як композиція позначається на діаграмах класів?

1.2. Завдання для аудиторної роботи

- 1.2.1. Реалізуйте структуру класів, UML діаграма яких зображена на рисунку

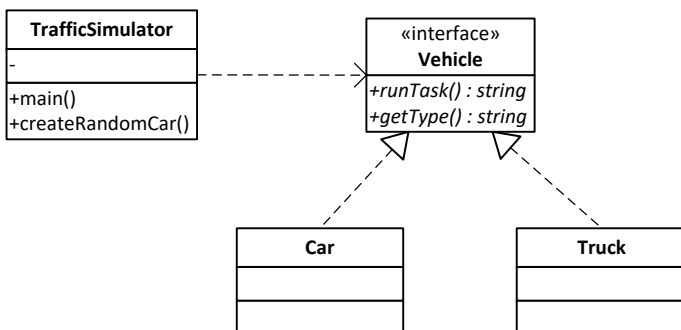


Рисунок 1.2.1. Діаграма класів.

1.2.2. Реалізуйте структуру класів, UML діаграма яких зображена на рисунку

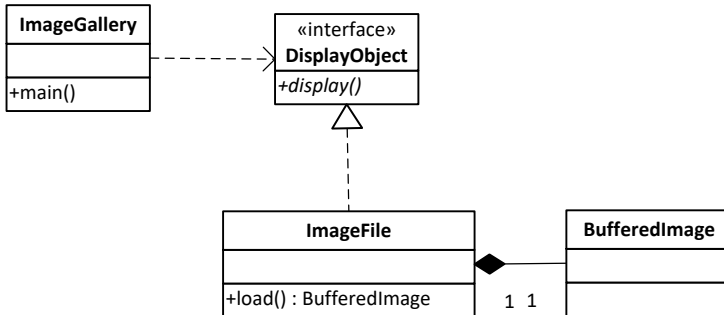


Рисунок 1.2.2. Діаграма класів.

1.2.3. Зобразить UML діаграму класів проекту, вихідний код якого розміщено за посиланням

https://github.com/krenevych/design-patterns-java/tree/main/Java/lab01_UML/task_2_3

1.2.4. Зобразить UML діаграму класів проекту, вихідний код якого розміщено за посиланням

https://github.com/krenevych/design-patterns-java/tree/main/Java/lab01_UML/task_2_4

1.3. Завдання для самостійної роботи

1.3.1. Реалізуйте структуру класів, UML діаграма яких зображена на рисунку

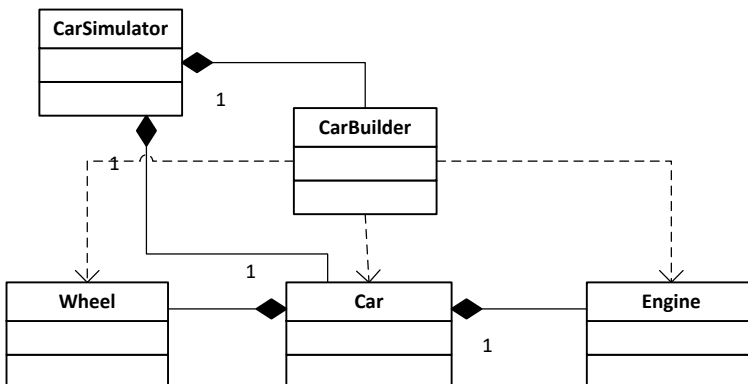


Рисунок 1.3.1. Діаграма класів.

1.3.2. Реалізуйте структуру класів, UML діаграма яких зображена на рисунку

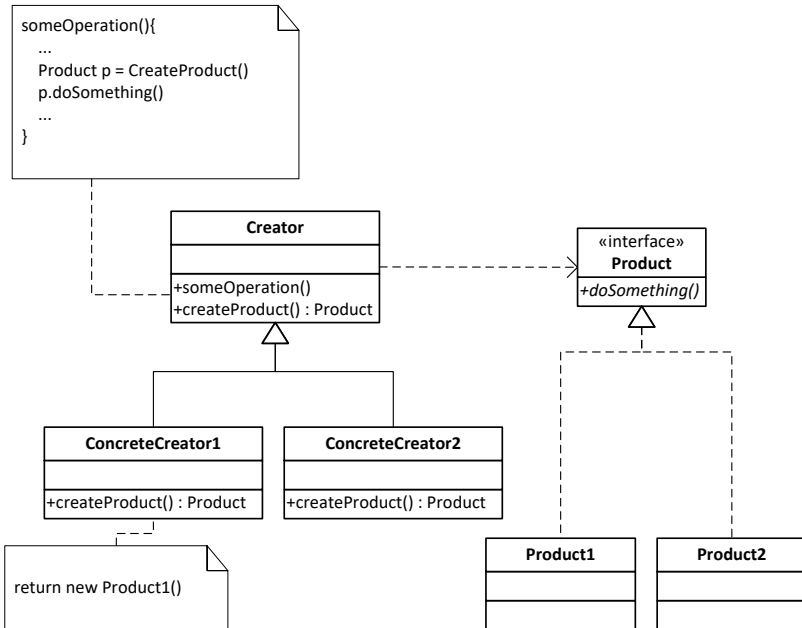


Рисунок 1.3.2. Діаграма класів.

1.3.3. Зобразіть UML діаграму класів проекту.

2. Фабричний метод

2.1. Контрольні запитання

- 2.1.1. Що таке фабрика? які види фабрик бувають?
- 2.1.2. Що таке породжуючий метод?
- 2.1.3. Що таке статичний фабричний метод?
- 2.1.4. Який шаблон програмування називається Проста фабрика? У чому недоліки і переваги цього шаблону проектування? Чому він не належить до класичних шаблонів проектування?
- 2.1.5. Який шаблон проектування називається Фабричний метод?
- 2.1.6. У яких ситуаціях застосовується (варто застосовувати) шаблон Фабричний метод?
- 2.1.7. Наведіть ідею та основні кроки реалізації шаблону проектування Фабричний метод?
- 2.1.8. Які переваги та недоліки виникають при застосуванні шаблону Абстрактна фабрика?

2.2. Завдання для аудиторної роботи

2.2.1. Розглянемо програму керування вантажними перевезеннями. Перевезенням товарів займається логістична компанія (class LogisticCompany) яка здійснює доставку товарів (class Logistics) вантажними автомобілями (class Truck). З часом компанія розростається та приймає рішення здійснювати доставку іншими видами транспорту (морським, авіа, тощо).

Проведіть рефакторинг програми з застосуванням шаблону проектування Фабричний метод. Додайте можливість реалізовувати вантажні перевезення

- морським шляхом;
- повітряним шляхом.

2.2.2. Реалізуйте шаблон фабричний метод на прикладі випуску автомобілів одного класу різними автовиробниками.

2.3. Завдання для самостійної роботи

2.3.1. Ось вже й закінчився карантин, що стримував розвиток ресторанного бізнесу. Іванко, вирішує відкрити свою власну справу і вирішує почати з невеликої кав'ярні, у ТРЦ MechMat Mall. Початково у кав'ярні він запланував готувати такі чотири напої: Еспресо, Американо, Капучіно та Латте. Кожен з цих напоїв має собівартість, що складається з сумарної вартості

компонент, що входять до напою та вартість продажу цього напою клієнту. Допоможіть Іванкові у цій не легкій справі. Змоделюйте роботу цієї кав'ярні, реалізуйте шаблон Фабричний метод та порахуйте прибуток, який отримає ця кав'ярня за певний проміжок часу. Розширте асортимент напоїв у кав'ярні іншими напоями.

