

МIНIСТЕРСТВО  ОСВIТИ  І  НАУКИ  УКРАЇНИ

НАЦІОНАЛЬНИЙ   ТЕХНІЧНИЙ   УНІВЕРСИТЕТ   УКРАЇНИ

“КИЇВСЬКИЙ  ПОЛІТЕХНІЧНИЙ  ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Курсова робота**

з дисципліни “ Об’єктно-орієнтоване програмування ”

тема

“Шаблони проектування в ООП. Програма холдингова компанія по будівництву.”

Виконав

студент II курсу

групи КП- 82

Мельничук Олексій Геннадійович

залікова книжка КП-8215

Керівник роботи

доц., к.т.н. Заболотня Т.М.

Оцінка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, підпис)

КИЇВ 2020

**ЗМІСТ**

**ВСТУП**.........................................................................................................................3

1. **ОПИС СТРУКТУРНО-ЛОГІЧНОЇ ОРГАНІЗАЦІЇ ПРОГРАМИ**.........5
   1. Модульна організація програми............................................................5
   2. Функціональні характеристики.............................................................6
   3. Опис реалізованих класів.......................................................................7
2. **ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЕКТУВАННЯ**................................................................25
   1. Обґрунтування вибору та опис шаблонів проектування для реалізації програмного забезпечення автомату..................................25
   2. Діаграма класів......................................................................................32
   3. Результати роботи програми................................................................33

**ВИСНОВКИ**..............................................................................................................41

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**......................................................42

**ВСТУП**

Дана курсова робота присвячена розробці програмного забезпечення холдингової компанії по будівництву за допомогою використання шаблонів проектування. Задача керування холдингом є актуальною, оскільки в умовах сучасного ведення бізнесу потрібно мати можливість швидко та зручно керувати цінними паперами та дочірніми компаніями. Програма надає інтерфейси для керування акціями та проектами. Дана тема була вибрана з урахуванням можливості застосування усіх вивчених протягом курсу методів та принципів об’єктно-орієнтованого програмування для створення програмного забезпечення, зокрема мати використати шаблони проектування.

*Об’єктом* курсової роботи є процес керування акціями дочірніх компаній та керування будівельними проектами.

*Метою роботи* є розроблення програмного забезпечення холдингу по будівництву з використанням шаблонів проектування.

Для досягнення описаної мети необхідно виконати такі *завдання:*

* абстрагувати об’єкти предметної галузі;
* розробити структурну організацію програмного забезпечення за допомогою застосування основних принципів об’єктно-орієнтованого програмування та шаблонів проектування;
* визначити та описати функціональні характеристики програми;
* обґрунтувати вибір шаблонів проектування;
* розробити дизайн інтерфейсу користувача;
* виконати реалізацію програмного забезпечення відповідно до технічного завдання;
* виконати тестування розробленої програми;
* оформити документацію з курсової роботи.

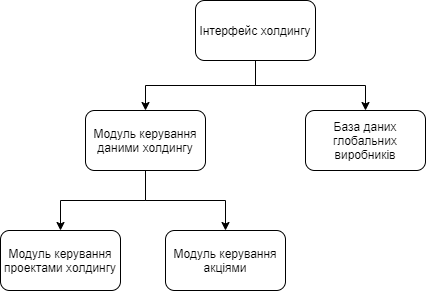
Розроблене програмне забезпечення складається з таких логічних частин: інтерфейсу холдингу, бази даних виробників, модуля керування даними холдингу, модуля керування проектами холдингу та модуля керування акціями.

Використані шаблони проектування:  **Observer, Chain of Responsibility, State, Strategy, Proxy (protection proxy), Decorator, Factory Method.**

Розроблене програмне забезпечення може бути використане у сфері менеджменту та фінансів з метою ефективнішого керування холдингом.

Пояснювальна записка складається зі вступу, двох розділів, загальних висновків та списку використаних джерел. Робота містить 60 рисунків. Загальний обсяг роботи – 38 друкованих сторінок.

1. **ОПИС СТРУКТУРНО-ЛОГІЧНОЇ СХЕМИ ПРОГРАМИ**
   1. **Модульна організація програми**



*Рис. 1.1.1. Модульна організація програми*

**Інтерфейс холдингу** – модуль, що відповідає за взаємодію з користувача з холдингом та списком усіх виробників.

**Модуль бази даних глобальних виробників** – модуль БД, що містить усіх відомих виробників у контексті програми та надає доступ до них.

**Модуль керування даними холдингу** – відповідає за взаємодію холдингу з інтерфейсом для користувача.

**Модуль керування проектами холдингу** – відповідає за керування проектами холдингу.

**Модуль керування акціями –** відповідає за керування акціями дочірніх компаній що належать холдингу.

* 1. **Функціональні характеристики**

При старті користувач вводить назву своєї холдингової компанії.

Після введення назви користувач потрапляє на головне меню, де він має бачить назву компанії та грошовий баланс. На головному меню користувач має змогу перейти до меню керування акціями та меню керування проектами, або ж завершити роботу.

В меню керування акціями відображаються усі куплені акції. Користувач має змогу перейти до меню покупки інших з каталогу акцій, або перейти до меню продажу акцій дочірніх компаній холдингу.

В меню покупки акцій відображаються усі промислові компанії з їх показниками доступних акцій для покупки, матеріал, що вони виробляють та ефективність їх роботи. Користувач має змогу вибрати одну по індексу та вказати процент акцій, який він бажає придбати.

В меню продажу акцій відображаються усі придбані акції. Користувач має змогу вибрати компанію та вибрати відсоток акцій, що він бажає продати.

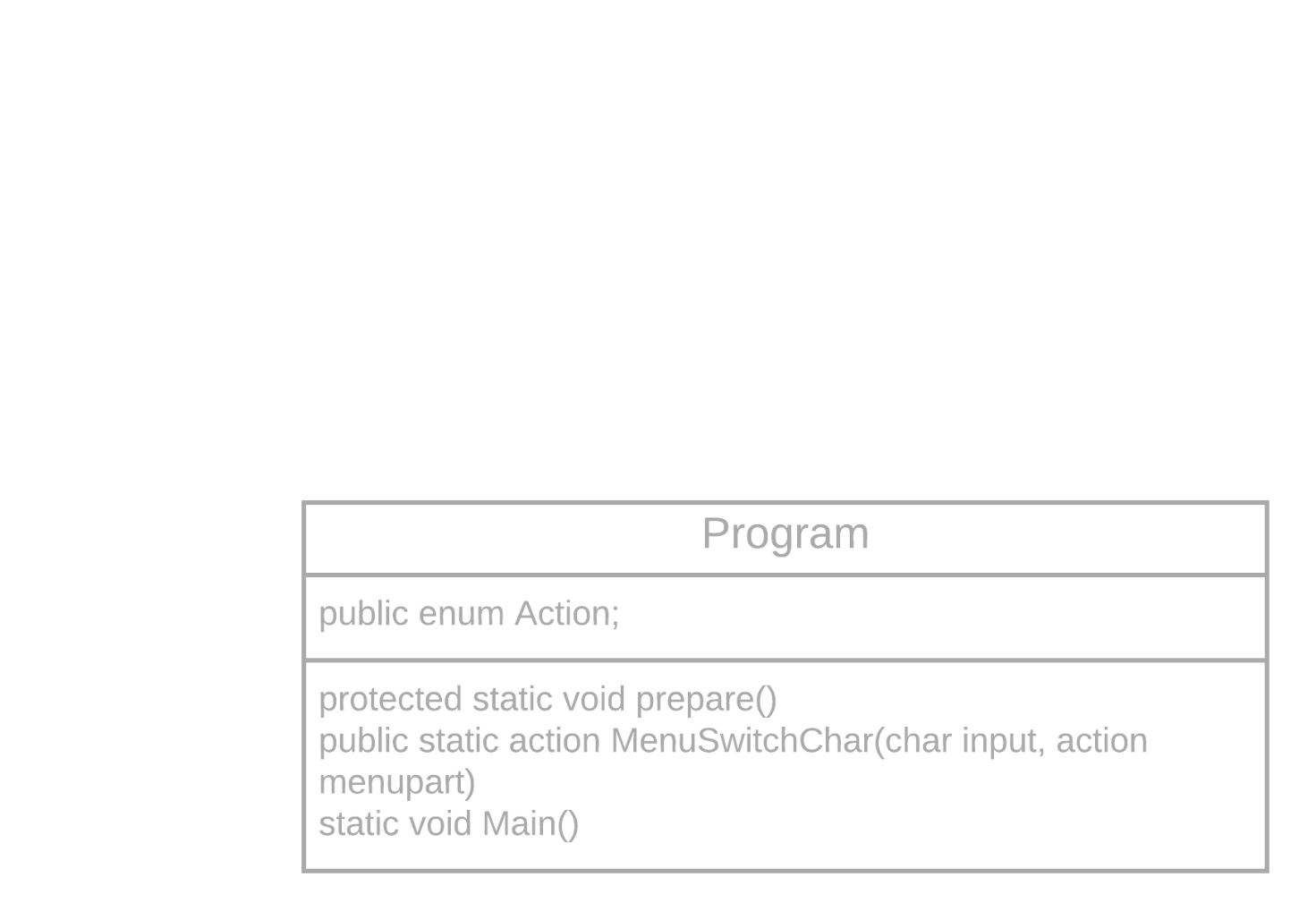
В меню керування проектами відображаються будівельні проекти холдингу. Користувач має змогу перейти до меню нового будівництва чи меню керування поточними.

В меню нового будівництва користувач вказує ім’я, тип будівництва (офіс, зона чи комплекс), технологію будівництва (мілкоштучна, монолітна чи каркасна) та географічне знаходження (край міста, спальний район та головний район).

В меню керування поточними проектами відображаються усі проекти з ресурсами, що потрібно зібрати, ціною та станом (не готовий, готовий до будівництва та готовий до продажу). Користувач має змогу дати команду розвитку проекту.

* 1. **Опис реалізованих класів**

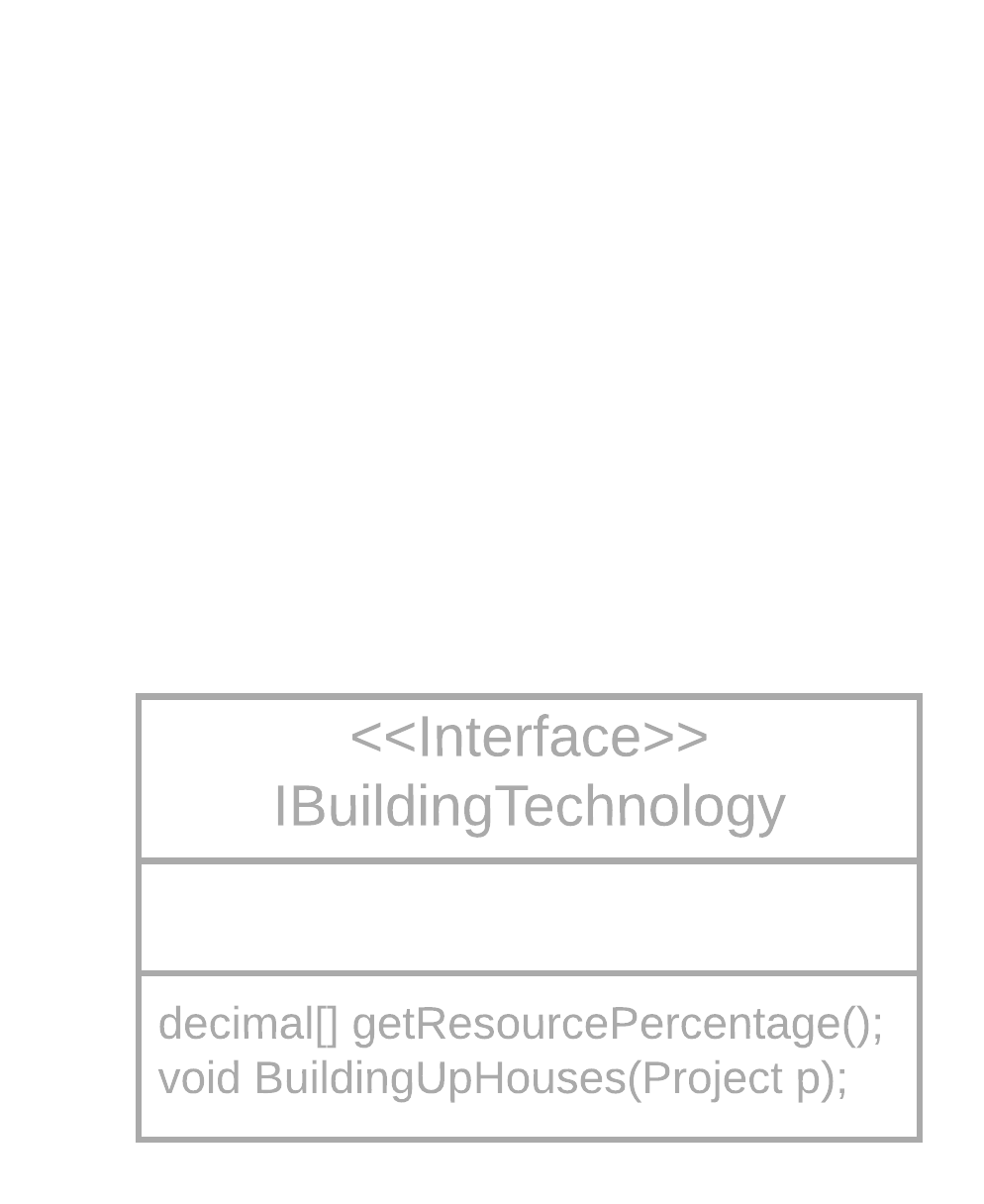
1. **Program**

*****Рис. 1.3.1. Program*

Основний клас додатку, завдяки якому відбувається робота з усіма модулями.

**Методи:**

1. Main – відповідає за роботу додатку;
2. Prepare – очистка консолі;
3. MenuSwitchChar – відповідає за переключення головних меню.
4. **IBuildingTechnology**

*****Рис. 1.3.2. Клас IBuildingTechnology*

Інтерфейс, що реалізує стратегію планування ресурсів проекту та його будівництва.

**Методи:**

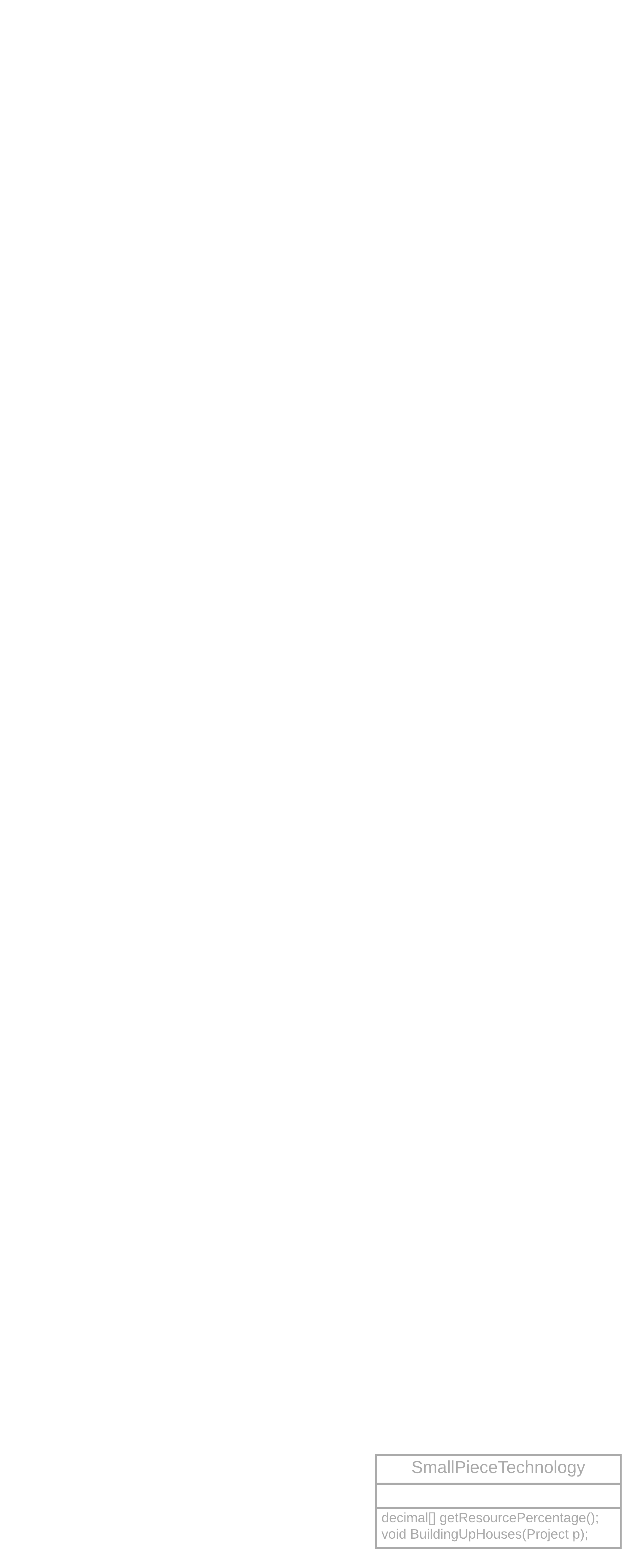
1. getResourcePercentage – повертає масив з відсотками ресурсів;
2. BuildingUpHouses – приймає проект та надсилає запити за ресурсами до дочірніх компаній холдингу.
3. **CastInPlaceTechnology**

*****Рис. 1.3.3. Клас CastInPlaceTechnology*

Клас стратегії монолітного будівництва, що реалізовує інтерфейс IBuildingTechnology.

