Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

з дисципліни «Об’єктно-орієнтоване програмування»

на тему

**Шаблони проектування в ООП. Комп’ютерна гра: Тетріс**

Виконав студент

ІІ курсу групи КП-42

Лисогор Дмитро Юрійович

залікова книжка КП-4209

Керівник роботи

доц., к.т.н. Заболотня Т.М.

Оцінка

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

(дата, підпис)

КИЇВ 2016

**ЗМІСТ**

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

1. **ОПИС СТРУКТУРНО-ЛОГІЧНОЇ СХЕМИ ПРОГРАМНОЇ СИСТЕМИ ОБЛІКУ ДАНИХ**…...................................................................5
   1. Модульна організація програми............................................................5
2. **ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЕКТУВАННЯ**..................................................................6
   1. Обґрунтування обраної реалізації структури програми та використаних у ній шаблонів.................................................................6
   2. Опис реалізованих класів.......................................................................7
   3. Опис реалізованих шаблонів...............................................................15
   4. Діаграма класів…..........................................................................……18

**ВИСНОВКИ**..............................................................................................................19

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**.....................................................20

**ДОДАТКИ**.................................................................................................................21

Додаток А. Опис інтерфейсу користувача…................................................21

**ВСТУП**

Дана курсова робота присвячена розробці комп’ютерної гри за допомогою використання шаблонів проектування. Дана тематика вибрана для курсової роботи тому, що результати абстрагування об’єктів у грі дозволяють застосувати всі вивчені методи та принципи об’єктно-орієнтованого програмування для створення програмного забезпечення (ПЗ), зокрема шаблони проектування.

*Об’єктом* курсової роботи є сама гра.

*Метою* роботи є розроблення ПЗ для гри з використанням шаблонів проектування.

Для досягнення визначеної мети необхідно виконати такі *завдання*:

* абстрагувати об’єкти предметної галузі;
* розробити структурну організацію ПЗ за допомогою використання основних принципів ООП та шаблонів проектування;
* визначити та описати функціональні характеристики програми;
* обґрунтувати вибір шаблонів проектування;
* розробити графічний інтерфейс користувача;
* виконати реалізацію ПЗ відповідно до технічного завдання (ТЗ);
* виконати тестування розробленої програми;
* оформити документацію з курсової роботи.

Розроблене ПЗ складається з 3-х логічних частин: інтерфейсу користувача, створення та відображення поля , модуля логіки взаємодії користувача з полем та зміни станів фігур на полі. Використані шаблони проектування: **Заступник, Адаптер, Легковаговик, Фабричний метод, Стан, Команда.**

До функціональних можливостей програми належать: користувачем маніпулювання фігурами, зберігання рекордів користувача, поставити гру на паузу . Для функціонування програми необхідно забезпечити наявність на комп’ютері приблизно 10 Мб дискового простору та ОС версії Windows, Linux,

Mac OS.

Розроблене ПЗ може бути використане з розважальною метою.

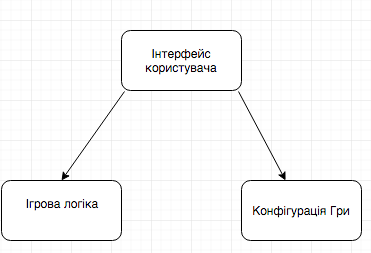
Пояснювальна записка складається зі вступу, двох розділів, загальних висновків, списку використаних джерел (4 найменування) та додатків (1 додаток). Робота містить 27 рисунків. Загальний обсяг роботи – 23 друковані сторінки, з них 20 сторінок основного тексту та 1 сторінка списку використаних джерел.

1. **ОПИС СТРУКТУРНО-ЛОГІЧНОЇ СХЕМИ ПРОГРАМНОЇ СИСТЕМИ ОББЛІКУ ДАНИХ**
   1. **Модульна організація програми**

У програмі можна виділити 3 основні логічні частини, що взаємодіють між собою. Модулям відповідають окремі класи чи сукупності класів. Основними модулями є: інтерфейс користувача, модуль створення та оновлення поля, модуль логіки взаємодії користувача з полем.