2.3.2. Опишіть клас Equation (Рівняння), що моделює лінійне алгебраїчне рівняння виду

$$bx + c = 0.$$

Опишіть у ньому метод solve(), що повертає усі розв'язки цього рівняння, залежно від його коефіцієнтів. Створіть класи QuadraticEquation (Квадратне рівняння) та BiQuadraticEquation (Біквадратне рівняння), що є нащадками класу Equation та моделюють рівняння виду

$$ax^2 + bx + c = 0.$$

та

$$ax^4 + bx^2 + c = 0.$$

відповідно.

Замість у цих класах метод solve(), який розв'язує кожне з цих рівнянь, повертаючи список його розв'язків.

Використовуючи наведену ієрархію класів розв'яжіть таку задачу: Задано список рівнянь, кожне з яких визначається списком коефіцієнтів починаючи з коефіцієнта при найвищому степені.

Наприклад, лінійне рівняння $2x + 1 = 0$ задається таким списком [2, 1], квадратне рівняння $3x^2 + x + 6 = 0$ відповідно списком [3, 1, 6], а біквадратне $3x^4 + 5x^2 + 6 = 0$ відповідно списком [3, 0, 5, 0, 6].

Усі рівняння зберігається у текстовому файлі – список коефіцієнтів кожного рівняння розміщується в окремому рядку. Коефіцієнти розділені одним або кількома символами пропуску.

Серед заданих рівнянь знайдіть ті з них, що

- не мають розв'язків;
- мають один розв'язок;
- мають два розв'язки;
- мають три розв'язки;
- мають чотири розв'язки;
- мають нескінченну кількість розв'язків.

Крім цього вкажіть рівняння, що має

- найменший розв'язок,

- найбільший розв'язок,
серед розв'язків усіх рівнянь, що мають рівно один розв'язок.

Розв'яжіть задачу для кожного заданого набору вхідних даних. Вхідні дані містяться в файлах

input01.txt

input02.txt

input03.txt

Зобразіть ієрархію класів описаних у програмі за допомогою UML діаграми.

3. Абстрактна фабрика

3.1. Контрольні запитання

- 3.1.1. Що таке фабрика? які види фабрик бувають?
- 3.1.2. Який шаблон проектування називається Абстрактна фабрика?
- 3.1.3. У яких ситуаціях застосовується (варто застосовувати) шаблон Абстрактна фабрика?
- 3.1.4. Наведіть основну ідею та кроки реалізації цього шаблону проектування.
- 3.1.5. Які переваги та недоліки виникають при застосуванні шаблону Абстрактна фабрика?

3.2. Завдання для аудиторної роботи

3.2.1. Змодельуйте рушій для створення графічного інтерфейсу користувача. Реалізуйте шаблон проектування Абстрактна фабрика для створення елементів керування (Button, Text Edit, Check Button, Radio Button) в різних операційних системах (Windows, MacOS, Linux, Android).

3.2.2. Мер міста Х, Володя Скічко, вирішив оновити парк муніципального транспорту. Для цього він видає наказ закупити А автобусів, Т трамваїв та Tr – тролейбусів. Кожен вид транспорту має вартість закупки та орієнтовну вартість експлуатації, що розраховується на один кілометр пробігу. Для економії часу на підготовку та заключення договорів, а також з метою економії запчастин у подальшому, керівництво міста вирішує закупити всю техніку у одного з виробників – Volvo або Skoda. Тепер перед адміністрацією мера постало серйозне запитання, з ким із виробників заключити контракт. Допоможіть адміністрації у цьому питанні, при умові, що закуплений транспорт має пройти N тис км. Розгляньте можливість виходу на сцену третього виробника транспорту – Hyundai.

Таблиця 3.2.1. Вартість одиниці транспорту (грн)

	Автобус	Трамвай	Тролейбус
Volvo	6,000,000	10,000,000	7,000,000
Skoda	4,500,000	9,000,000	6,800,000
Hyundai	5,500,000	9,500,000	7,000,000

Таблиця 3.2.2. Вартість експлуатації транспорту (грн/км)

	Автобус	Трамвай	Тролейбус
Volvo	20	7	13

Skoda	25	8	12
Hyundai	20	6	11

3.3. Завдання для самостійної роботи

3.3.1. Після вдалого запуску власної кав'ярні (див задачу 2.3.1), Іванко націлювся спробувати продавати каву приготовану різними виробниками кавових машин, щоб знайти оптимальний варіант в контексті отриманого прибутку від продажу кави, собівартості напоїв, вартості кавової машини та вартості її обслуговування,. Допоможіть Іванкові визначитися з вибором, спираючись на той факт, що Іванко планує вийти на отримання прибутку протягом N днів. Для цього реалізуйте шаблон Абстрактна фабрика, кожна конкретна реалізація якої відповідає за певного виробника кавових машин.

4. Будівельник

4.1. Контрольні запитання

- 4.1.1. Який шаблон проектування називається Будівельник?
- 4.1.2. У яких ситуаціях застосовується (варто застосовувати) шаблон Будівельник?
- 4.1.3. Наведіть основну ідею та кроки реалізації цього шаблону проектування.
- 4.1.4. Які переваги та недоліки виникають при застосуванні шаблону Будівельник?

4.2. Завдання для аудиторної роботи

4.2.1. Процес випікання піци є складний багатокроковий процес, який включає етап підготовки тіста та додавання різних видів начинок. Для піци можуть використовуватися різних типів тіста (тонке, стандартне, тощо), кожен з яких може створюватися з різних видів борошна. Далі піца формується за рахунок різних типів та різної кількості добавок (сир, оливки, ковбаски, тощо).

Реалізуйте шаблон Будівельник для процесу виготовлення піци та визначення її вартості у піцерії. Реалізуйте клас Пиццайоло (Директор) у якому реалізуйте різні методи для виготовлення різних видів піци відповідно до меню. Зобразіть діаграму класів отриманої моделі.

4.3. Завдання для самостійної роботи

4.3.1. Процес виробництва автомобіля є складний багатокроковий процес, адже автомобіль складається зі значної кількості окремих компонент, кожна з яких у свою чергу може мати перелік власних характеристик. Розглянемо клас Автомобіль, що має такі атрибути, як тип кузова (наприклад, седан, хетчбек, SUV, тощо), двигун, трансмісія, колісні диски, гума, колір кузова, тощо. В свою чергу, атрибути автомобіля можуть мати свої власні атрибути, наприклад, двигун має потужність, робочий об'єм, крутний момент, тип пального, що використовується. Колісні диски виготовляються з різних матеріалів (сталь, легкосплавні матеріали) та бувають різного діаметру. Трансмісія в свою чергу буває механічною або автоматичною та має визначену кількість передач. Реалізуйте шаблон Будівельник для виготовлення автомобіля. Для кожної комплексної складової частини сформуєте власного будівельника. На додачу сформууйте керуючий клас Директор, задачею якого буде поєднати в одне ціле комплексний процес виготовлення автомобіля. У

класі Директор реалізуйте методи для формування різних типів та класів автомобіля.

Зобразіть діаграму класів реалізованого шаблону.

4.3.2. Реалізуйте шалон Будівельник для формування рядка. Передбачте методи додавання послідовності символів у кінець та вставки послідовності символів у вказану позицію.