**Методи:**

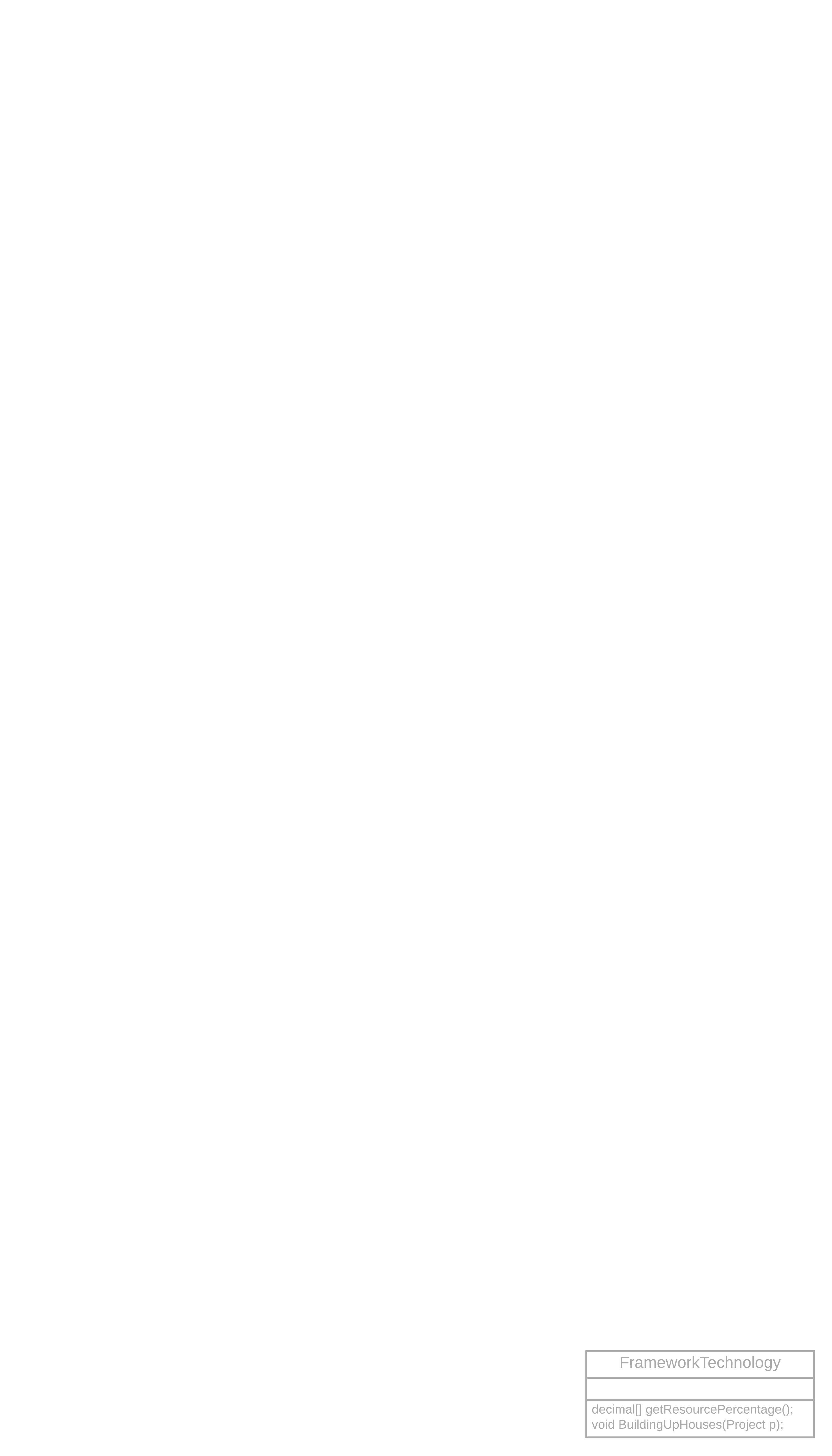
1. getResourcePercentage – повертає масив з відсотками ресурсів;
2. BuildingUpHouses – приймає проект та надсилає запити за ресурсами до дочірніх компаній холдингу.
3. **SmallPieceTechnology**

*****Рис. 1.3.4. Клас SmallPieceTechnology*

Клас стратегії мілкоштучного будівництва, що реалізовує інтерфейс IBuildingTechnology.

**Методи:**

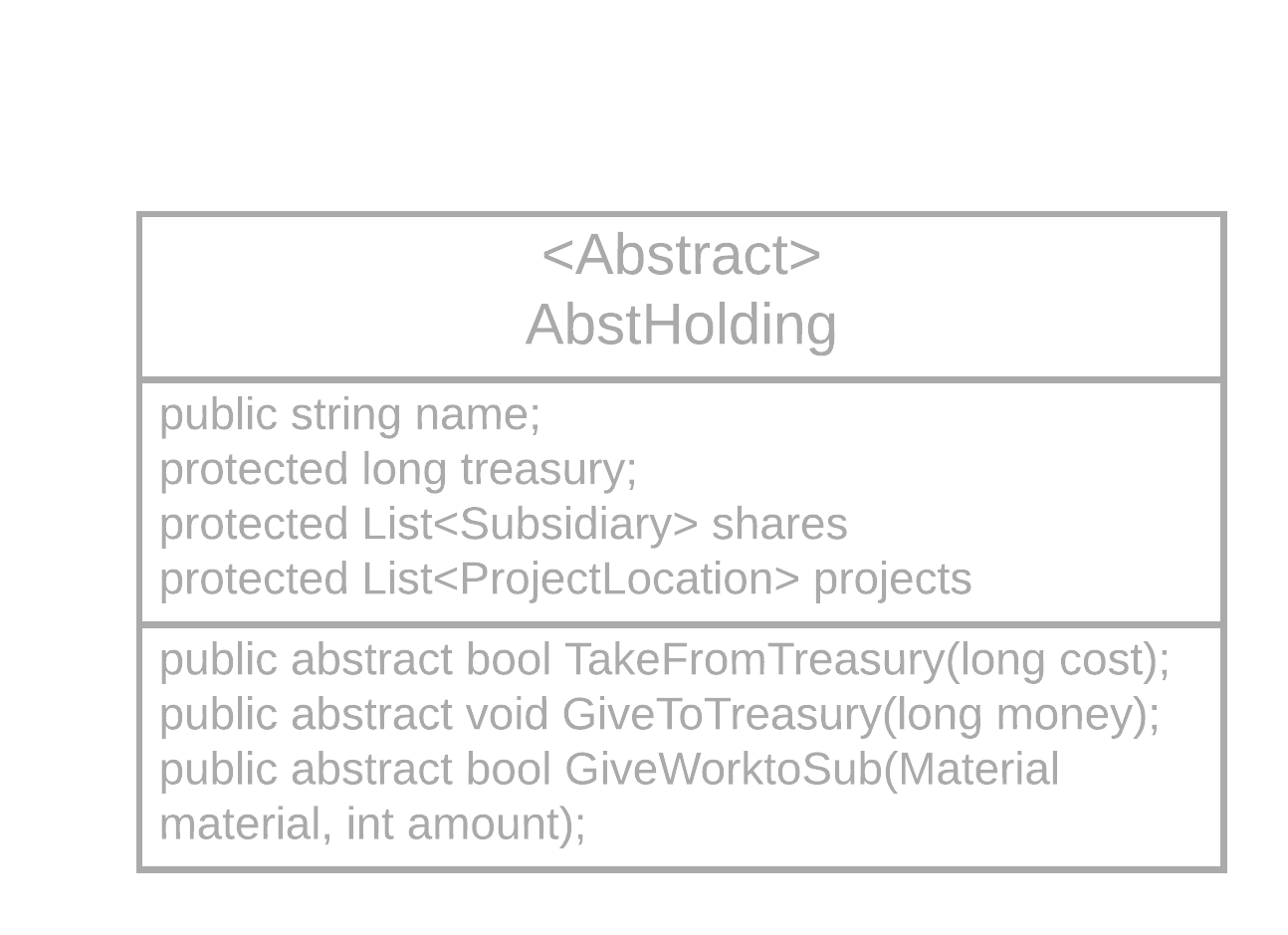
1. getResourcePercentage – повертає масив з відсотками ресурсів;
2. BuildingUpHouses – приймає проект та надсилає запити за ресурсами до дочірніх компаній холдингу.
3. **FrameworkTechnology**

*****Рис. 1.3.5. Клас FrameworkTechnology*

Клас стратегії каркасного будівництва, що реалізовує інтерфейс IBuildingTechnology.

**Методи:**

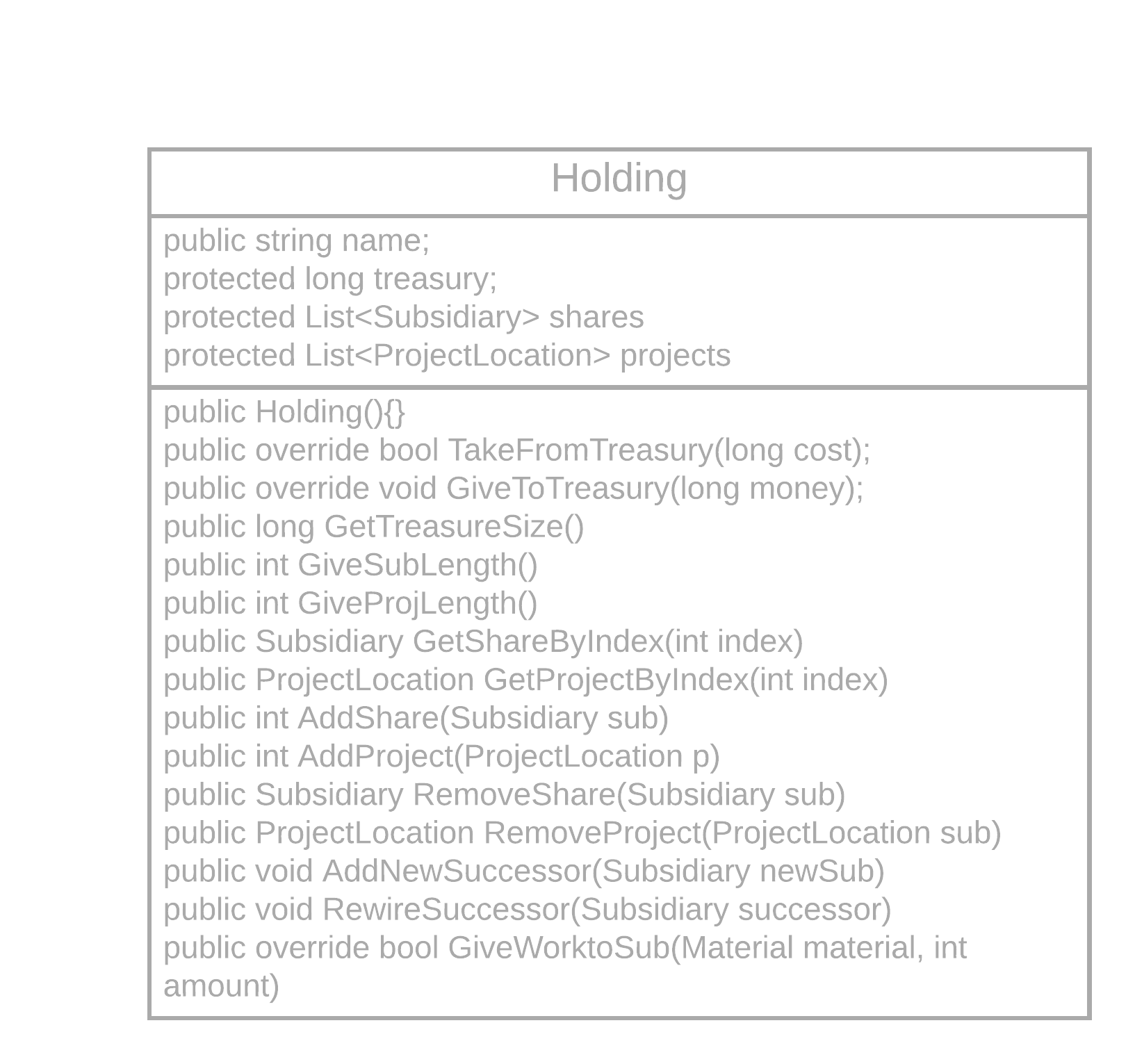
1. getResourcePercentage – повертає масив з відсотками ресурсів;
2. BuildingUpHouses – приймає проект та надсилає запити за ресурсами до дочірніх компаній холдингу.
3. **AbstHolding**

*****Рис. 1.3.6. Клас AbstHolding*

Абстрактний клас холдингу, що встановлює основні методи та поля.

**Методи:**

1. TakeFromTreasury – вираховування коштів з балансу холдингу.
2. GiveToTreasury – нарахування коштів на баланс холдингу.
3. **Holding**

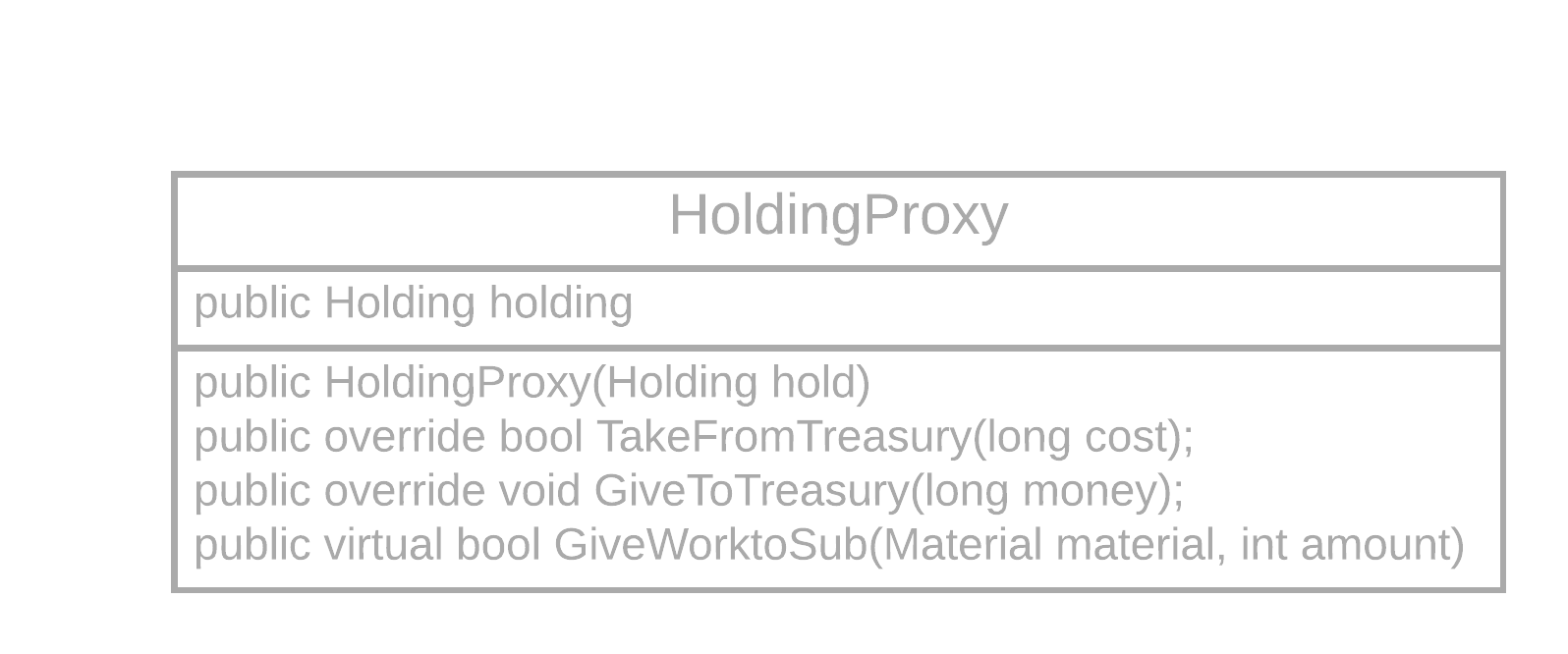


*Рис. 1.3.7. Клас Holding*

Основний клас холдингу, реалізовує методи AbstHolding та нові методи.