Найважливішим модулем є модуль логіки гри. Тут закладено взаємодію користувача із полем.

Другим за важливістю модулем є модуль створення поля. Він включає в себе створення нових фігур та маніпуляції над ними.

Модуль користувацького інтерфейса є посередником, який реалізує взаємодію між користувачем і самою грою.

1. **ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗА ДОПОМОГОЮ ШАБЛОНІВ ПРОЕКТУВАННЯ**
   1. **Обґрунтування обраної реалізації структури програми та використаних у ній шаблонів**

Для виконання даної курсової роботи шаблони обирались після детального ознайомлення з ними, вивчення та реалізації багатьох із них.

Застосовуючи абстрагування до предметної області програмного продукту, було виділено декілька модулів.

Під час запуску програми, при авторизації користувачів, виконується перевірка на коректність введених даних за допомогою шаблона «Заступник», який перевіряє чи було введене ім’я і чекає не дозволяє зайти без нього в гру.

При створенні нових фігур використовується шаблон проектування «Легковаговик». Використаний для створення 1 з 7 фігур, яка має обратися випадково та з’явитися на полі.. Взаємодія користувача з полем, адаптація інтерфейсу через маніпуляції користувача, реалізована через шаблон «Адаптер». Для створення певного типу панелі(поле чи ігрове меню) було використано шаблон фабричний метод, який за необхідністю створює об’єкт певного типу

Для структурування системи обробки маніпуляцій користувача використовується шаблон Команда, який виконує певну функцію, в залежності від натиснутої клавіші(зсув вліво – вправо, повернути фігуру за чи проти годинниковою стрілкою). Для можливості поставити гру на паузу було використано шаблон Стан.

* 1. **Опис реалізованих класів**

1. **Клас Adaptee**

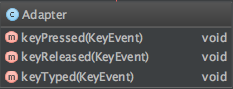
**../Desktop/Снимок%20экрана%202016-04-22%20в%2023.36.48.pngЗавдання:**

Клас використовується для функціонування шаблону «Адаптер».

**Методи:**

* Key() – Визначає існуючий інтерфейс класу, який потребує адаптації

1. **Клас Adapter**

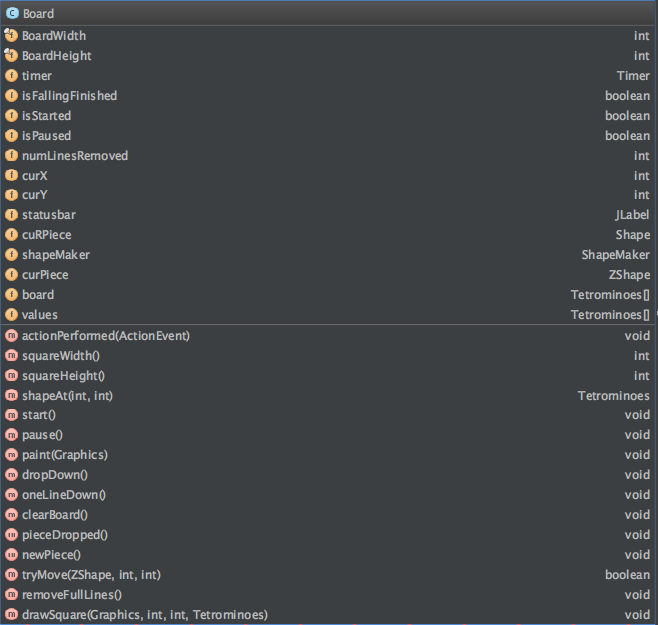


**Завдання:** Використовується для функціонування шаблону «Адаптер».

**Методи:**

* KeyPressed() – зберігає введені логін та пароль для подальшої перевірки.