5. Прототип

5.1. Контрольні запитання

- 5.1.1. Який шаблон проектування називається Прототип?
- 5.1.2. У яких ситуаціях застосовується (варто застосовувати) шаблон Прототип?
- 5.1.3. Наведіть основну ідею та кроки реалізації цього шаблону проектування.
- 5.1.4. У яких випадках не можна обійтися лише конструктором копіювання, а вимагається реалізація цього шаблону проектування?
- 5.1.5. Які переваги та недоліки виникають при застосуванні шаблону Прототип?

5.2. Завдання для аудиторної роботи

5.2.1. Реалізуйте шаблон Прототип для копіювання екземплярів класу Фігура та його нащадків.

5.2.2. Доповніть задачу 4.2.1 реалізацією шаблону Прототип для класу Піца та класів, що описують його компоненти.

5.3. Завдання для самостійної роботи

5.3.1. Реалізуйте шаблон проектування Прототип для класу Equation (Рівняння) та усіх його нащадків (задача 2.3.2).

5.3.2. Доповніть задачу 4.3.1. реалізацією шаблону Прототип для класу Автомобіль та класів, що описують його компоненти.

6. Одинак

6.1. Контрольні запитання

- 6.1.1. Який шаблон проектування називається Одинак?
- 6.1.2. У яких ситуаціях застосовується (варто застосовувати) шаблон Одинак?
- 6.1.3. Наведіть основну ідею та кроки реалізації цього шаблону проектування.
- 6.1.4. Які переваги та недоліки виникають при застосуванні шаблону Одинак?
- 6.1.5. Які основні небезпеки ховає в собі шаблон проектування Одинак при в контексті багатопоточного програмування?

6.2. Завдання для аудиторної роботи

6.2.1. Реалізуйте

- шаблон проектування Одинак з реалізацією на старті програми;
- шаблон проектування Одинак з відкладеною ініціалізацією;
- потоко-безпечний шаблон проектування Одинак з відкладеною ініціалізацією.

6.2.2. Реалізуйте потоко-безпечний шаблон проектування Одинак з відкладеною ініціалізацією для доступу до бази даних автомобілів. Для спрощення будемо вважати, що база даних містить пари – номер автомобіля та ім'я її власника.

6.2.3. У задачі 4.2.1 застосуйте шаблон проектування Одинак для можливості існування лише одного Піццоло у піцерії.

6.3. Завдання для самостійної роботи

6.3.1. Реалізуйте шаблон проектування Одинак для читання та обробки текстових файлів у програмі.

6.3.2. У задачі 4.3.1 застосуйте шаблон проектування Одинак для можливості існування лише одного Директора.

7. Адаптер.

7.1. Контрольні запитання

- 7.1.1. Для чого застосовується шаблон проектування Адаптер.
- 7.1.2. Наведіть основні кроки реалізації шаблону проектування Адаптер.
- 7.1.3. Які два способи реалізації шаблону проектування Адаптер ви можете назвати? Який спосіб більш рекомендований?

7.2. Завдання для аудиторної роботи

7.2.1. Розглянемо деяку спостережну станцію, що моніторить температури в різних містах Північної Америки та Азії і видає попередження, якщо температура виходить за межі допустимої норми. Проте, у містах Північної Америки для вимірювання температури використовують Фаренгейти, у той час як у Азії – градуси Цельсія. Коли розроблялася спостережна станція, то планувалося, що вона буде використовуватися лише у Північній Америці, відтак, її подальше впровадження у країни Азії було призупинено. Вам необхідно реалізувати Адаптер, який дозволить поширити використання станції на країни Азії.

Реалізуйте обидва підходи реалізації Адаптера, як через наслідування, так і через агрегацію.

7.2.2. Реалізуйте шаблон проектування Адаптер для моделювання процесу заряджання мобільного телефона з роз'ємом USB Type-C через зарядний пристрій з роз'ємом micro-USB.

Реалізуйте обидва способи реалізації цього шаблону проектування – як через композицію так і через наслідування. Порівняйте отримані реалізації.

7.3. Завдання для самостійної роботи

7.3.1. Для задачі 7.2.2 реалізуйте шаблон проектування Адаптер який дозволить використовувати зарядний пристрій з інтерфейсом USB Type-C для заряджання мобільних інтерфейсів зі стандартом micro-USB.

Проаналізуйте реалізовані шаблони та дайте відповідь на питання, чи можна для поставленої задачі створити двосторонній адаптер.

7.3.2. На американському аукціоні для визначення вартості вживаного автомобіля використовується спеціальна програма – Автомобільний калькулятор – яка обчислює вартість автомобіля залежно від моделі, його віку, пробігу та ступеня пошкодження внаслідок дорожньо-транспортної пригоди.

Вартість автомобіля обчислюється методом `calculatePrice()`, що повертає результат у доларах США. Українська митниця вирішила для обчислення вартості автомобіля, розміру мита та повної вартості автомобіля після розмитнення використовувати цей автомобільний калькулятор. Проте, програмне забезпечення, що використовує митниця вже розроблене та затверджене, крім цього всі розрахунки на митниці ведуться у гривнях. Реалізуйте шаблон Адаптер, для того, щоб адаптувати програмне забезпечення Автомобільного калькулятора для потреб української митниці.

Нижче наведено орієнтовну діаграму класів Автомобільного калькулятора та української митниці, відповідно.

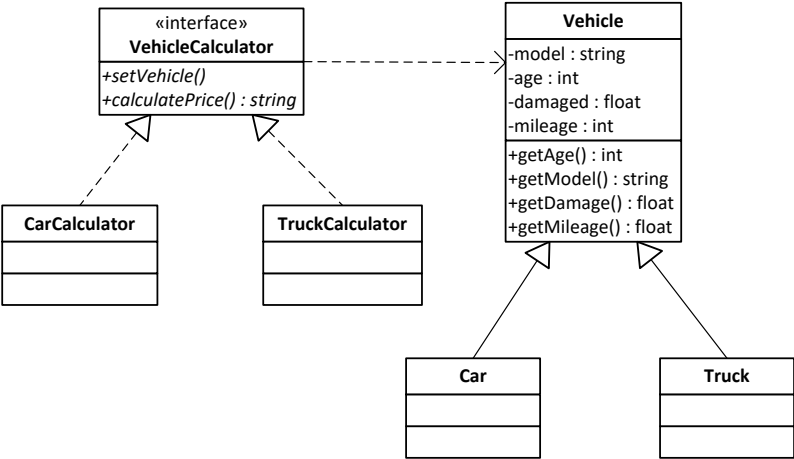


Рисунок 7.3.1. Діаграма класів Автомобільного калькулятора.

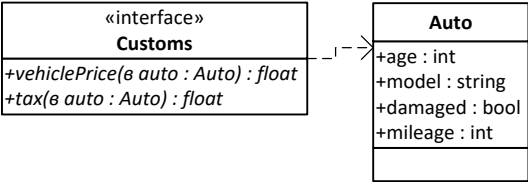


Рисунок 7.3.2. Діаграма класів української митниці.

8. Міст.

8.1. Контрольні запитання

- 8.1.1. Для чого застосовується шаблон проектування Міст?
- 8.1.2. Що таке абстракція та реалізація в контексті шаблону проектування Міст?
- 8.1.3. Наведіть основні етапи реалізації шаблону проектування Міст.

8.2. Завдання для аудиторної роботи

8.2.1. Проведіть рефакторинг коду, із застосуванням шаблону проектування Міст.