**Методи:**

1. TakeFromTreasury – вираховування коштів з балансу холдингу.
2. GiveToTreasury – нарахування коштів на баланс холдингу.
3. GetTreasureSize – повертає баланс холдингу.
4. GiveSubLength – повертає кількість акціонерів.
5. GiveProjLength – повертає кількість поточних проектів.
6. GetShareByIndex – повертає акцію по індексу.
7. GetProjectByIndex – повертає проект по індексу.
8. AddShare – додає нову дочірню компанію до холдингу.
9. AddProject – додає новий проект до холдингу.
10. RemoveShare – прибирає дочірню компанію з холдингу.
11. RemoveProject – прибирає проект з холдингу.
12. AddNewSuccessor – надає наступника компанії.
13. RewireSuccessor – налагоджує зв’язки компаній після видалення.
14. GiveWorktoSub – дає роботу компанії.
15. **HoldingProxy**

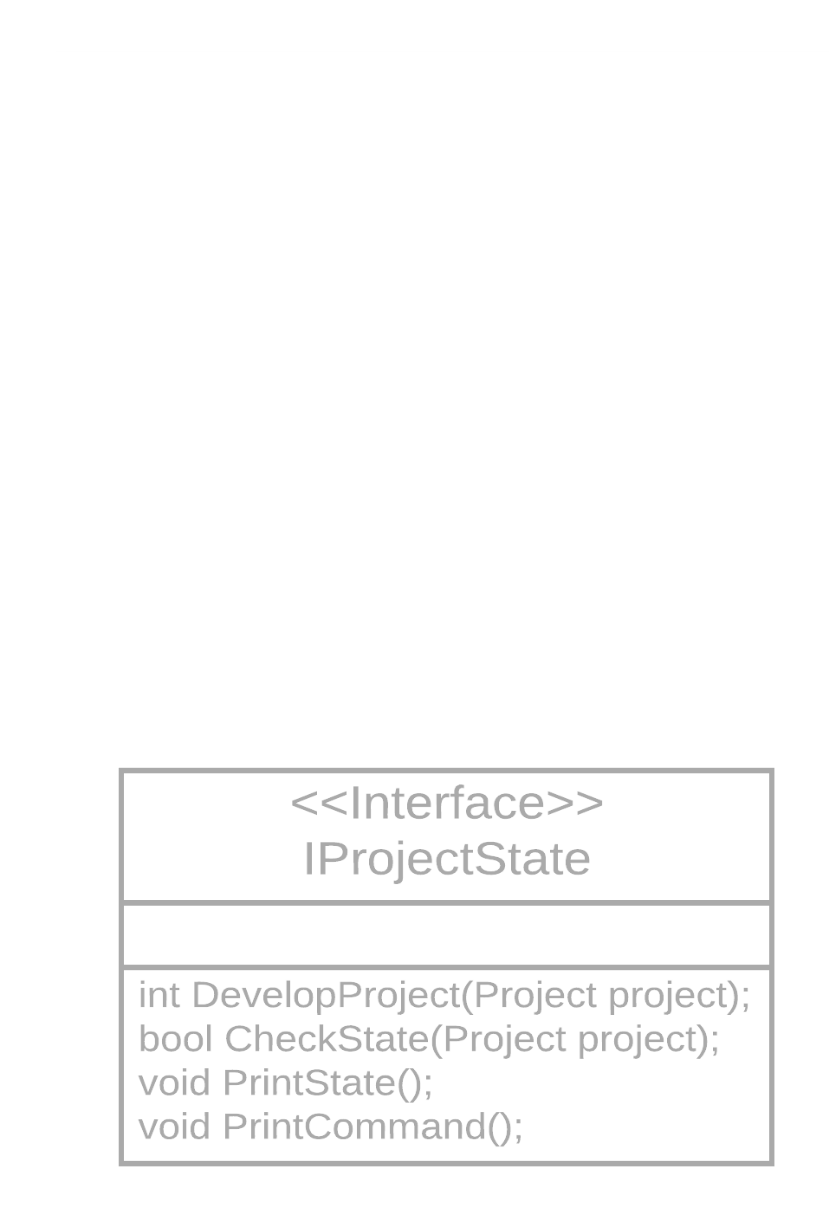


*Рис. 1.3.8. Клас HoldingProxy*

Клас-заступник холдингу, що регулює зняття коштів з балансу.

**Методи:**

1. TakeFromTreasury – вираховування коштів з балансу холдингу.
2. GiveToTreasury – нарахування коштів на баланс холдингу.
3. **IProjectState**

****

*Рис. 1.3.9. Клас IProjectState*

Інтерфейс стану проекту, що визначає основні методи.

**Методи:**

1. DevelopProject – команда розвитку проекту.
2. CheckState – перевірка та зміна стану проекту в разі задоволення потреб.
3. PrintState – друк стану.
4. PrintCommand – друк опису проекту по стану.
5. **NotReadyState**



*Рис. 1.3.10. Клас NotReadyState*

Клас стану неготового проекту, що реалізує інтерфейс IProjectState.

**Методи:**

1. DevelopProject – команда розвитку проекту.
2. CheckState – перевірка та зміна стану проекту в разі задоволення потреб.
3. PrintState – друк стану.
4. PrintCommand – друк опису проекту по стану.
5. **ReadyForBuildingState**

****

*Рис. 1.3.11. Клас ReadyForBuildingState*

Клас стану готового до будівництва проекту, що реалізує інтерфейс IProjectState.

**Методи:**

1. DevelopProject – команда розвитку проекту.
2. CheckState – перевірка та зміна стану проекту в разі задоволення потреб.
3. PrintState – друк стану.
4. PrintCommand – друк опису проекту по стану.
5. **ReadyForSaleState**

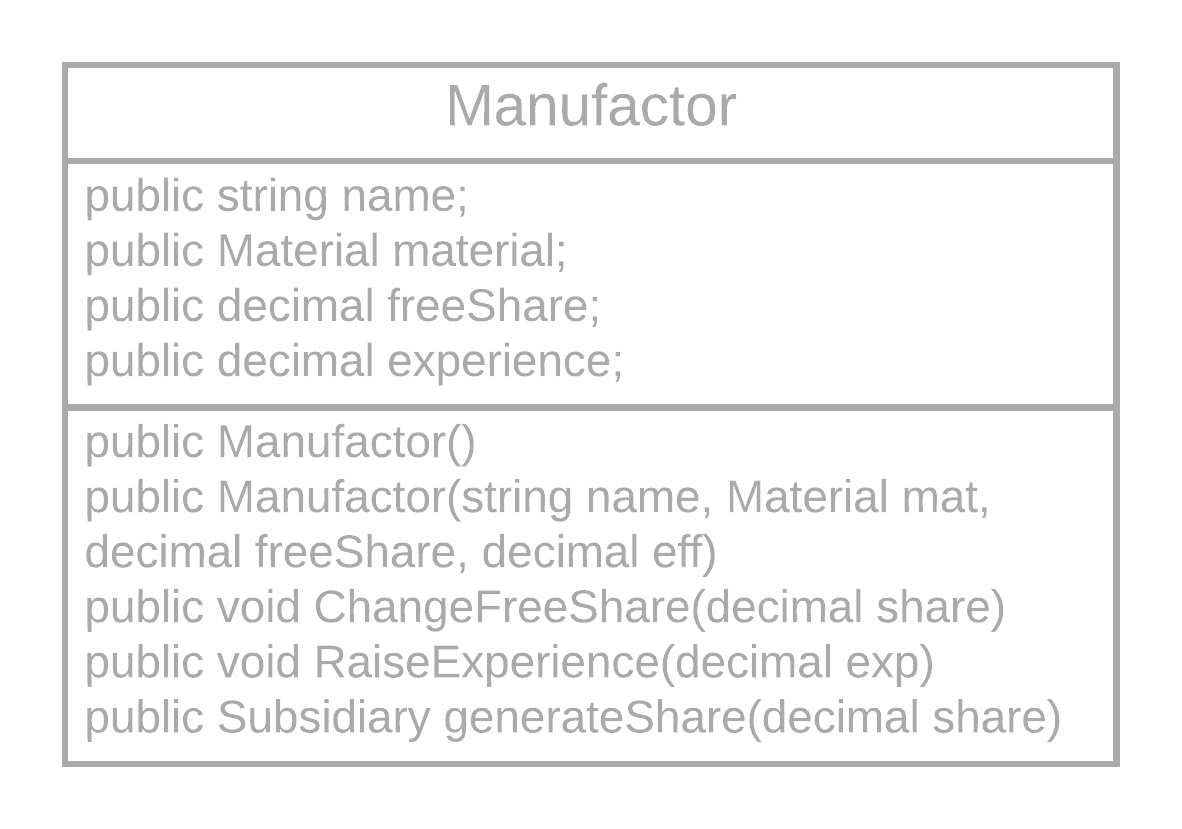
****

*Рис. 1.3.12. Клас ReadyForSaleState*

Клас стану готового до продажу проекту, що реалізує інтерфейс IProjectState.

**Методи:**

1. DevelopProject – команда розвитку проекту.
2. CheckState – перевірка та зміна стану проекту в разі задоволення потреб.
3. PrintState – друк стану.
4. PrintCommand – друк опису проекту по стану.
5. **Manufactor**

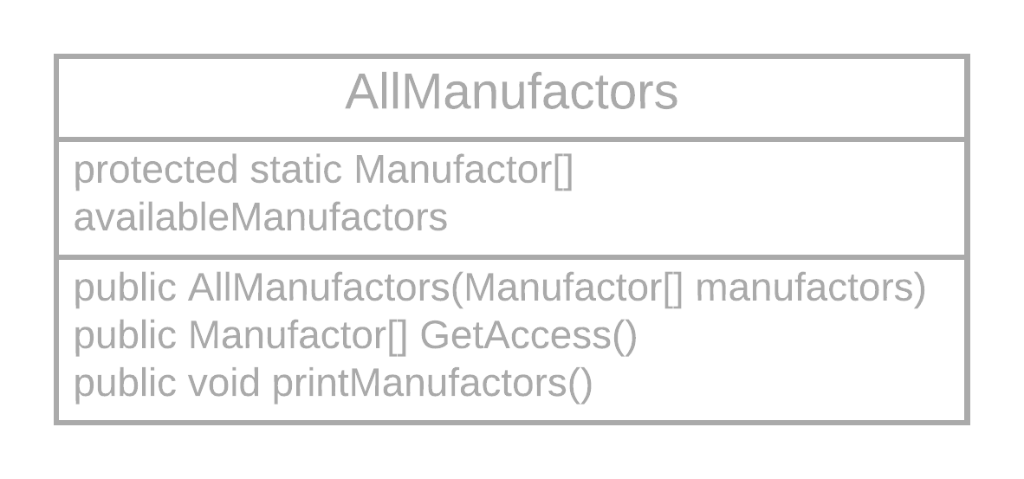
****

*Рис. 1.3.13. Клас Manufactor*

Клас виробника.

**Методи:**

1. generateShare – генерує акцію виробника;
2. RaiseExperience – піднімає ефективність виробництва:
3. ChangeFreeShare – змінює доступну вільну частку акцій.
4. **AllManufactors**



*Рис. 1.3.14. Клас AllManufactors*

Клас усіх виробників.

**Методи:**

1. getAccess – надає доступ до виробників;
2. printManufactors – друкує усіх виробників.
3. **MaterialsPrice**

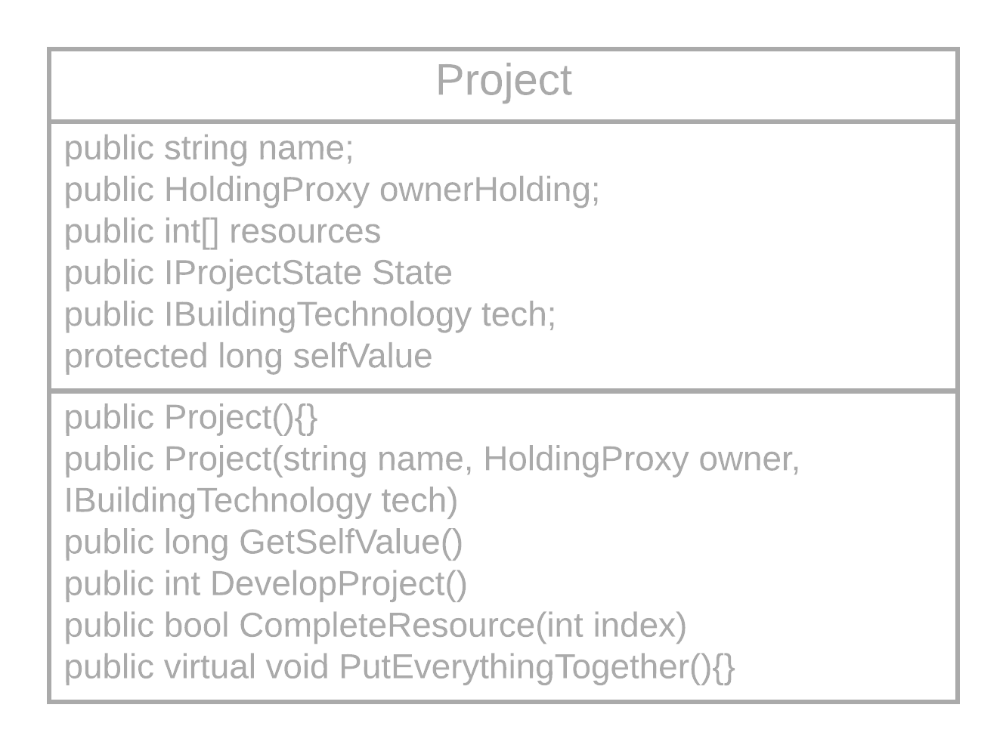


*Рис. 1.3.15. Клас MaterialsPrice*

Клас цін матеріалів.

**Методи:**

1. getPrice – повертає ціну матеріалу.
2. **Project**

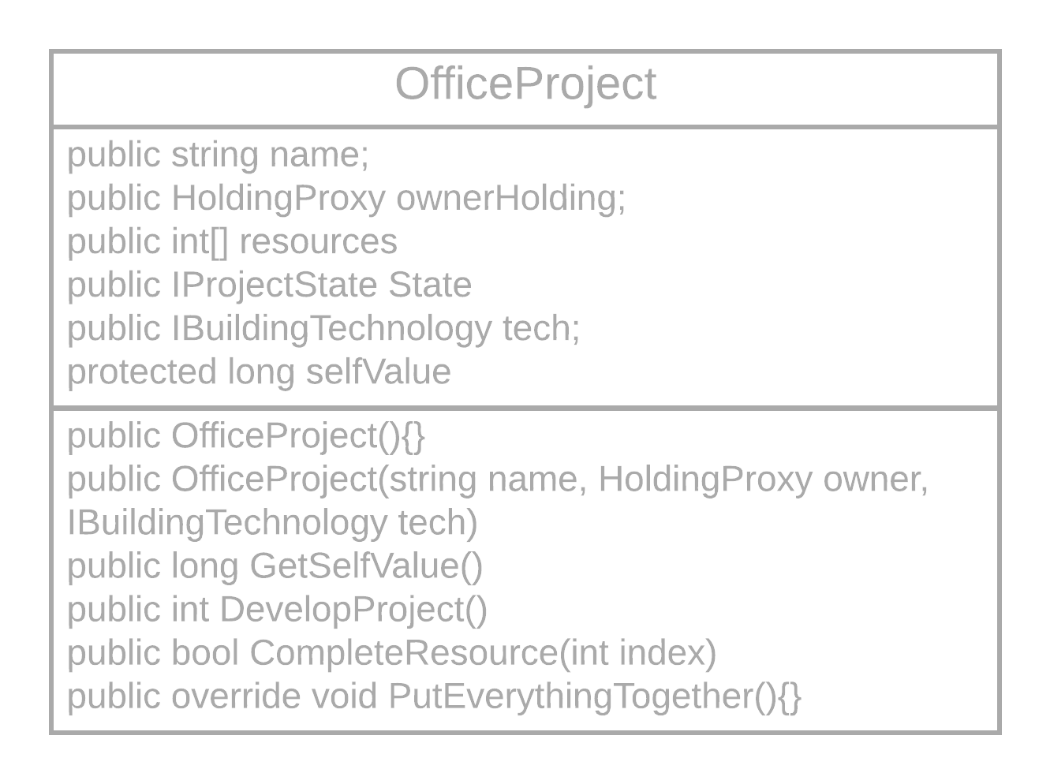


*Рис. 1.3.16. Клас Project*

Клас проекту забудови.