1. **Клас Board**

****

**Завдання:** Використовується для функціонування шаблону «Адаптер».

Є одним з основних у цій программі

**Поля:**

* curX, curY – координати;
* isFallingFinished, isStarted, isPaused – поля які використовуються при перевірці;
* shapeMaker – повертає 1 з 7 можливих об’єктів фігур різного типу;
* timer – використовується для встановлення швидкості гри;

**Методи:**

* actionPerformed() – метод який описує створення та падіння фігури.
* start() – початок гри;
* pause() – зображає в статусбарі, що гра на паузі;
* paint() – малюємо всі об’єкти на дошці;
* dropDown() – пришвидшує падіння фігури;
* pieceDropped() – встановлює нову фігуру на дошку;
* newPiece() – створює нову фігуру;
* tryMove() – перевірка на колізію;
* removeFullLines() – перевірка чи є заповнений рядок, щоб стерти його;
* drawSquare() – розфарбування фігури;

1. **Клас Executor**

**Завдання:** Використовується для функціонування шаблону «Команда».

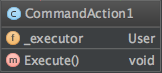
**../Desktop/Снимок%20экрана%202016-04-23%20в%2000.40.00.png**

Визначає інтерфейс для виконання операцій

**Методи:**

* Execute() –виконання команди.

1. **Клас CommandAction1,2,3,4,5,6**

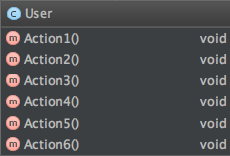
****

**Завдання:** Використовується для функціонування шаблону «Команда».

**Методи:**

* Execute() – збереження виконаної команди.

1. **Клас User**

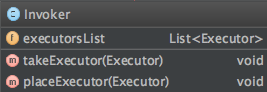
****

**Завдання:** Визначає інтерфейс для виконання операцій.

**Методи:**

* Action1() – визначає конкретну команду для виконання.
* Action2,3,4,5,6() – визначає конкретну команду для виконання

1. **Клас Invoker**

****

**Завдання:** Звертається до команди для виконання запиту, використовується для шаблону «Команда» .

**Методи:**

* takeExecutor() – Додає команду.
* placeExecutor() – Виконує команду;

1. **Клас Factory**

**Завдання:** створює інтерфейс для роботи з шаблоном «Фабричний метод».

../Desktop/Снимок%20экрана%202016-04-23%20в%2001.32.23.png

**Методи:**

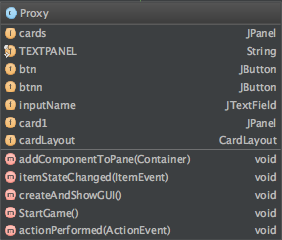
* getMenu() – в залежності від передання параметру, створює відповідний об’єкт.

1. **Інтерфейс Game**

../Desktop/Снимок%20экрана%202016-04-23%20в%2001.30.01.png**Завдання:** створенийдля роботи з шаблоном «Фабричний метод» та «Proxy».

**Методи:**

* StartGame() – Створення відповідних деталей в панелі.

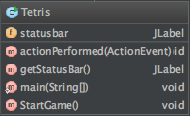
1. **Клас Proxy**

**Завдання:** створенийдля роботи з шаблоном «Фабричний метод» та «Proxy».

**Методи:**

* StartMenu() – виводить на екран панель головного меню.
* actionPerformed(ActionEvent) – контролює доступ до самої гри(тобто створення об’єкту)
* addComponentToPane(),ItemStateChanged(),
* createAbdShowGUI() – методи для створення панелі головного меню

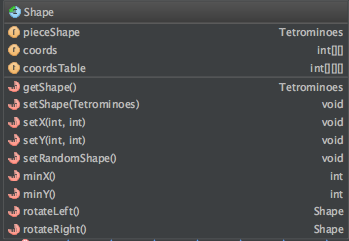
1. **Клас Tetris**

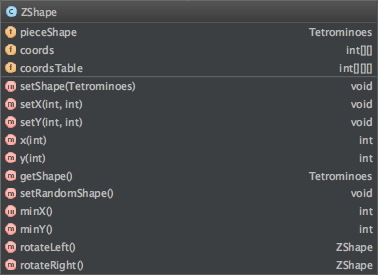
**Завдання:** створенийдля роботи з шаблоном «Фабричний метод» та «Proxy». Оди з основних класів програми

**Методи:**

* getStatusBar() – повертає статус бар в клас Board для його оновлення
* StartGame() – встановлення всіх компонентів на поле та початок гри
* actionPerformed() – дозволяє назад перейти до ігрового меню.