8.2.2. Розглянемо прилад Телевізор, яким можна керувати за допомогою пульта. За допомогою цього пульта телевізор можна вмикати/вимикати, збільшувати/зменшувати гучність а також повністю вимикати/вмикати вимикати звук. Також за допомогою пульта можна перемикає канали та здійснювати інше керування Телевізором.

Реалізуйте шаблон проектування Міст, щоб розділити монолітний код приладів та пультів керування до них. Реалізацію здійсніть так, щоб прилади брали участь у реалізації шаблону як «реалізацію», а пульти керування ними – відповідно «абстракцією» (клієнт працює з абстракцією, відповідно з пультом).

Додайте кілька різних пристроїв, якими можна керувати за допомогою цього пульта – Телевізор, Радіо тощо.

8.3. Завдання для самостійної роботи

8.3.1. Розглянемо діаграму класів, що наведена на рисунку 8.3.1, яка описує структуру напоїв, що виготовляються у деякій невеликій кав'ярні.

Проведіть рефакторинг класів, цього проекту з застосуванням шаблону проектування Міст. По тому, додайте відповідні класи для можливості готування напоїв для вживання їх як в ресторані, так і «на винос».

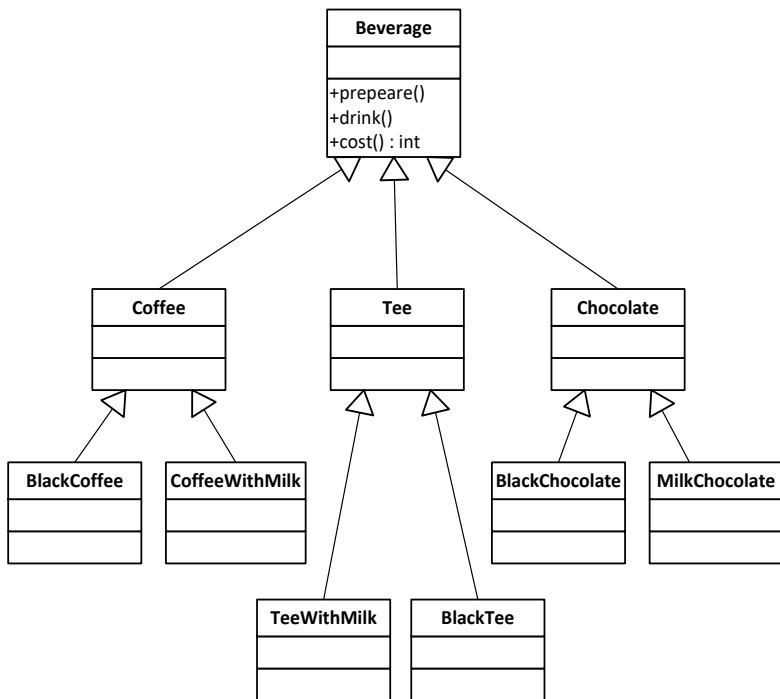


Рисунок 8.3.1. Діаграма класів напоїв кафе.

8.3.2. Проведіть рефакторинг коду із застосуванням шаблону проектування Міст. Після реалізації шаблону Міст, розширте проект, додавши новий клас ImageButton (Графічна кнопка), та новий клас UserSize (Користувачський розмір).

9. Компонувальник

9.1. Контрольні запитання

9.1.1. Для чого застосовується шаблон проектування Компонувальник?

9.1.2. Наведіть основні етапи реалізації шаблону Компонувальник.

9.2. Завдання для аудиторної роботи

9.2.1. Реалізуйте базовий шаблон проектування Компонувальник відповідно до означення.

9.2.2. Розглядається бібліотека, що може містити художні та документальні книжки. Кожна книга, як художня так і документальна має назву та ознаку чи вона доступна для замовлення у бібліотеці. Крім цього художня книжка має означу чи є твір викладений в ній п'єсою.

Для кожної книги, зокрема, та бібліотеки, в цілому, визначені операції – замовити (тобто, забронювати) книгу, якщо вона доступна для резервування, та повернути її до бібліотеки (зробити доступною для замовлення іншим клієнтам).

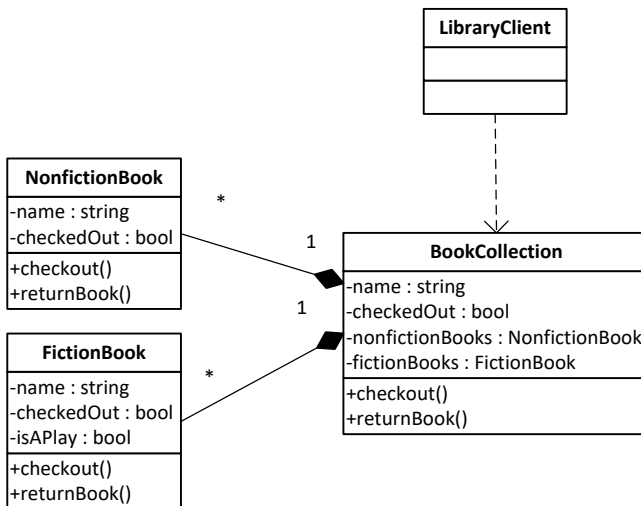


Рисунок 9.2.1. Діаграма класів Бібліотеки.

Проведіть рефакторинг коду, із застосуванням шаблону проектування Компонувальник для більш гнучкого керування бібліотекою.

9.3. Завдання для самостійної роботи

9.3.1. Розглянемо модель магазину, що містить два типи співробітників – менеджери та продавці. Всі співробітники магазину отримують заробітну платню відповідно до посади. Наша модель нараховує заробітну плату для всіх співробітників магазину.

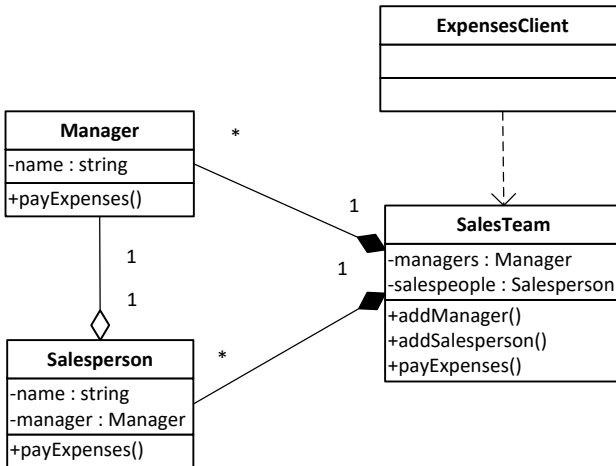


Рисунок 9.3.1. Діаграма класів Бібліотеки.

Проведіть рефакторинг коду, із застосуванням шаблону проектування Компонувальник так щоб програма могла розрахувати виплату заробітної плати працівникам супермаркету, що очолюється директором (менеджером) містить три відділи (м'ясний відділ, відділ молокопродуктів, кондитерський відділ), кожен з яких очолюється менеджером, та містить принаймні трьох співробітників-продавців.

10. Декоратор.

10.1. Контрольні запитання

- 10.1.1. Для чого застосовується шаблон проектування Декоратор?
- 10.1.2. Наведіть основні етапи реалізації шаблону проектування Декоратор.
- 10.1.3. У чому відмінність та що спільного у шаблонів проектування Декоратор та Адаптер?
- 10.1.4. Що спільного у структурі класів Декоратора та Компонувальника?