**Методи:**

1. getSelfValue – повертає собівартість проекту;
2. AddToValue – додає ціну до собівартості;
3. DevelopProject – розвиває проект;
4. AssignResource – назначає кількість потрібного ресурсу при плануванні;
5. CompleteResource – анулює кількість потрібних ресурсів;
6. PutEverythingTogether – будує проект.
7. **OfficeProject**

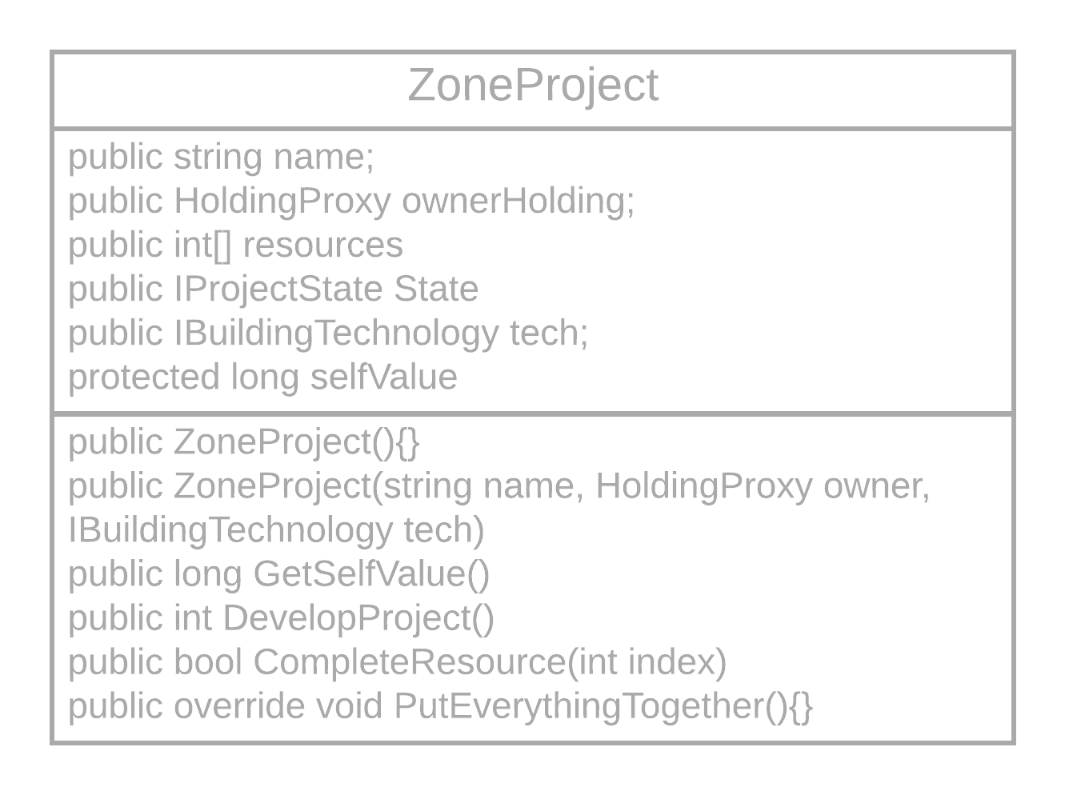


*Рис. 1.3.17. Клас OfficeProject*

Клас проекту офісної забудови.

**Методи:**

1. PutEverythingTogether – будує проект.
2. **ZoneProject**

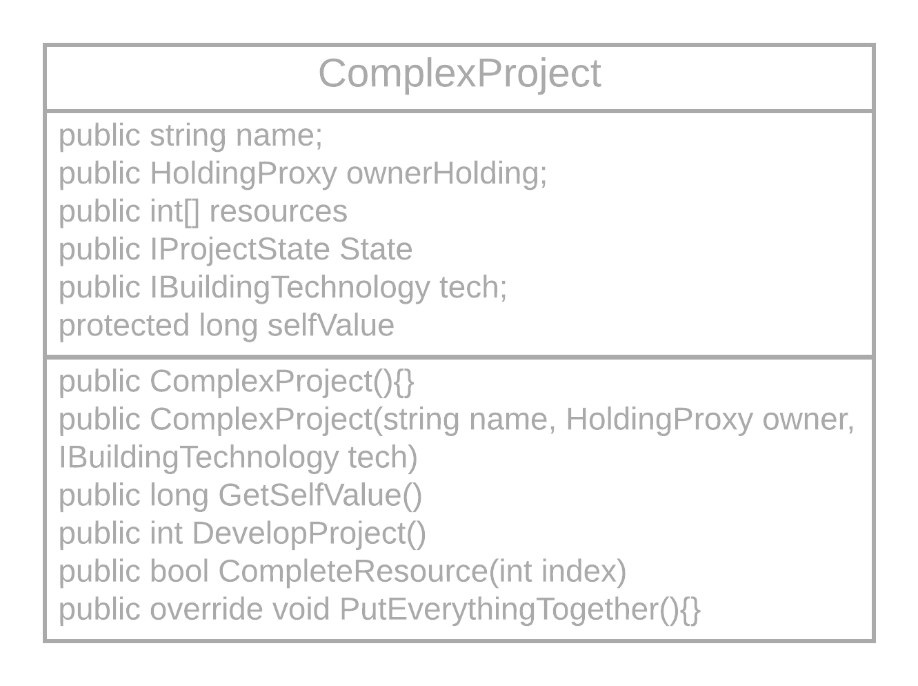
****

*Рис. 1.3.18. Клас ZoneProject*

Клас проекту забудови зони.

**Методи:**

1. PutEverythingTogether – будує проект.
2. **ComplexProject**

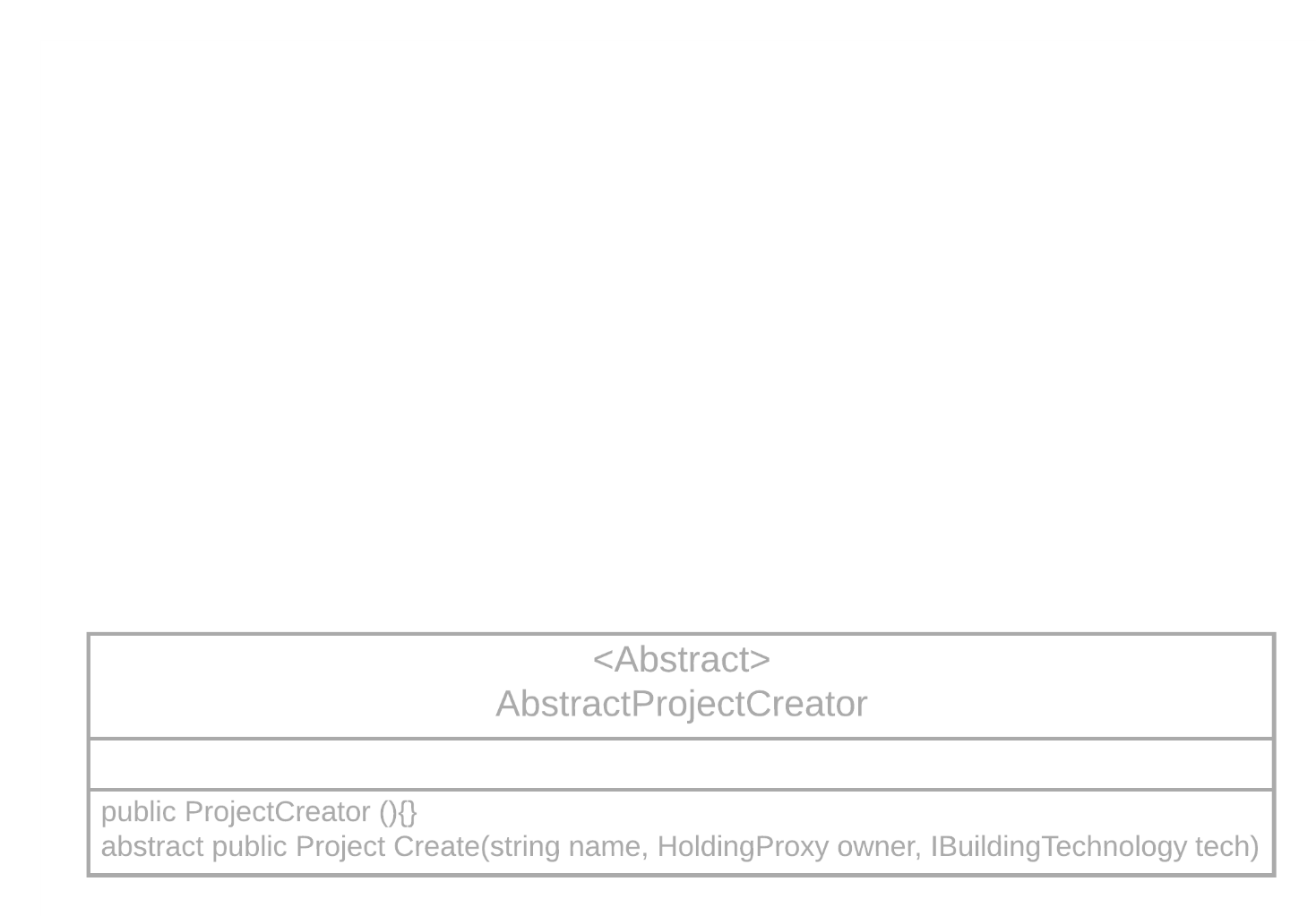
****

*Рис. 1.3.19. Клас ComplexProject*

Клас проекту забудови комплексу.

**Методи:**

1. PutEverythingTogether – будує проект.
2. **AbstractProjectCreator**

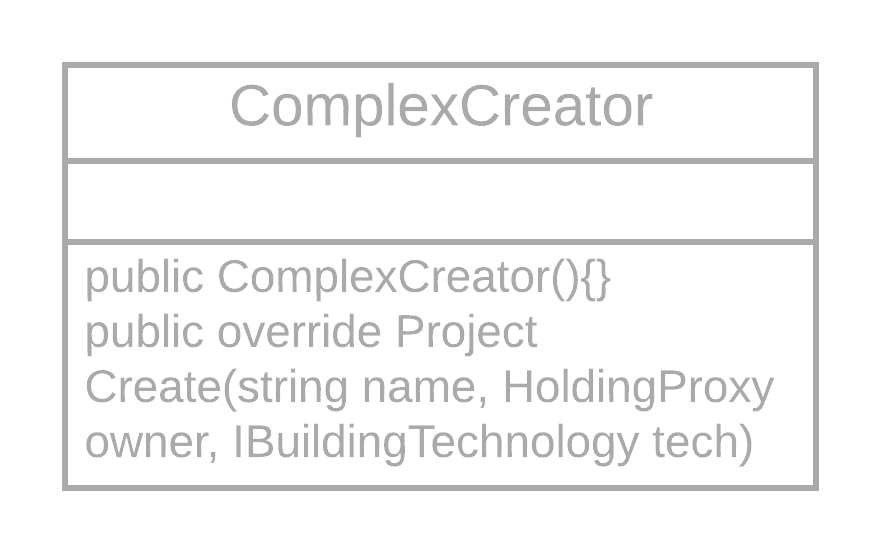
****

*Рис. 1.3.20. Клас AbstractProjectCreator*

Абстрактний клас фабрики проектів.

**Методи:**

1. Create – планує проект та його ціни.
2. **ComplexCreator**

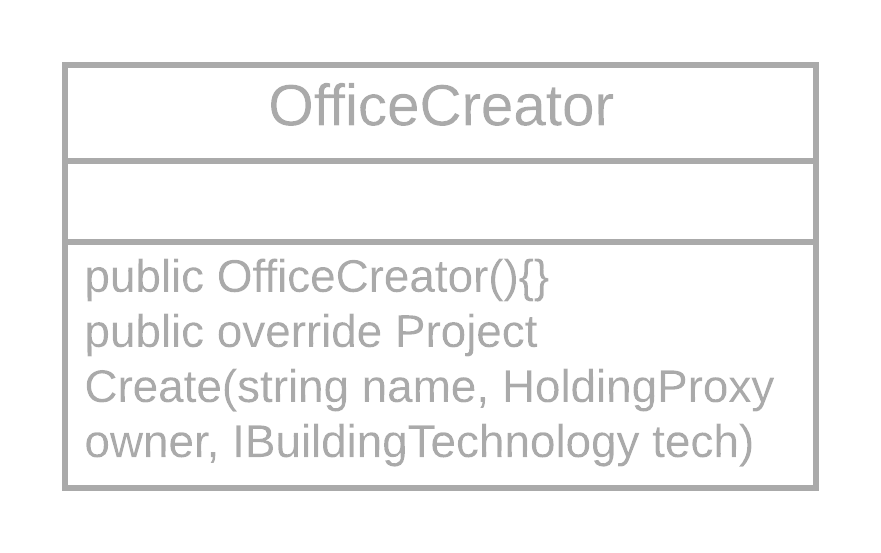
**

*Рис. 1.3.21. Клас ComplexCreator*

Клас фабрики проектів комплексів, що реалізує AbstractProjectCreator.

**Методи:**

1. Create – планує проект та його ціни.
2. **OfficeCreator**

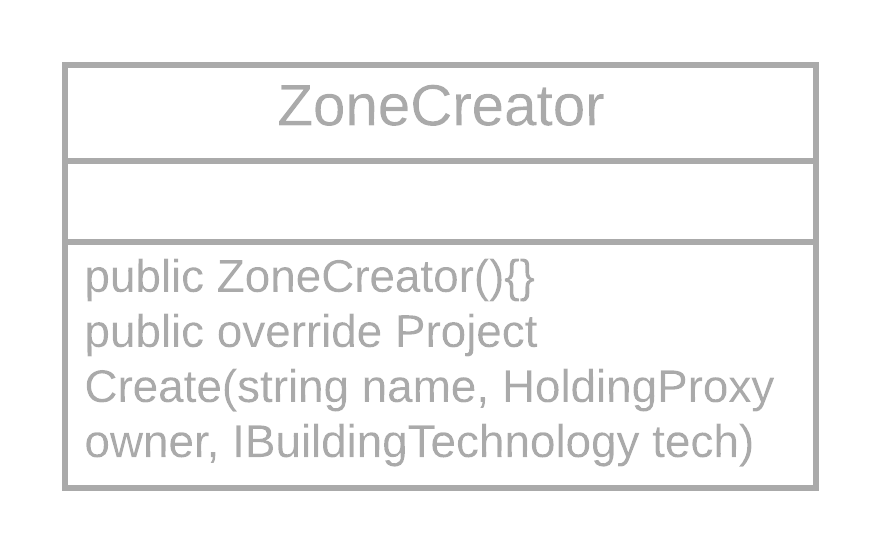
**

*Рис. 1.3.22. Клас OfficeCreator*

Клас фабрики проектів офісів, що реалізує AbstractProjectCreator.

**Методи:**

1. Create – планує проект та його ціни.
2. **ZoneCreator**

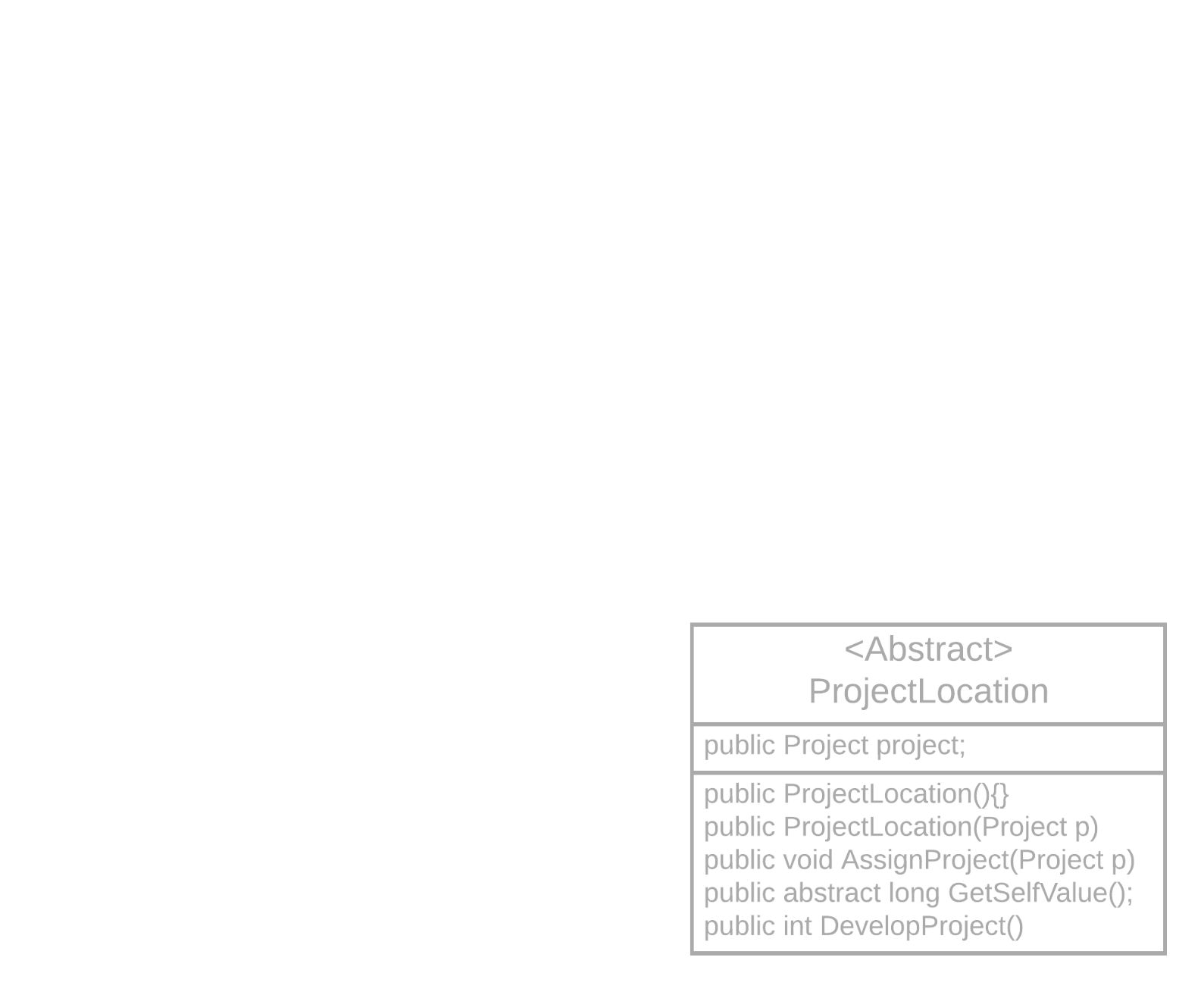
****

*Рис. 1.3.23. Клас ZoneCreator*

Клас фабрики проектів зон, що реалізує AbstractProjectCreator.