10.2. Завдання для аудиторної роботи

10.2.1. Розглянемо клас `Circle`, що використовується для зображення плоских кіл на площині. Кожен екземпляр визначається позицією центра кола і значенням радіуса.

Опишіть декоратор, що застосовується до екземплярів класу `Circle` для зображення їх різними кольорами – червоним, синім та жовтим.

10.2.2. Розглянемо клас `Pizza` (Піца) та його нащадки `ThickCrustPizza` (піца на товстому тісті) та `ThinCrustPizza` (піца на тонкому тісті), що мають методи `description()` та `cost()`, що повертають опис піци та її вартість відповідно.

Опишіть декоратори, для додавання у до екземплярів вищенаведених класів додаткових інгредієнтів таких як сир, оливки та перець.

Використовуючи цей декоратор створіть піцу з двома додатковими порціями сиру, однією порцією оливки та порцією перцю.

10.3. Завдання для самостійної роботи

10.3.1. Розглянемо клас `PrintableString`, що є контейнером для поля типу рядок та містить єдиний публічний метод для виведення цього рядка у консоль.

Для цього класу опишіть набір декораторів:

- `PostComaDecorator` – декоратор для додавання знаку кома у кінець рядка.
- `PostEndlDecorator` – декоратор для додавання знаку переходу на наступний рядок у кінець рядка.
- `PostSpaceDecorator` – декоратор для додавання знаку пропуску у кінець рядка

- `PostExclaimDecorator` – декоратор для додавання знаку оклику у кінець рядка.
- `PostWordDecorator` – декоратор для додавання слова у кінець рядка.
- `PreWordDecorator` – декоратор для додавання слова у початок рядка.

Використовуючи цей набір декораторів виведіть у консоль вираз

«Hello, Wold»

починаючи з порожнього базового рядка.

10.3.2. Розглядається `Beverage` (Напій) та його нащадки `Espresso` (Еспресо), `DarkRoast` (Чорна кава), `Decaf` (Кава без кофеїну), що мають методи `description()` та `cost()`, що повертають опис кави та її вартість відповідно.

Опишіть декоратори для додавання до екземплярів вищенаведених класів додаткові наповнювачі – молоко, цукор, вершки, крем.

Використовуючи цей декоратор, створіть такі напої:

- Еспресо з двома порціями цукру.
- Чорну каву з вершками та двома порціями цукру.
- Чорну каву з кремом та порцією цукру.
- Кава без кофеїну з молоком та двома порціями цукру.

11. Фасад.

11.1. Контрольні запитання

- 11.1.1. Для чого застосовується шаблон проектування Фасад?
- 11.1.2. Наведіть основні етапи реалізації шаблону Фасад.
- 11.1.3. Наведіть основні переваги та недоліки шаблону проектування Фасад.
- 11.1.4. Наведіть приклади відносин шаблону проектування Фасад з іншими шаблонами проектування.

11.2. Завдання для аудиторної роботи

11.2.1. Організація відпустки або відрядження вимагає виконання деякого набору рутинних операцій таких як купівля квитків на літак, замовлення готелю та оренда автомобілю.

Реалізуйте шаблон програмування Фасад, для того щоб спростити рутинну послідовність дій для організації відпустки.

11.2.2. Режим керування автомобілем передбачає багато функцій таких як старт двигуна, повороти керма та перемикання режимів трансмісії.

У клієнтському коді використовується трансмісія з механічним перемиканням передач. Реалізуйте шаблон проектування Фасад, який перетворить механічну трансмісію у роботизовану.

11.3. Завдання для самостійної роботи

11.3.1. Конвертування відео з одного формату в інший це багатокроковий процес, що вимагає використання цілої серії допоміжних інструментів.

Замість цього можна реалізувати шаблон проектування Фасад, що надає користувачеві лише один простий метод, приховуючи за собою цілу систему з відеокодеками, аудіомікшерами та іншими, не менш складними об'єктами.

Цей метод повинен приймати на вхід файл, який необхідно перекодувати та формат у який необхідно перетворити файл

Проведіть рефакторинг коду із застосуванням шаблону проектування Фасад, щоб спростити роботу клієнта зі складною бібліотекою відеоконвертації.

12. Легковаговик.

12.1. Контрольні запитання

- 12.1.1. Для чого застосовується шаблон проектування Легковаговик?
- 12.1.2. Наведіть основні етапи реалізації шаблону Легковаговик.
- 12.1.3. Наведіть приклади відносин шаблону проектування Легковаговик з породжувальними шаблонами проектування.

12.2. Завдання для аудиторної роботи

12.2.1. Розглядається проста модель перевезень вантажним транспортом вантажів та пасажирів легковим транспортом.

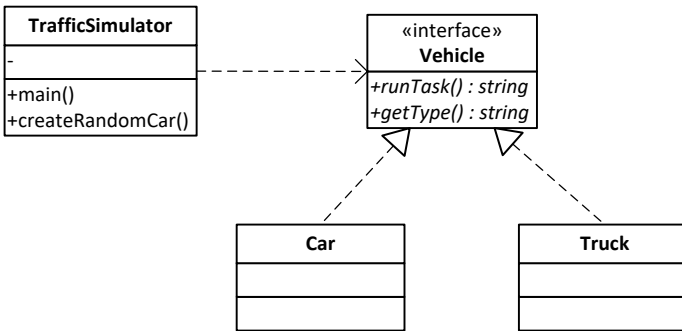


Рисунок 12.2.1. Діаграма класів симулятора перевезень.

При цьому клієнт TrafficSimulator багаторазово створює та використовує об'єкти екземплярів класів Car та Truck, що реалізують інтерфейс Vehicle.

Реалізуйте шаблон проектування Легковаговик, для повторного використання об'єктів класів Car та Truck.

12.3. Завдання для самостійної роботи

12.3.1. Більшість автомобілів складається з великої кількості деталей таких як двигун, колеса, кермо тощо. Під час збірки автомобіля у нього встановлюють типові запчастини, що повторюються у багатьох модифікаціях. Відповідно, програмуючи деяку гру чи симулятор, що використовує подібні ієрархії класів можна суттєво заощадити оперативну пам'ять та зменшити час

створення об'єктів класу Car, якщо закешувати однакові деталі. Нижче наведено просту ієрархію такого симулятора.

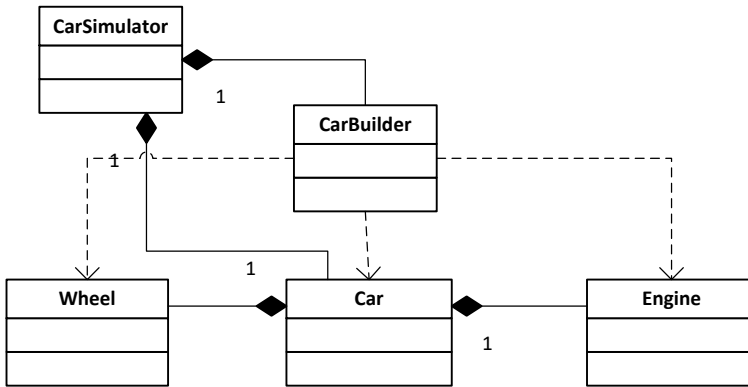


Рисунок 12.3.1. Діаграма класів .

Реалізуйте шаблон проектування Легковаговик для повторного використання створених раніше об'єктів для різних частин автомобіля.

13. Замісник.

13.1. Контрольні запитання

13.1.1. Для чого застосовується шаблон проектування Замісник?

13.1.2. Наведіть основні етапи реалізації шаблону Замісник.

13.1.3. Наведіть зв'язок шаблону проектування Замісник з шаблонами проектування Адаптер та Декоратор

13.2. Завдання для аудиторної роботи

13.2.1. Розглядається симулятор проектів, діаграма класів якого наведено на рисунку нижче

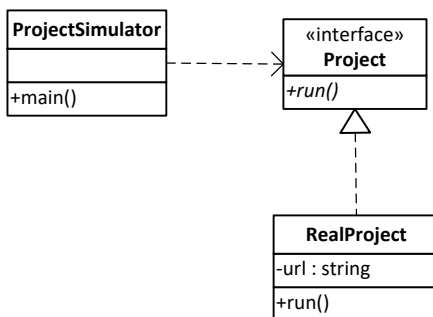


Рисунок 13.2.1. Діаграма класів .

Клієнт **ProjectSimulator** завантажує перелік проектів зі сервера GitHub відповідно до їхніх `url` адрес та запускає один з проектів на виконання.

Реалізуйте шаблон проектування Замісник, таким чином, щоб уникнути завантаження проектів з сервера, якщо вони не будуть запускатися на виконання.

13.3. Завдання для самостійної роботи

13.3.1. Розглядається програма призначена для послідовного виведення зображень на екран. Діаграма класів проектів наведена на рисунку нижче

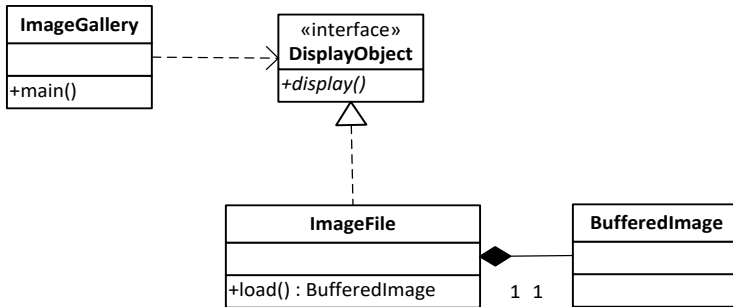


Рисунок 13.3.1. Діаграма класів.

Реалізуйте шаблон проектування Замісник, таким чином, щоб уникнути завантаження зображень з носія інформації, до безпосереднього їхнього виведення на екран.

14. Ланцюжок обов'язків.

14.1. Контрольні запитання

14.1.1. Для чого застосовується шаблон проектування Ланцюжок обов'язків?

14.1.2. Наведіть основні етапи реалізації шаблону Ланцюжок обов'язків.

14.2. Завдання для аудиторної роботи

14.2.1. Розглядається система, що виконує заданий набір запитів різних типів. Для кожного запиту виконується набір команд, відповідно до типу цього запиту. Реалізуйте шаблон проектування Ланцюжок обов'язків для обирання необхідного набору операцій, відповідно до типу запиту.

14.2.2. Під час створення фотографії цифровою камерою, до отриманого файлу додається велика кількість додаткових атрибутів, таких як ім'я, час створення фотографії, місце створення фотографії, тип та назва камери (фотоапарату) яким було створено фотографію тощо. Реалізуйте модуль для додавання усіх необхідних атрибутів до файлу реалізувавши шаблон проектування Ланцюжок обов'язків.

14.3. Завдання для самостійної роботи

14.3.1. Реалізуйте механізм аутентифікації для надання прав користувачам залежно від їхнього типу на базі логіну та паролю.

14.3.2. Реалізуйте програму для (симуляції) відкриття файлів різних типів. Реалізацію здійсніть використовуючи шаблон проектування Ланцюжок обов'язків.

Вказівка: тип файлу, що відкривається визначається його розширенням.

14.3.3. Реалізуйте симулятор, що опрацьовує запити на виконання арифметичних операцій. Кожен запит складається з двох чисел та типу операції, що має над ними виконатися. Реалізацію симулятора здійсніть з використанням шаблону проектування Ланцюжок обов'язків.

15. Команда.

15.1. Контрольні запитання

- 15.1.1. Для чого застосовується шаблон проектування Команда?
- 15.1.2. Наведіть основні етапи реалізації шаблону Команда.
- 15.1.3. Що спільного та чим відрізняється шаблон проектування Команда з такими шаблонами як Ланцюг обов'язків, Спостерігач та посередник?

15.2. Завдання для аудиторної роботи

- 15.2.1. Реалізуйте шаблон проектування Команда відповідно до класичної UML діаграми класів.
- 15.2.2. Змодельуйте роботу пульта керування увімкнення та вимкнення світильника.

15.3. Завдання для самостійної роботи

- 15.3.1. Змодельуйте роботу пульта керування увімкнення та вимкнення світильників. Реалізуйте універсальний пульт, що здійснює вимикання світла усіх світильників у системі.
- 15.3.2. Розглянемо роботу (симуляцію роботи) з текстовим редактором, що може виконувати кілька операцій, таких як створювати документ, зберігати його, виводити на друк тощо. Операції в текстовому редакторі мають здійснюватися натиском кнопок користувацького інтерфейсу, вибираючи відповідний пункт з меню або натискаючи на клавіатурі відповідну комбінацію клавіш.
Додайте у проект реакцію натиску на кнопки користувацького інтерфейсу, вибір відповідного пункту меню та реакцію на натиск комбінацій швидких клавіш, реалізувавши для цього шаблон проектування Команда.
- 15.3.3. Змодельуйте роботу пульта дистанційного керування для телевізора та радіоприймача. Реалізуйте універсальний пульт, що вимикає всі пристрої в системі.

16. Ітератор.

16.1. Контрольні запитання

16.1.1. Для чого застосовується шаблон проектування Ітератор?

16.1.2. Наведіть основні етапи реалізації шаблону Ітератор.

16.2. Завдання для аудиторної роботи

16.2.1. Реалізуйте шаблон проектування Ітератор відповідно до діаграми класів цього шаблону.

16.2.2. Розглянемо колекцію, що містить набір цілих чисел. Нехай у колекції реалізовані операції вставки нових елементів, звернення до елемента колекції за його позиційним номером в цій колекції тощо.

Реалізуйте для цієї колекції шаблон проектування Ітератор, що допомагає перебирати елементи колекції у порядку зростання.

16.3. Завдання для самостійної роботи

16.3.1. Реалізуйте шаблон проектування Ітератор, для послідовного перебору працівників деякої компанії відповідно до алфавітного порядку.

16.3.2. Розглядається колекція елементів, що містить назву товару та його кількість у магазині. Реалізуйте шаблон проектування Ітератор для перебору усіх товарів у магазині

- За алфавітним порядком назв товарів
- За спаданням кількості товарів

16.3.3. Розглядається модель у якій є елементи двох типів – Вох (Коробка) та Item (Елемент). Коробка може містити як інші коробки, так і звичайні елементи. Для моделювання такої системи реалізовано шаблон проектування Компонувальник.

Реалізуйте шаблон проектування Ітератор, для перебору всіх елементів моделі використовуючи

- пошук в глибину;
- пошук в ширину;

для обходу всіх вузлів дерева Компонувальника.

Використовуючи цей Ітератор знайдіть

- найдешевший елемент у колекції;
- найдорожчий елемент у колекції;

17. Посередник.

17.1. Контрольні запитання

17.1.1. Для чого застосовується шаблон проектування Посередник?

17.1.2. Наведіть основні етапи реалізації шаблону Посередник.