**Методи:**

1. Create – планує проект та його ціни.
2. **ProjectLocation**

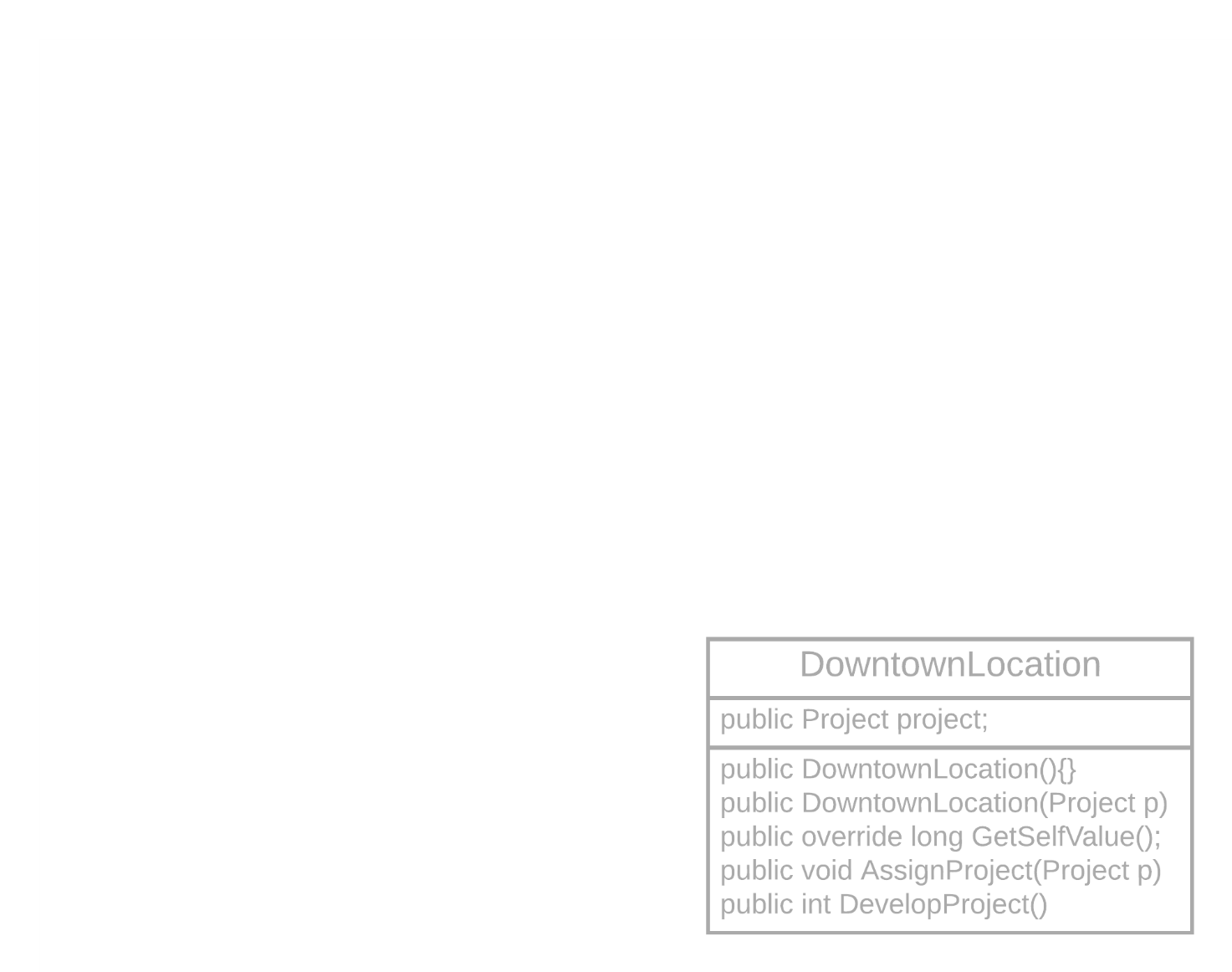


*Рис. 1.3.24. Клас ProjectLocation*

Абстрактний клас-декоратор локації проекту.

**Методи:**

1. AssignProject – ставить проект;
2. DevelopProject – дає команду розроблювати проект;
3. GetSelfValue – повертає змінену ціну проекту з урахуванням локації.
4. **DowntownLocation**



*Рис. 1.3.25. Клас DowntownLocation*

Клас-декоратор локації ділового району.

**Методи:**

1. AssignProject – ставить проект;
2. DevelopProject – дає команду розроблювати проект;
3. GetSelfValue – повертає змінену ціну проекту з урахуванням локації.
4. **CommuterLocation**

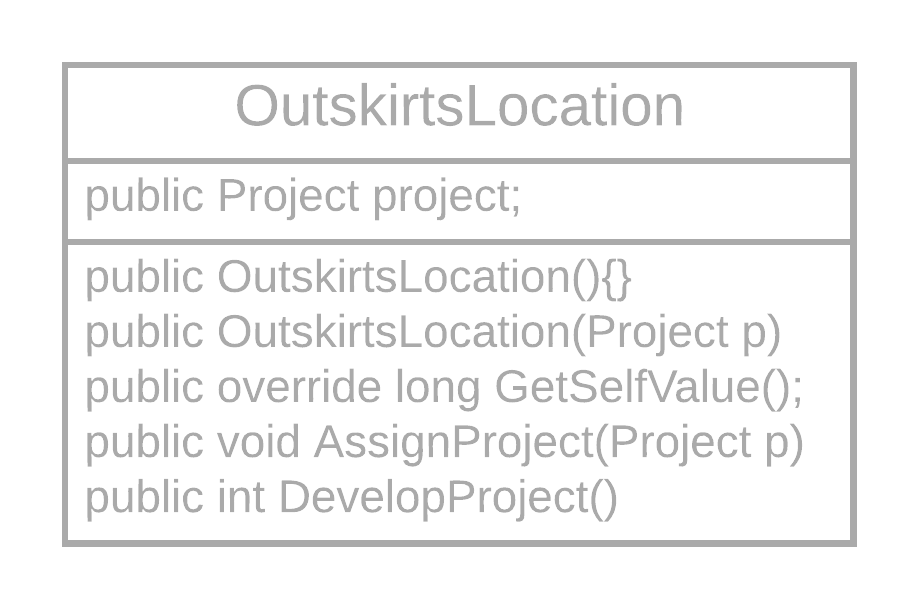
**

*Рис. 1.3.26. Клас CommuterLocation*

Клас-декоратор локації спального району.

**Методи:**

1. AssignProject – ставить проект;
2. DevelopProject – дає команду розроблювати проект;
3. GetSelfValue – повертає змінену ціну проекту з урахуванням локації.
4. **OutskirtsLocation**

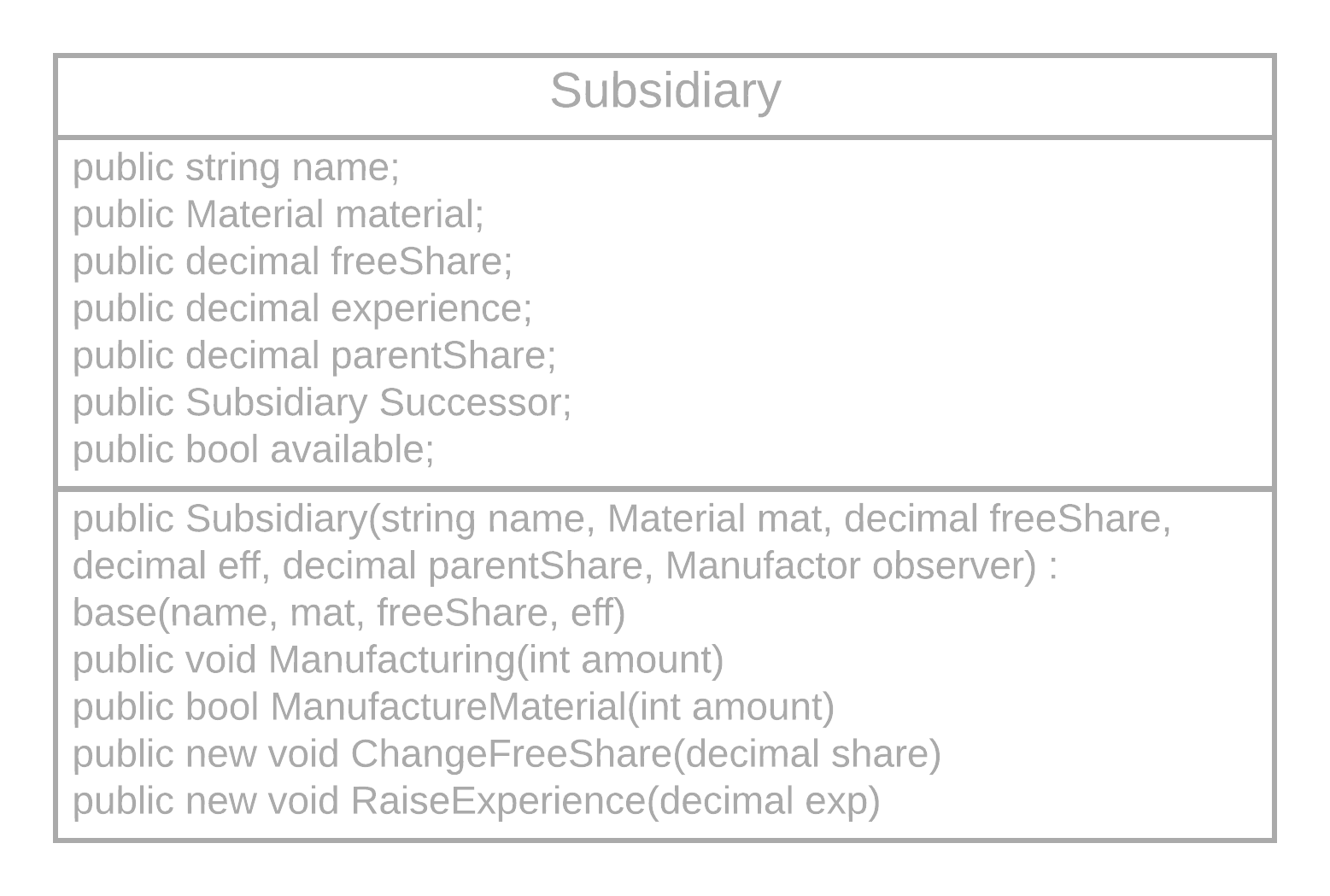


*Рис. 1.3.27. Клас OutskirtsLocation*

Клас-декоратор заміської локації.

**Методи:**

1. AssignProject – ставить проект;
2. DevelopProject – дає команду розроблювати проект;
3. GetSelfValue – повертає змінену ціну проекту з урахуванням локації.
4. **Subsidiary**

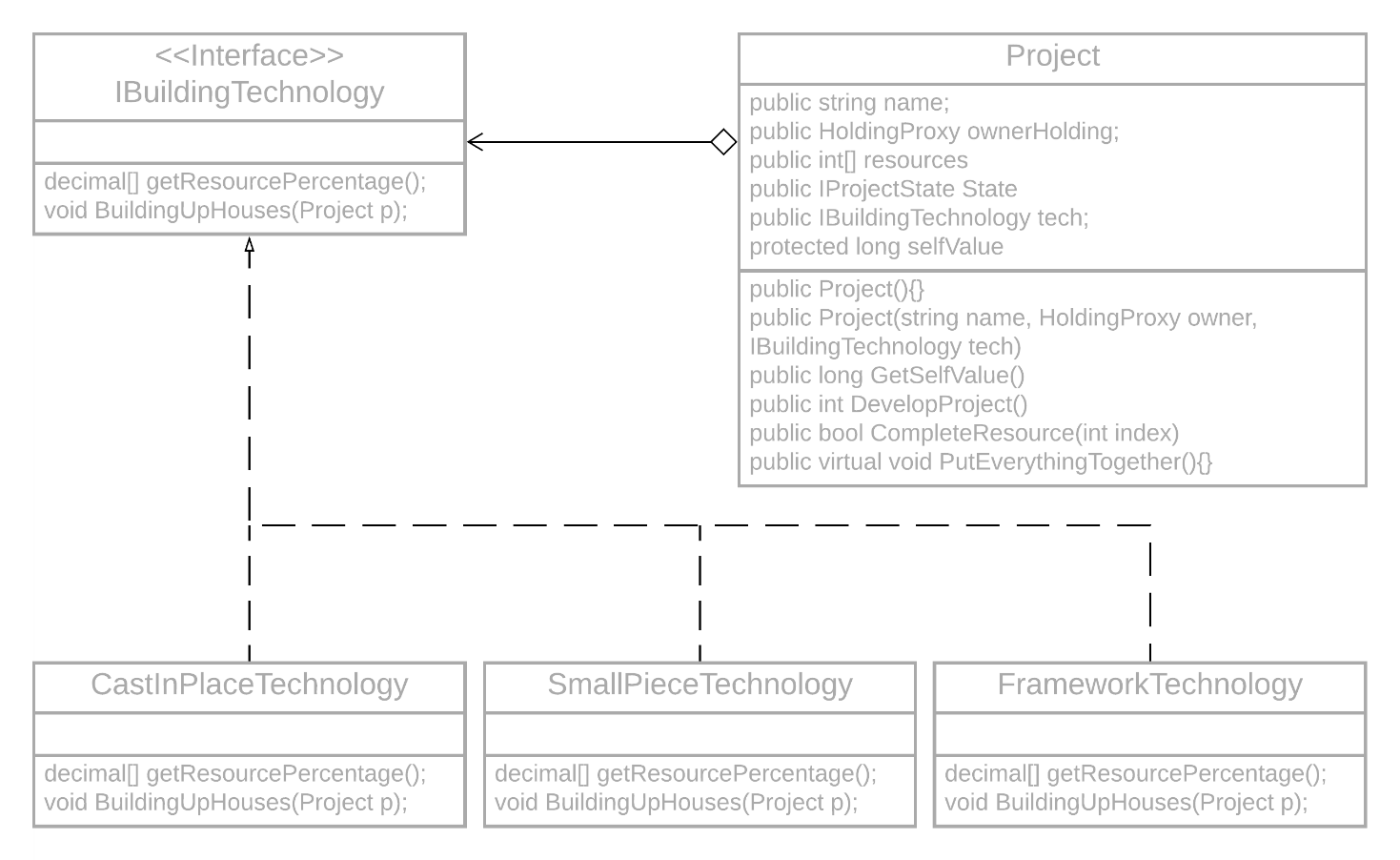


*Рис. 1.3.28. Клас Subsidiary*

Клас дочірньої компанії/акції.

**Методи:**

1. RaiseExperience – збільшує ефективність компанії;
2. Manufacturing – займає компанію роботою на проміжок часу;
3. ManufactureMaterial – вирішує чи доступний виробник для роботи і чи потрібно передати роботу наступнику;
4. ChangeFreeShare – змінює відсоток доступних акцій;
5. PrintSuccessor – друкує назву компанії-заступника.
6. **ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЕКТУВАННЯ**
   1. **Обґрунтування вибору та опис шаблонів проектування для реалізації програмного забезпечення автомату**
7. **Strategy**



*Рис. 2.1.1. UML-діаграма шаблону “****Стратегія****”*

*Поведінковий шаблон*, що визначає сімейство алгоритмів, інкапсулює кожен з них та робить їх взаємозамінними. Дозволяє змінювати алгоритми незалежно від коду клієнтів. Також відомий як *Policy*.

*Структура*.