17.2. Завдання для аудиторної роботи

17.2.1. Реалізуйте шаблон проектування Посередник, для обміну повідомленнями між елементами деякої системи.

17.2.2. Проведіть рефакторинг коду, що моделює роботу інтернет-магазину для замовлення та доставлення товарів споживачем. Застосуйте шаблон проектування Посередник для відправлення повідомлень.

17.2.3. Реалізуйте симулятор чату для обміну повідомленнями між користувачами. Передбачте функції надсилання повідомлень усім користувачам чату, конкретним користувачам. Реалізацію здійсніть із застосуванням шаблону проектування Посередник.

17.3. Завдання для самостійної роботи

17.3.1. Проведіть рефакторинг коду, що здійснює симуляцію керування польотами літаків. Використайте шаблон проектування Посередник для обміну повідомленнями між усіма елементами системи.

17.3.2. Розширте функціонал системи, реалізованої під час виконання задачі 17.2.3 так, щоб повідомлення можна було розсилати окремим групам користувачів, наприклад, адміністраторам/модераторам чату.

18. Знімок.

18.1. Контрольні запитання

18.1.1. Для чого застосовується шаблон проектування Знімок?

18.1.2. Наведіть основні етапи реалізації шаблону Знімок.

18.1.3. Які спільні риси має шаблон проектування Знімок з шаблоном проектування Прототип?

18.2. Завдання для аудиторної роботи

18.2.1. Реалізуйте шаблон проектування Знімок, використовуючи реалізацію на основі внутрішніх класів.

18.2.2. Реалізуйте шаблон проектування Знімок, використовуючи реалізацію з проміжним порожнім інтерфейсом.

18.2.3. Реалізуйте шаблон проектування Знімок, використовуючи Знімки з підвищеним захистом.

18.3. Завдання для самостійної роботи

18.3.1. Розглянемо клас Трикутник, який містить три поля, які визначають довжини сторін цього трикутника. Для Трикутника визначені операції зміни його сторін, у яких здійснюється перевірка чи буде існувати трикутник після такої операції. Крім цього для цього трикутника визначені операції обчислення площі та периметра. Реалізуйте шаблон проектування Знімок для зберігання історії змін сторін трикутника та відновіть стан трикутника у якому трикутник мав найбільшу площу.

Реалізуйте всі три способи реалізації шаблону проектування Знімок:

- на основі внутрішніх класів;
- реалізацію з порожнім інтерфейсом;
- з підвищеним захистом.

18.3.2. Розглянемо клас Трикутник описаний в задачі 18.3.1. Визначимо додатково в задачі операції збільшення та зменшення сторін трикутника вдвічі. Реалізуйте шаблон проектування Знімок в парі з шаблоном проектування Команда для зберігання історії змін та відновіть стан трикутника у якому він мав найменшу площу.

19. Спостерігач.

19.1. Контрольні запитання

- 19.1.1. Для чого застосовується шаблон проектування Спостерігач?
- 19.1.2. Наведіть основні етапи реалізації шаблону Спостерігач.
- 19.1.3. Наведіть аналогії з реального життя шаблону проектування Спостерігач
- 19.1.4. Наведіть приклади відносин шаблону Спостерігач з іншими поведінковими шаблонами проектування такими Команда та Посередник.

19.2. Завдання для аудиторної роботи

19.2.1. Розглядається набір метеорологічних станцій, що розташовуються у різних містах світу. Кожна станція, публікує такі погодні показники як температура повітря, вологість та атмосферний тиск. Всі опубліковані показники надходять у єдиний дата-центр.

Використовуючи зазначений симулятор дата-центру, реалізуйте три сервіси, що виводять повідомлення-попередження про те що в одному з міст температура, вологість чи атмосферний тиск виходять за межі допустимих норм.

19.3. Завдання для самостійної роботи

19.3.1. Розглянемо систему для порядкового зчитування текстових файлів.

Використовуючи шаблон проектування Спостерігач, реалізуйте механізми для:

- відшукування найдовшого рядка файлу;
- відшукування найдовшого слова, що міститься у файлі;
- підрахунку кількості слів у файлі;
- знаходженні рядка файлу, що містить найдовше слово.

20.Стан.

20.1. Контрольні запитання

20.1.1.Для чого застосовується шаблон проектування Стан?

20.1.2.Наведіть основні етапи реалізації шаблону Стан.

20.2. Завдання для аудиторної роботи

20.2.1. Реалізуйте шаблон проектування Стан відповідно до його діаграми класів.

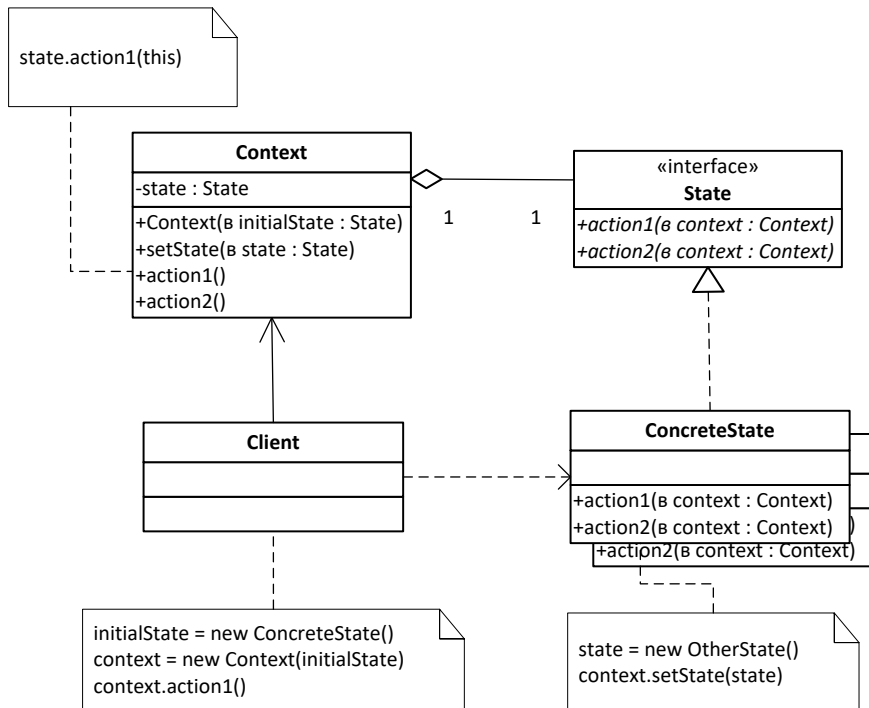


Рисунок 20.2.1. Діаграма класів шаблону Стан.

20.2.2. Реалізуйте шаблон проектування Стан на прикладі документу, що може знаходитися в одному з трьох станів – чернетка, модерування, опублікований. Схема переходу зі стану до стану зображена на рисунку нижче

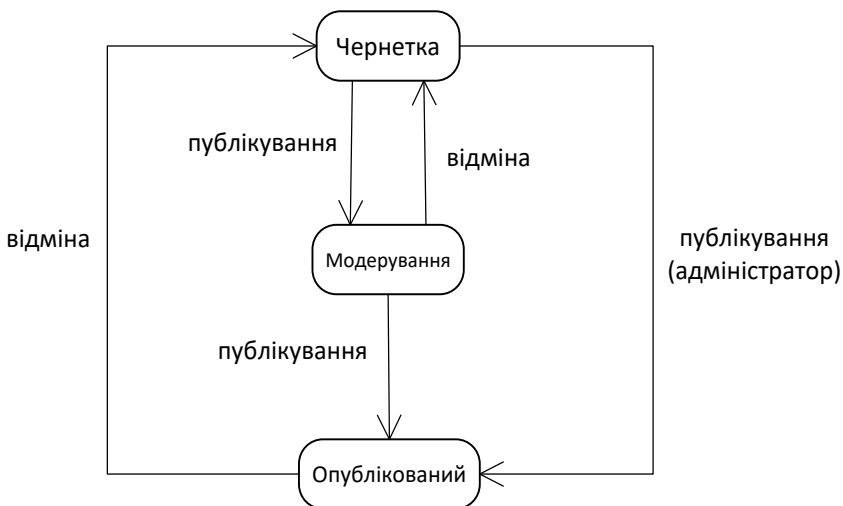


Рисунок 20.2.2. Схема переходів між станами документу.