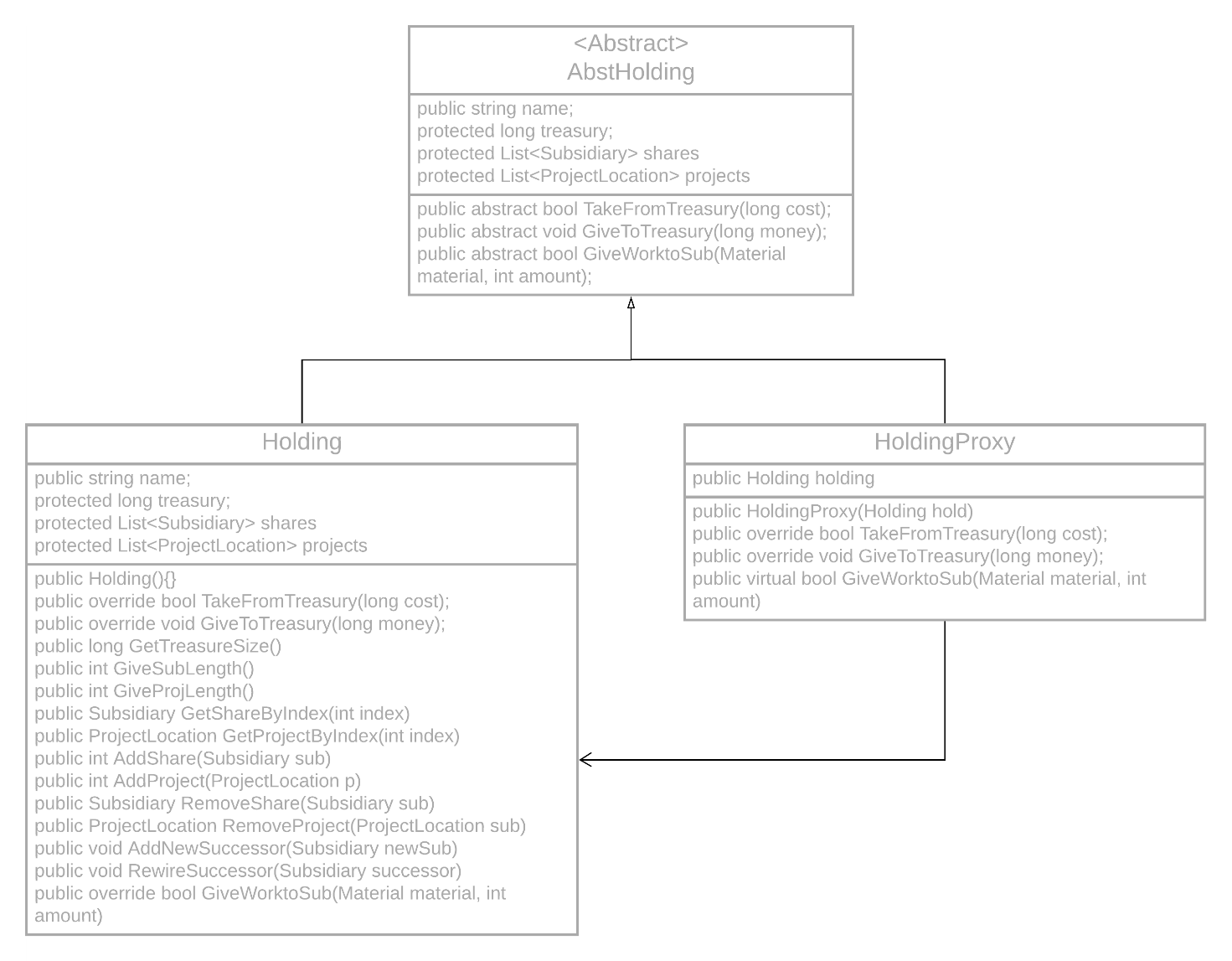
**IBuildingTechnology** – інтерфейс алгоритмів збору ресурсів;

**CastInPlaceTechnology, SmallPieceTechnology, FrameworkTechnology –** реалізація інтерфейсу **IBuildingTechnology;**

**Project –** містить посилання на об’єкт **ISorting** та викликає алгоритм у разі потреби.

*Обґрунтування використання даного шаблону*. Потреба у шаблоні виникла у процесі планування видів будівельного проекту, оскільки різні види потребують різні методи будівництва.

1. **Proxy (protection proxy)**



*Рис. 2.1.2. UML-діаграма шаблону “****Заступник-захисник****”*

*Структурний шаблон*, що забезпечує створення заступника об’єкта для контролю доступу до останнього через перехоплення всіх викликів.

*Структура*.

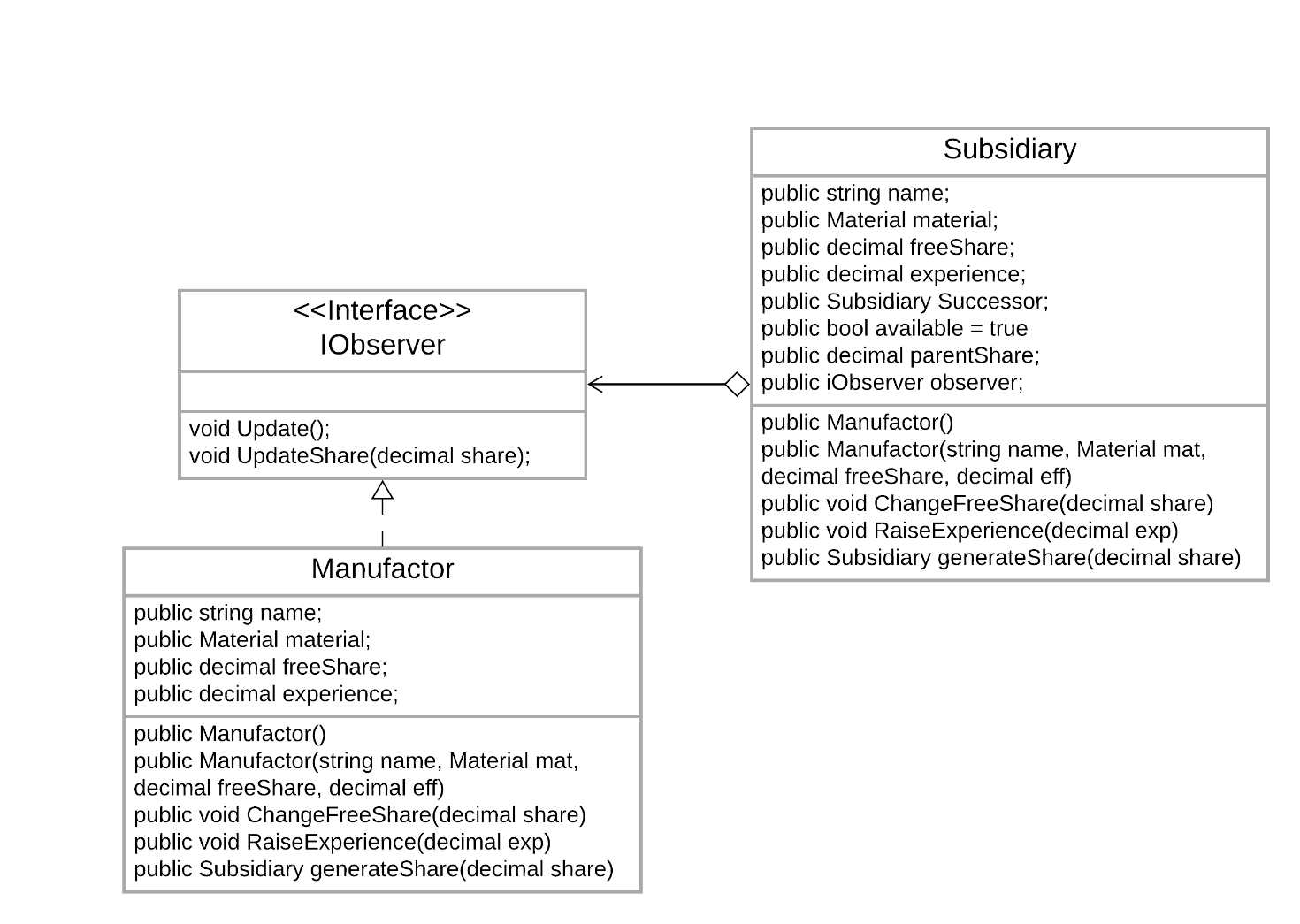
**AbstHolding –** абстрактний клас холдингової компанії **Holding.**

**Holding –** реалізація реального об’єкта холдингу.

**HoldingProxy –** заступник **Holding**, що зберігає посилання на реальний об’єкт та керує доступом до нього.

*Обґрунтування використання даного шаблону*.Шаблон був потрібний для контролювання доступу до грошового балансу **Holding**.

1. **Observer**



*Рис. 2.1.3. UML-діаграма шаблону “****Спостерігач****”*

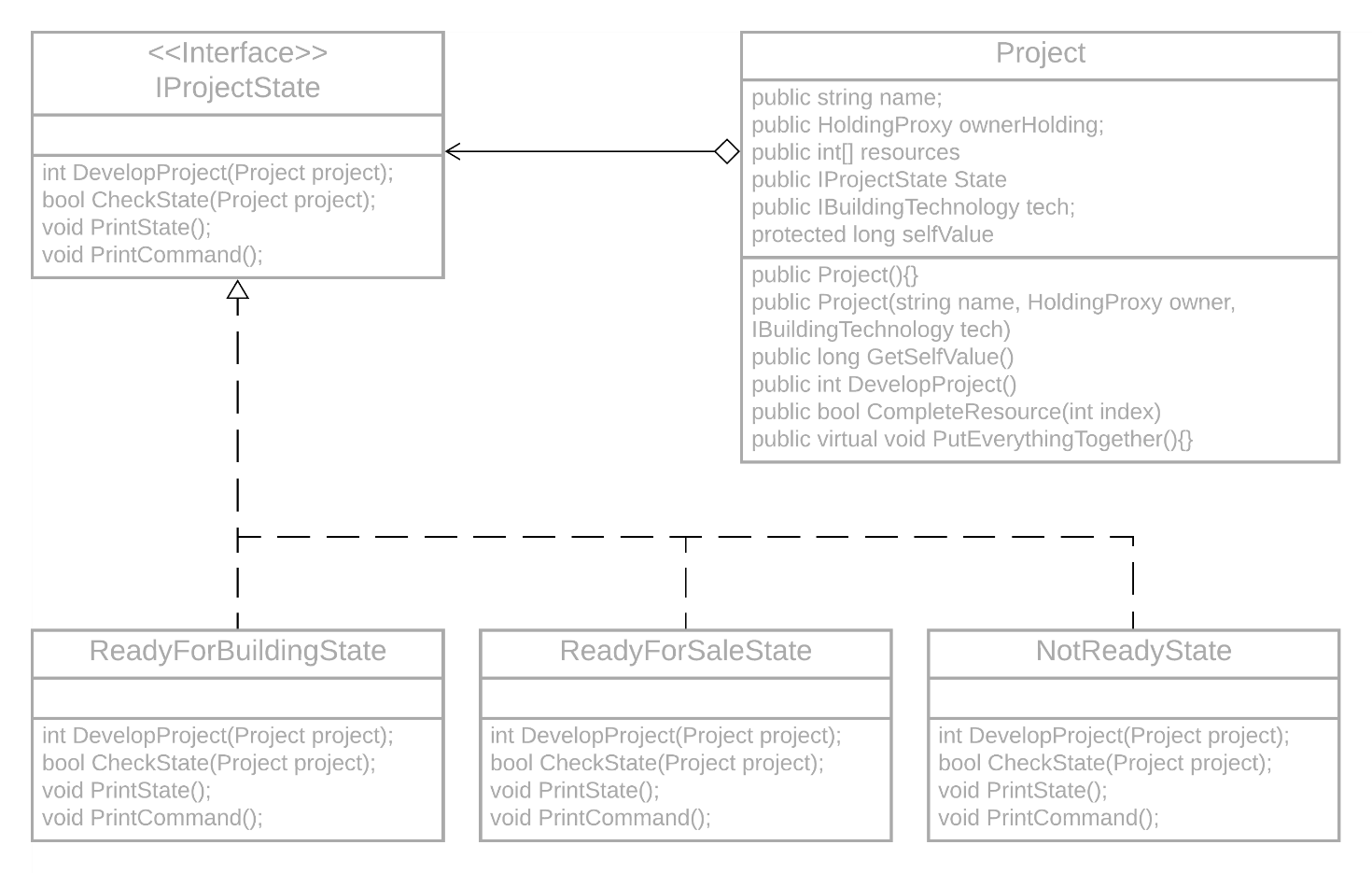
*Поведінковий шаблон*, що створює механізм підписки, за допомогою якого одні об'єкти можуть підписуватися на оновлення, що відбуваються в інших об'єктах.

*Структура*.

**Subsidiary –** клас дочірньої компанії.

**Manufactor –** клас цієї компанії, що належить до масиву усіх компаній.

*Обґрунтування використання даного шаблону*.При зміні значень у **Subsidiary** повинні також змінюватись значення у **Manufactor,** оскільки він представляє **Subsidiary** на загальному ринку і холдерам потрібно знати, скільки акцій вони ще можуть придбати.

1. **State**

*Рис. 2.1.4. UML-діаграма шаблону “****Стан****”*

*Поведінковий шаблон,* який дозволяє об'єкту варіювати свою поведінку залежно від внутрішнього стану.

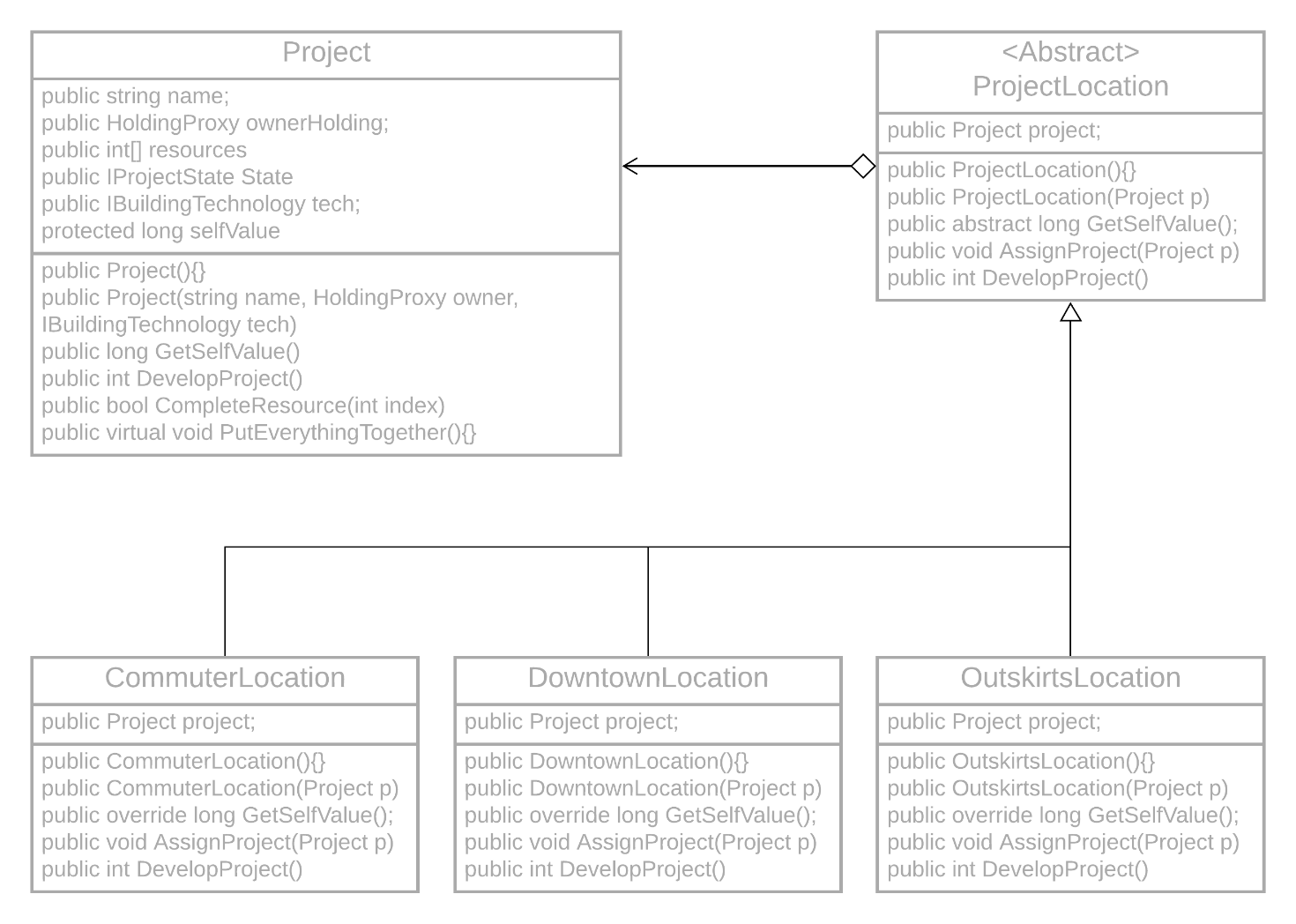
*Структура*.

**IProjectState –** інтерфейс, що оголошує стан.

**ReadyForBuildingState, ReadyForSaleState, NotReadyState –** реалізації інтерфейсу **IProjectState.**

**Project –** об’єкт проекту, що містить стан як поле.

*Обґрунтування використання даного шаблону*.Від стану **IProjectState** залежить поведінка **Project** під час викликів методів роботи з ним; не можна продати недобудований комплекс.

1. **Decorator**

*Рис. 2.1.5. UML-діаграма шаблону “****Декоратор****”*

*Cтруктурний шаблон* проектування, який дозволяє динамічно додавати об'єктів нову функціональність, обертаючи їх в корисні «обгортки».

*Структура*.

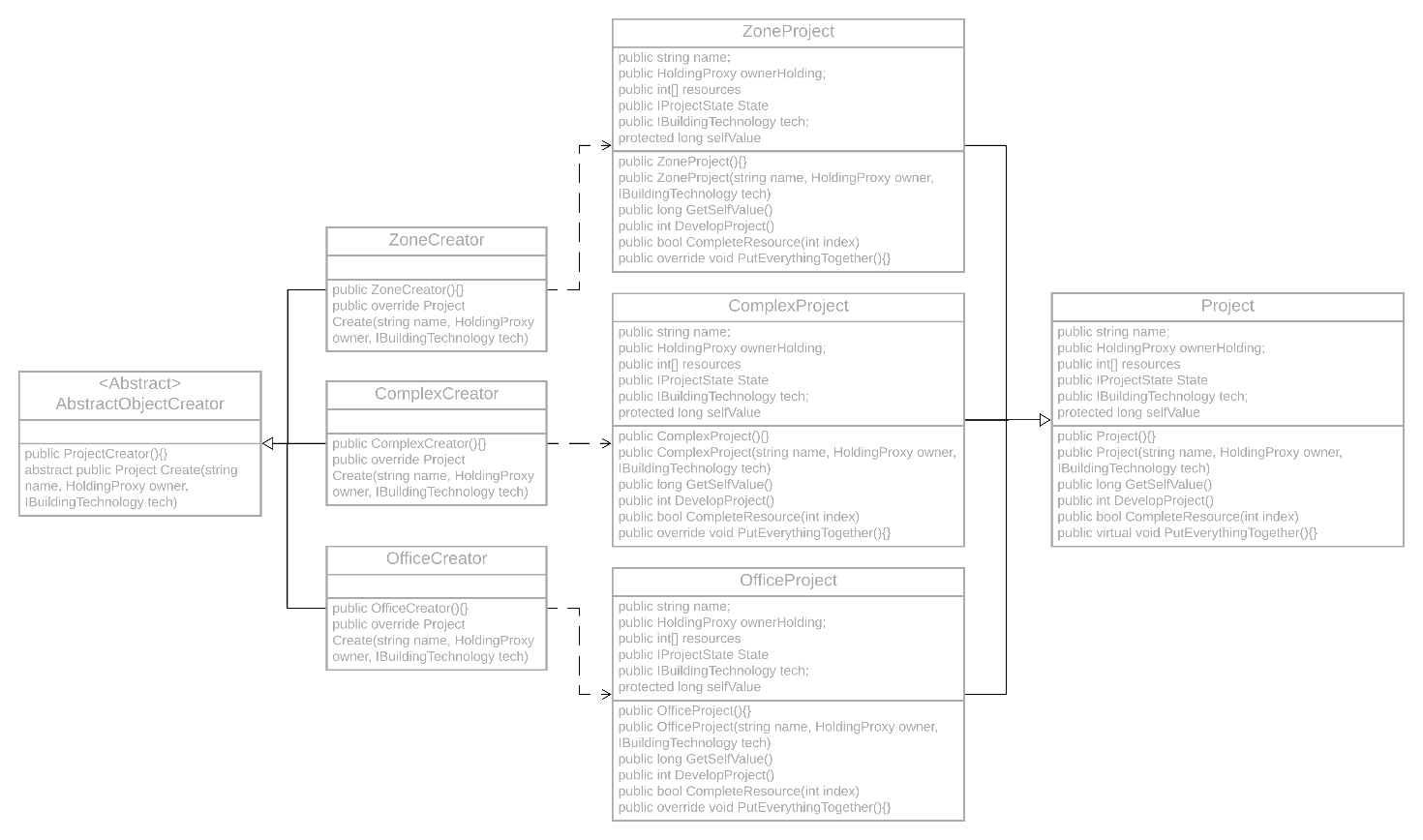
**ProjectLocation –** абстрактний клас, що визначає локацію **Project.**

**CommuterLocation, DowntownLocation, OutskirtsLocation –** реалізації **ProjectLocation.**

**Project –** об’єкт, що знаходиться у **ProjectLocation.**

*Обґрунтування використання даного шаблону*.Даний шаблон був використаний для надання нових властивостей **Project** оскільки він вже має дочірні класи.

1. **Factory Method**



*Рис. 2.1.6. UML-діаграма шаблону “****Фабричний Метод****”*

*Породжуючий шаблон*, який визначає інтерфейс для створення об’єкта, але рішення про те, який саме об’єкт створювати, залишає за підкласами.

*Структура*.

**AbstractObjectCreator –** абстрактний клас, що визначає методи фабрик.

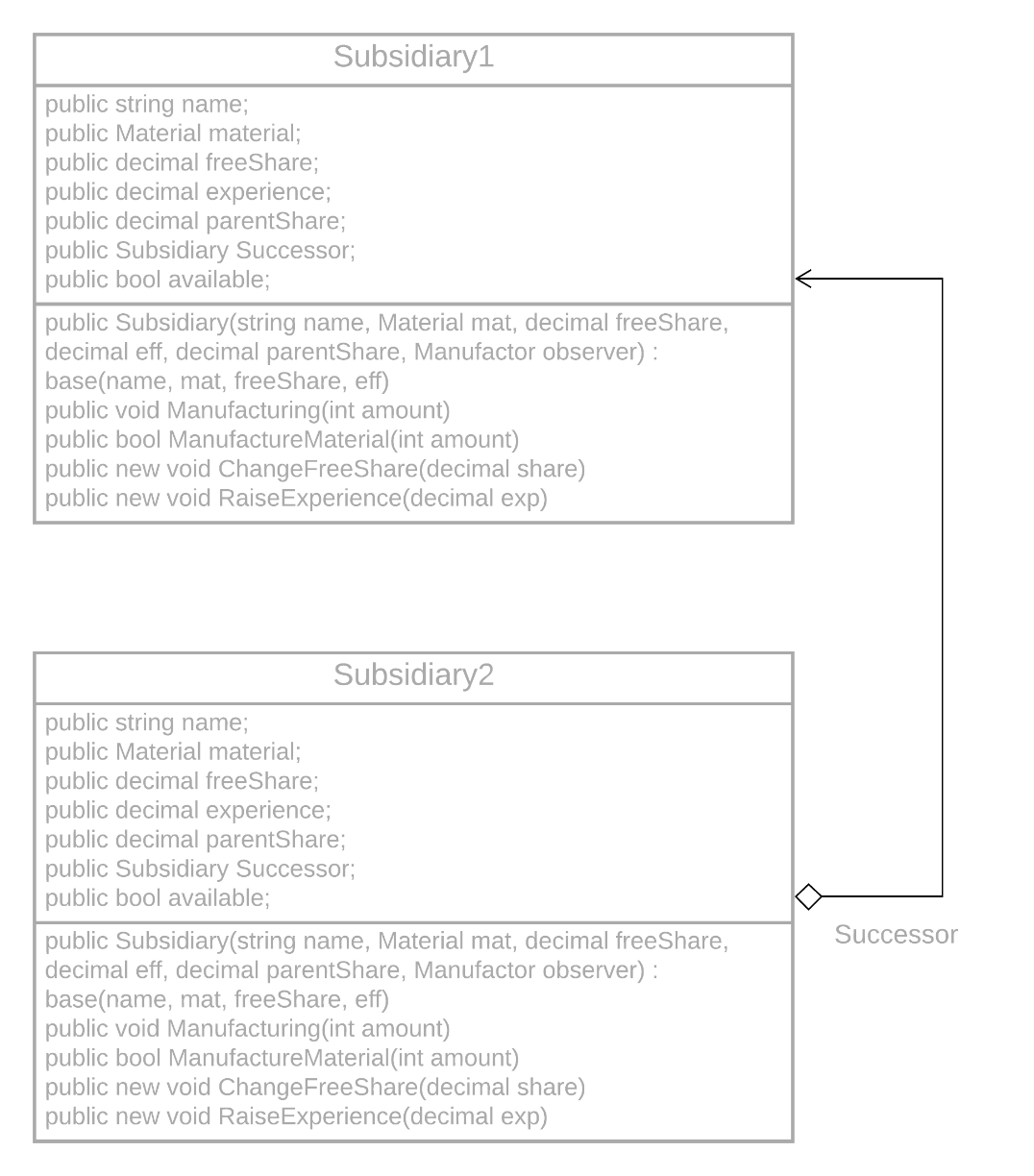
**Project –** об’єкт проекту з підкласами.

**ZoneProject, ComplexProject, OfficeObject –** підкласи **Project.**

**ZoneCreator, ComplexCreator, OfficeCreator –** реалізації класу **AbstractObjectCreator,** що створюють відповідні класи.

*Обґрунтування використання даного шаблону*.Даний шаблон був використаний для спрощення роботи з підкласами класу **Project.**

1. **Chain of Responsibility**



*Рис. 2.1.7. UML-діаграма шаблону “****Ланцюжок обов’язків****”*

*Поведінковий шаблон,* призначений для організації в системі рівнів відповідальності. Будує об’єкти складених частин програми зв’язаними між собою ланцюжком для передачі запиту на обробку від більш низьких, деталізованих прошарків до більш глобальних.

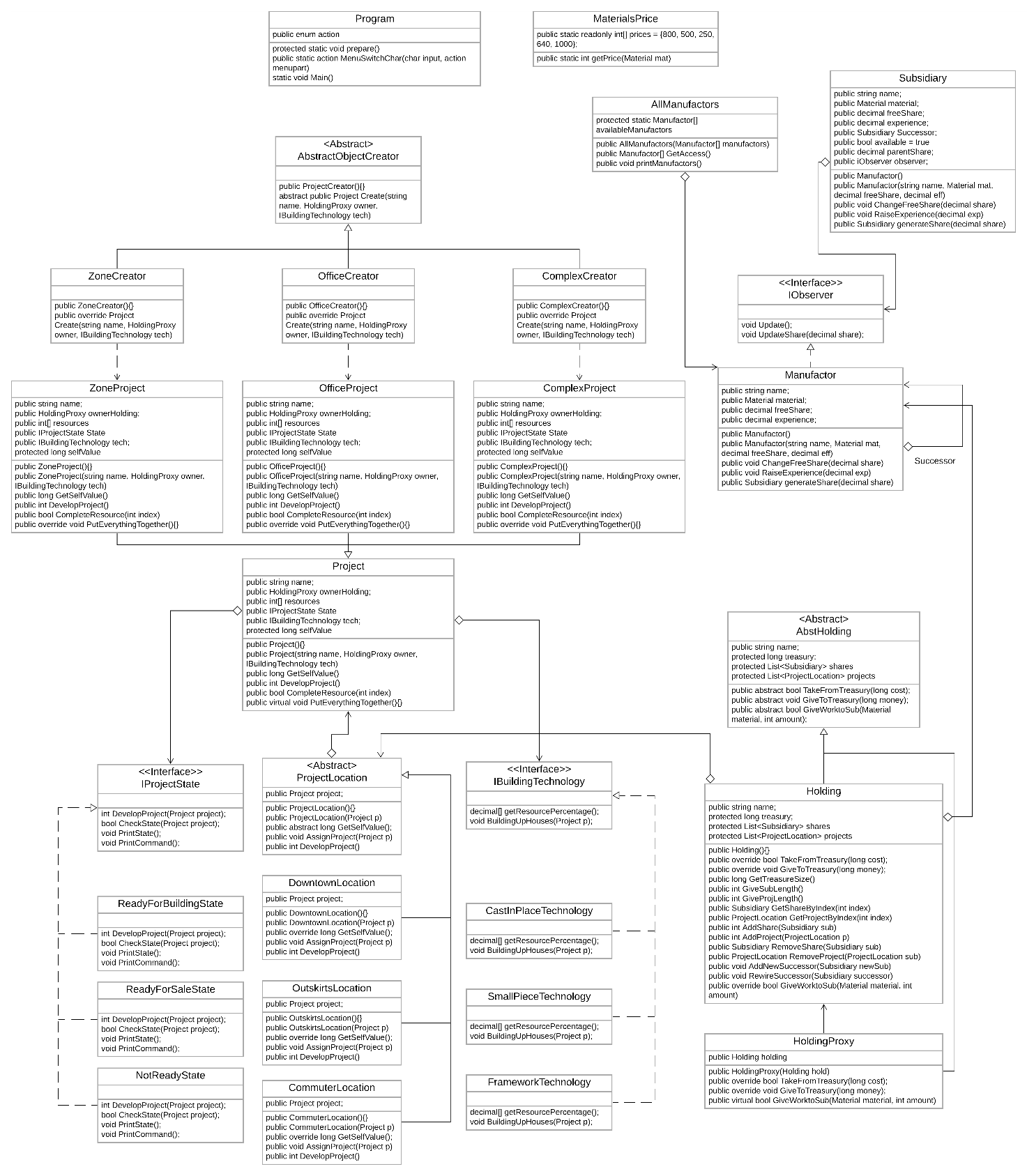
*Структура*.

**Subsidiary1 –** перша дочірня компанія, що має заступника **Subsidiary2.**

**Subsidiary2 –** друга дочірня компанія, що є заступником **Subsidiary1,** але не має свого заступника.

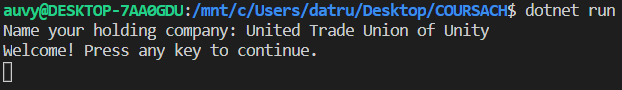
*Обґрунтування використання даного шаблону*.Даний шаблон був використаний, оскільки один об’єкт не може одночасно виконати два запроси по виробництву.

* 1. **Діаграма класів**

****

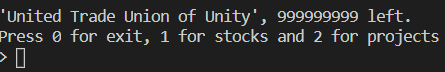
*Рис. 2.2.1. Діаграма класів*

* 1. **Результати роботи програми**

**Початок роботи з програмою починається з введення назви компанії

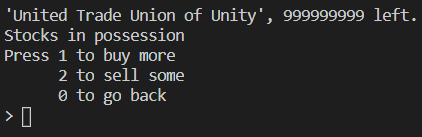
*Рис. 2.3.1-2. Ввід назви холдингу*

Основне меню холдингу, показує назву та баланс. 1 переходить до меню акцій, 2 до меню проектів, 0 закриває програму.



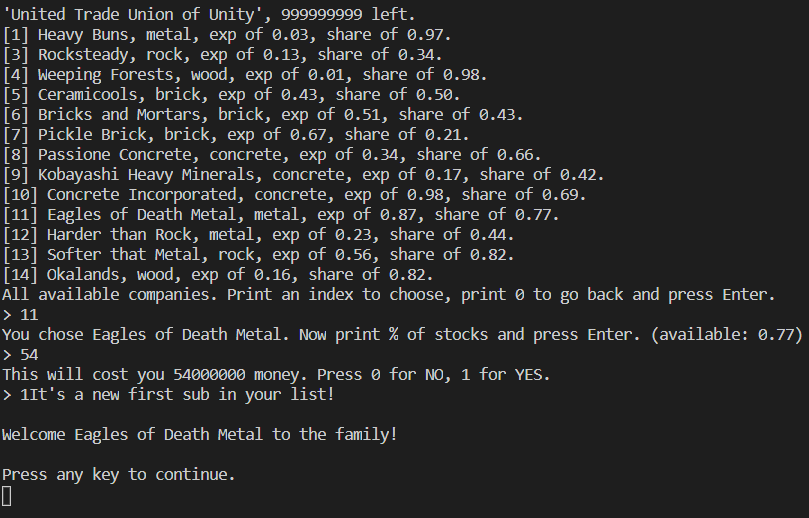
*Рис. 2.3.3. Головне меню*

Меню акцій. 1 переходить на біржу, 2 переходить до меню контролю куплених акцій, 0 повертає до головного меню.

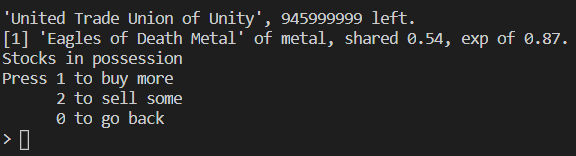
**

*Рис. 2.3.4. Меню акцій*

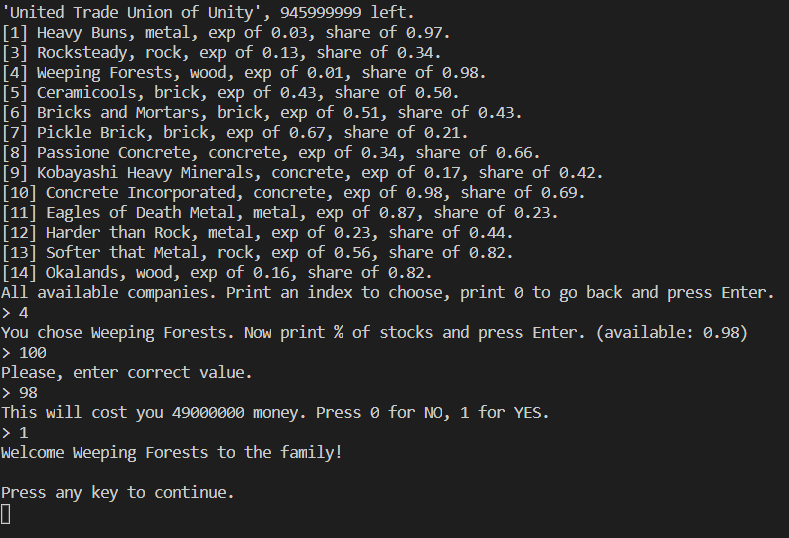
Меню біржі виводить усі доступі компанії з назвами, індексами, профілем (матеріалом, який вони виготовляють), кваліфікацією та доступним відсотком покупки. Користувач вводить індекс, відсоток (у вигляді цілого числа), бачить скільки це зніме грошей з рахунка і натискає 1 для підтвердження і 0 для скасування операції.