20.3. Завдання для самостійної роботи

20.3.1. Проведіть рефакторинг коду програмного продукту застосувавши шаблон проектування Стан.

20.3.2. Реалізуйте симуляцію роботи музичного плеєра використовуючи шаблон проектування Стан.

20.3.3. Проведіть рефакторинг коду програмного продукту, застосувавши шаблон проектування Стан.

21.Стратегія.

21.1. Контрольні запитання

21.1.1.Для чого застосовується шаблон проектування Стратегія?

21.1.2.Наведіть основні етапи реалізації шаблону Стратегія.

21.2. Завдання для аудиторної роботи

21.2.1. Реалізуйте симулятор автомобільного навігатора. Шаблон проектування Стратегія обирається для вибору способу побудови маршруту: автомобільним транспортом, велосипедом тощо.

21.2.2. Існує безліч алгоритмів шифрування, таких як AES, DES, RSA, ECC і т. д. Розглядається симуляція процесів шифрування файлів.

Необхідно провести рефакторинг коду, використовуючи шаблон проектування Стратегія, так, щоб спосіб вибору алгоритму шифрування визначався застосуванням конкретної стратегії.

21.3. Завдання для самостійної роботи

21.3.1. У сучасному житті існують безліч систем оплати в інтернеті: перерахунок з банківського рахунку, PayPal, Google Pay і т. д. Розглядається моделювання процесу оплати в інтернеті.

Необхідно провести рефакторинг коду, використовуючи шаблон проектування Стратегія, так, щоб спосіб вибору методу оплати визначався застосуванням конкретної стратегії.

21.3.2. Розглядається модель калькулятора, що може робити прості арифметичні дії над двома операндами: додавання, віднімання та множення.

Реалізуйте шаблон проектування Стратегія при якому конкретна операція визначається конкретною стратегією.

22.Шаблонний метод.

22.1. Контрольні запитання

22.1.1.Для чого застосовується шаблон проектування Шаблонний метод?

22.1.2.Наведіть основні етапи реалізації шаблону Шаблонний метод.

22.2. Завдання для аудиторної роботи

22.2.1. Реалізуйте шаблон проектування Шаблонний метод відповідно до його діаграми класів.

22.2.2. Процес виготовлення піци, це комплексний процес, що складається з послідовності подібних кроків. Розглядається процес виготовлення двох видів Піци - Вегетаріанська піца та Піца з шинкою.

Проведіть рефакторинг коду, застосувавши до цього коду шаблон проектування Шаблонний метод.

22.3. Завдання для самостійної роботи

22.3.1. Проведіть рефакторинг коду, застосувавши до цього коду шаблон проектування Шаблонний метод.

23. Відвідувач.

23.1. Контрольні запитання

23.1.1. Для чого застосовується шаблон проектування Відвідувач?

23.1.2. Наведіть основні етапи реалізації шаблону Відвідувач.

23.1.3. Наведіть основні переваги та недоліки застосування шаблону Відвідувач.

23.2. Завдання для аудиторної роботи

23.2.1. Розглядається набір товарів (хліб, молоко, тощо), кожен з яких має ціну для продажу. Крім цього розглядається клас, що є списком товарів та має метод, що повертає загальну вартість усіх товарів списку.

Реалізуйте шаблон проектування Відвідувач, додавши можливість змінювати ціну товарів у такі способи:

- встановлювати знижку;
- підвищувати ціну на товар

для кожного типу товарів окремо.

Після цього додайте в симуляцію новий товар, наприклад, каву та поширте підтримку реалізованого шаблону Відвідувач на цей новий товар.

23.3. Завдання для самостійної роботи

23.3.1. Розглядається підприємство на якому працюють різні типи співробітників (продавець, менеджер, тощо), кожен з яких отримує заробітну плату. Також є клас, що об'єднує усіх співробітників організації та повідомляє загальну суму видатків, що має заплатити організація як заробітну плату.

Реалізуйте шаблон проектування Відвідувач, додавши можливість змінювати заробітну плату для співробітників у такі способи:

- підвищувати заробітну плату на заданий відсоток;
- накладати штрафи на співробітників.

Після цього додайте в симуляцію новий тип співробітників, наприклад, працівника IT-підтримки та поширте підтримку реалізованого шаблону Відвідувач на нього.

23.3.2. Розглянемо абстрактний клас Військовий об'єкт. Військовий об'єкт можуть відвідувати шпигуни, які можуть нанести йому шкоду або викрасти секретні папери.

Розглянемо клас «Генеральний штаб», який є нащадком класу Військовий об'єкт та у якому зосереджений старший офіцерський склад та зберігаються секретні папери. Також, розглядається клас Військова база, на якій зосереджені особовий склад та техніка.

Реалізуйте шаблон програмування Відвідувач (Visitor) для моделювання процесу відвідування військових об'єктів шпигунами. Для цього опишіть абстрактний клас Шпигун, що містить абстрактні методи для відвідування різних типів військових об'єктів. Розгляньте два типи шпигунів – Секретний агент та Диверсант – нащадки класу Шпигун. Метою відвідування військового об'єкту класу Секретний агент є збір або викрадення секретної інформації. Метою відвідування військового об'єкту шпигуном Диверсант є знищення секретної документації, знищення особового складу та техніки військового об'єкту.

Список літератури та додаткових джерел

1. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides., 1995.
2. Рефакторинг.Гуру [Електронний ресурс] – Режим доступу до ресурсу: <https://refactoring.guru/uk>.
3. Head First. Патерни проектування. Легкий для сприйняття довідник / Е.Фрімен, Е. Робсон, К. Сьєрра, Б. Бейтс., 2020. – 672 с.
4. Методичні вказівки до лабораторних робіт з дисципліни "Шаблони проектування програмного забезпечення" [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/krenevych/design-patterns-java>.
5. Крєневич А. П. Python у прикладах і задачах Частина 1. Структурне програмування [Електронний ресурс] / Андрій Павлович Крєневич. – 2017. – Режим доступу до ресурсу: <http://www.matfiz.univ.kiev.ua/books..>
6. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с.
7. Крєневич, А.П. С у задачах і прикладах : навчальний посібник із дисципліни "Інформатика та програмування" / А.П. Крєневич, О.В. Обвінцев. – К. : Видавничо-поліграфічний центр "Київський університет", 2011. – 208 с.
8. Обвінцев О.В. Інформатика та програмування. Курс на основі Python. Матеріали лекцій. – К., Основа, 2017
9. Навчальні матеріали: Python [Електронний ресурс] – Режим доступу до ресурсу: <http://www.matfiz.univ.kiev.ua/pages/13>.