**

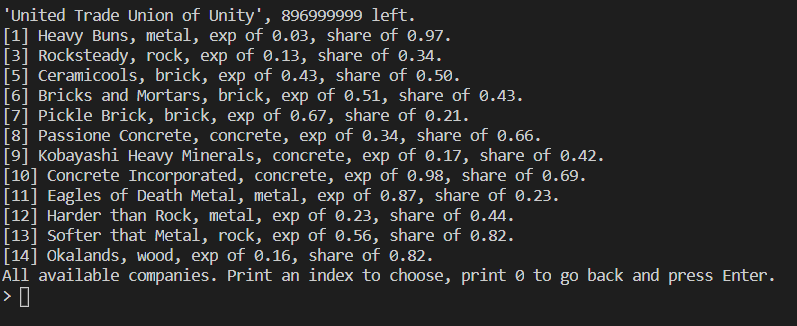
*Рис. 2.3.5. Меню біржі, покупка та підтвердження*



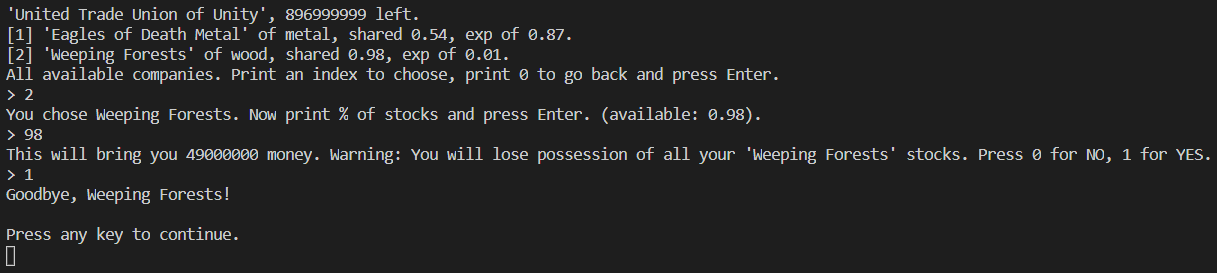
*Рис. 2.3.6. Меню акцій, результат покупки*



*Рис. 2.3.7. Меню біржі, покупка усіх акцій компанії*

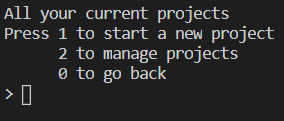
**

*Рис. 2.3.8-9. Результати покупки усіх акцій компанії*

Меню продажу акцій виводить усі куплені акції. Акції можна вибрати аналогічно біржі та продати. При виборі усіх акцій компанія зникає зі списку. ****

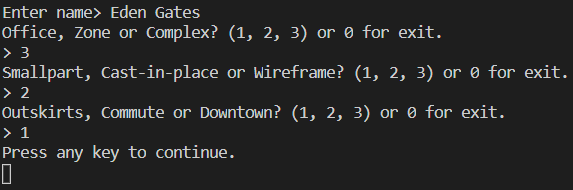
*Рис. 2.3.10. Результати покупки усіх акцій компанії*

Меню проектів виводить поточні проекти компанії. 1 переходить до меню створення, 2 переходить до меню керування, 0 переходить до головного меню.

****

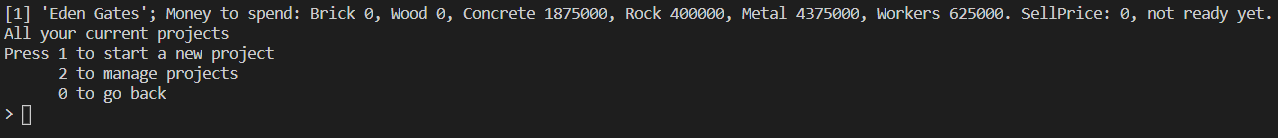
*Рис. 2.3.11. Меню проектів*

Меню створення проектів приймає назву користувача та дає вибір класифікації проекту. Проект класифікується по масштабу (office, zone, complex), технології будови (smallpart, cast-in-place, wireframe) та місцю локації (outskirts, commute, downtown).



*Рис. 2.3.12. Меню створення проекту*

Меню проектів виводить поточні проекти з індексами, іменами, усіма значеннями ресурсів, яких не вистачає, поточною собівартістю та статусом виконання.

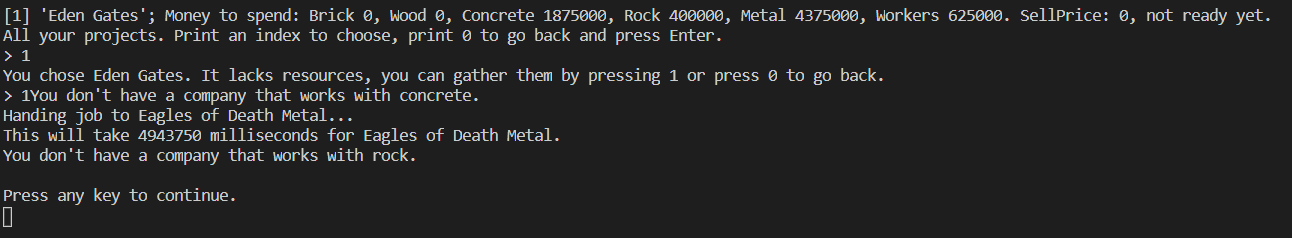


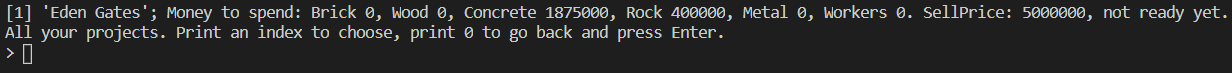
*Рис. 2.3.13. Меню проектів (2)*

Меню керування проектами виводить список усіх проектів з індексами. Користувач може вибрати по індексу та дати команду продовження роботи над проектом.

При незавершеному стані проект кличе дочірні компанії холдингу для задоволення ресурсних потреб будівництва. Якщо знайдеться компанія, що працює з потрібним ресурсом, проект знімає кошти холдингу, викликає компанію та задовольняє потреби конкретного ресурсу, компанія змінює статус на «Зайнята» на деякий проміжок часу, впродовж якого буде недоступна для виклику проектом.

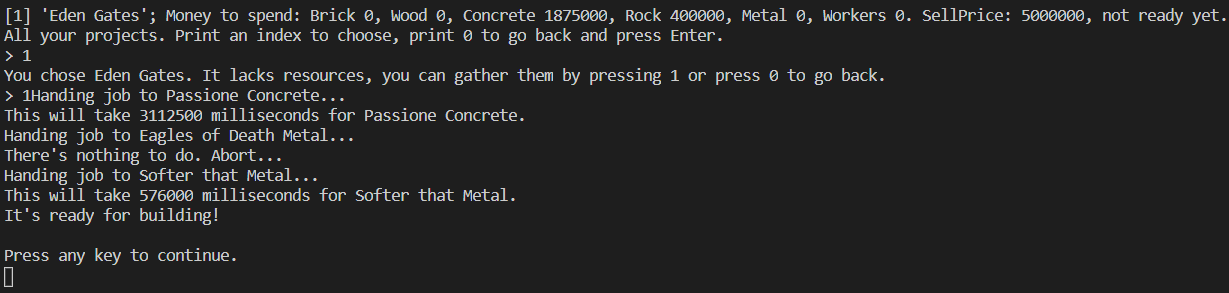
Якщо такої компанії немає, потреби ресурсу незадоволені.





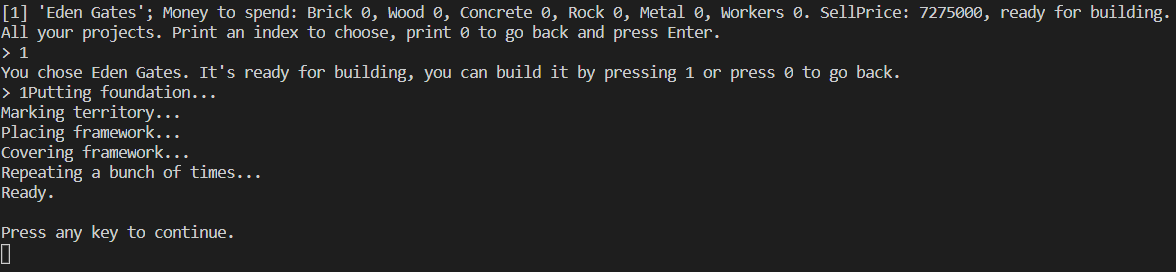
*Рис. 2.3.14-15. Меню проектів (3)*

(були куплені акції потрібних компаній, потреби проекту задоволені і ван готовий до будівництва)

****

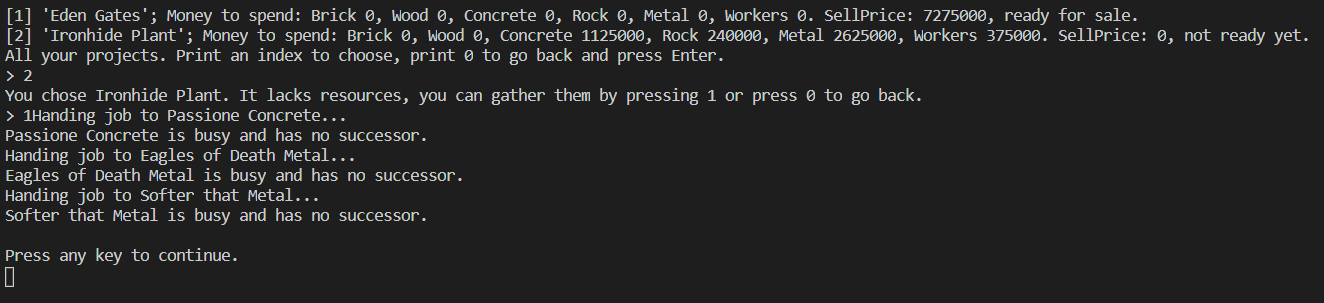
*Рис. 2.3.16. Меню керування проектом: збір матеріалів*

Будівництво різних масштабів відбувається по-різному.

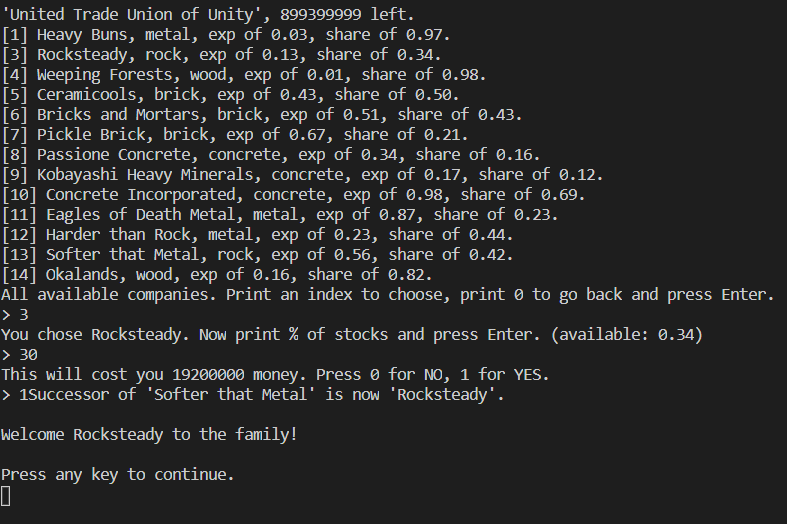


*Рис. 2.3.17. Меню керування проектом: будівництво*

Збір ресурсів для нового проекту невдалий, оскільки усі дочірні компанії зайняті та не мають наступників.

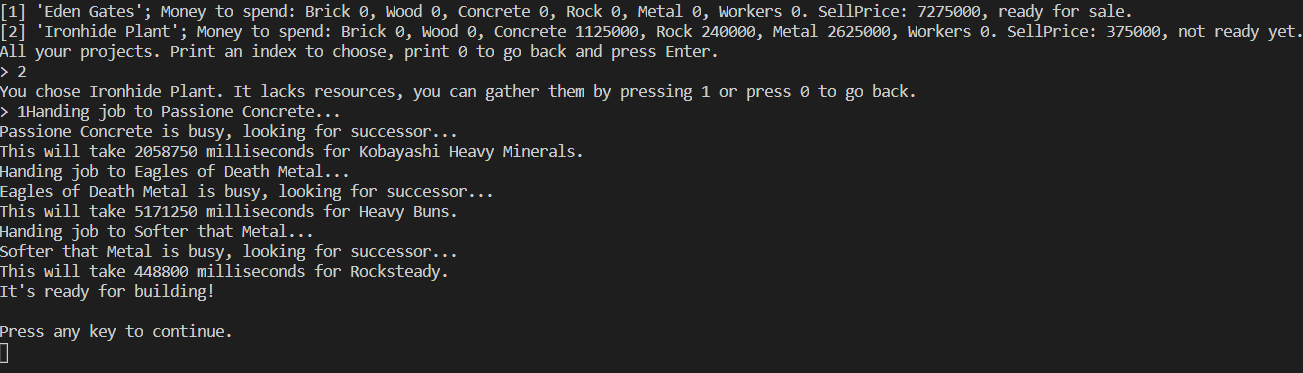


*Рис. 2.3.18. Меню керування проектом: ресурси нового проекту*

Покупка нових акцій: стара компанія отримала наступника, що може завершити інший проект.

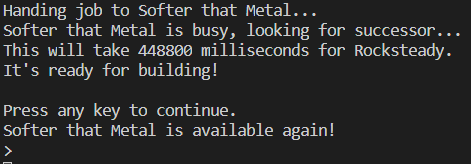
*Рис. 2.3.19. Меню біржі: наступник*

Після придбання потрібних акціонерів збір ресурсів завершився успішно.

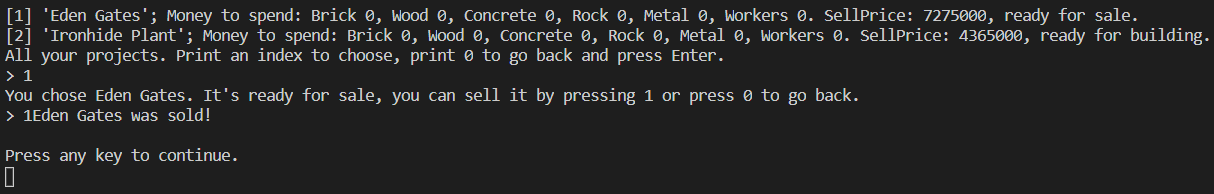
****

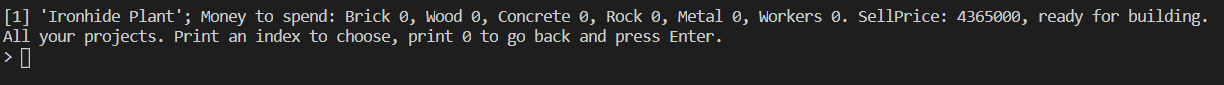
*Рис. 2.3.20. Меню проектів: наступники*

Поки відбувалось постачання ресурсів, одна з компаній знов стала вільною і може працювати над наступним проектом.

**

*Рис. 2.3.21. Меню проектів: вільна компанія*

Старий завершений проект можна продати й отримати гроші. Він зникає зі списку проектів.

**

*Рис. 2.3.22-23. Продаж проекту*

**ВИСНОВКИ**

Метою даної курсової роботи було розроблення програмного забезпечення холдингової компанії по будівництву з використанням шаблонів проектування. Підставою для розроблення стало завдання на виконання курсової роботи з дисципліни «Об’єктно-орієнтоване програмування» студентами ІІ курсу кафедри ПЗКС НТУУ «КПІ ім. І. Сікорського».

Для досягнення поставленої мети у повному обсязі виконано завдання, визначені у аркуші завдання на курсову роботу; розроблено графічні матеріали; реалізовано всі вимоги до програмного продукту, наведені у технічному завданні; створено відповідну документацію.

Розроблене програмне забезпечення дозволяє користувачу керувати холдингом, купляти акції будівельних компаній, продавати їх, створювати нові проекти, давати їх будівельним компаніям для реалізації та продавати їх.

Програму створено на основі використання шаблонів проектування. Зокрема, до структури програмного забезпечення входить реалізація семи різних шаблонів, які належать до різних груп.

Для розроблення програмного забезпечення була використана мова програмування C#.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Язык шаблонов. Города. Здания. Строительство. / Кристофер Вольфганг Александер. – 1977. – 1096 с.
2. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влиссидес. – 1994. – 395 с.
3. Руководство Microsoft по проектированию архитектуры приложений. / С. Сомасегар, Скотт Гатри, Дэвид Хилл. – 2009. – 529 с.
4. Шаблони проектування Методичні вказівки до виконання лабораторних робіт з дисципліни «Об’єктно-орієнтоване програмування»..

Заболотня Тетяна Миколаївна, Сулема Євгенія Станиславівна, Марченко Олександр Іванович. - 2015. - 154 с.

1. Патерни проектування [Електронний ресурс]: [Веб-сайт]. - Режим доступу: https://refactoring.guru/uk/design-patterns (дата звернення 30.03.2020) - Назва з екрана
2. Патерни проектування в C# та .NET [Електронний ресурс]: [Веб-сайт]. - Режим доступу: https://metanit.com/sharp/patterns (дата звернення 19.04.2020) - Назва з екрана