1. **Абстрактний клас Shape**

**Завдання:** для роботи з шаблоном «Легковаговик». Визначає інтерфейс, за допомогою якого пристосуванці можуть отримувати зовнішній стан або певним чином на нього впливати

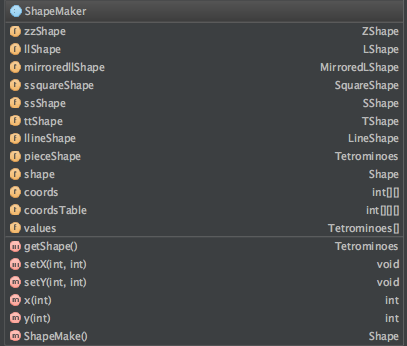
****

1. **Клас ZShape, SShape, LineShape, TShape,**

**SquareShape, LShape, MirroedLShape**

**Завдання:** Реалізує інтерфейс Shape

1. **Клас ShapeMaker**



**Завдання:** Забезпечує необхідне розділення пристосуванців

**Методи:**

* ShapeMake() – повертає одну з семи фігур;

1. **../Desktop/Снимок%20экрана%202016-04-23%20в%2002.22.18.pngКлас State**

**Завдання:** для роботи з шаблоном «Стан», створює інтерфейс для інкапсюляції поведінки

1. **../Desktop/Снимок%20экрана%202016-04-23%20в%2002.22.25.pngКлас PressedState**

**Завдання:** для роботи з шаблоном «Стан»

**Методи:**

actionPerformed() – переводить користувача в саму гру

1. **../Desktop/Снимок%20экрана%202016-04-23%20в%2002.22.32.pngКлас BackPressed**

**Завдання:** для роботи з шаблоном «Стан»

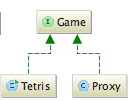
**Методи:**

actionPerformed() – переводить користувача в меню гри

1. **../Desktop/Снимок%20экрана%202016-04-23%20в%2002.22.39.pngКлас PauseOff**

**Завдання:** для роботи з шаблоном «Стан».При виклику об’єкта класу гра призупиняється або запускається в залежності від попереднього стану

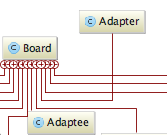
* 1. **Опис реалізованих шаблонів**

1. **Заступник**

Це структурний шаблон проектування, який забезпечує створення заступника об’єкта для контролю доступу до останнього через перехоплення всіх викликів.

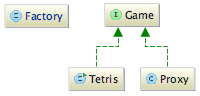
**Структура.** інтерфейс Game визначає спільний інтерфейс для класу заступника і класу, заступником якого він є. Клас Tetris визначає реальний об'єкт, який подається за допомогою заступника. Клас Proxy видає помилки при некоректному введенні вхідних даних і контролює доступ до об’єкту Tetris.

**Обґрунтування використання даного шаблону.** Даний шаблон в курсовій роботі використовується для перевірки авторизації користувача для подальшого доступу до об’єкта .

1. **Адаптер**

Це структурний шаблон проектування, призначений для організації використання функцій об'єкта, недоступного для модифікації, через спеціально створений інтерфейс. **Структура.** Клас Board визначає інтерфейс для користувача. Клас Adaptee адаптований інтерфейс. Визначає існуючий інтерфейс класу в залежності від дій користувача. Клас Adapter адаптує інтерфейс Adaptee до інтерфейсу Board.

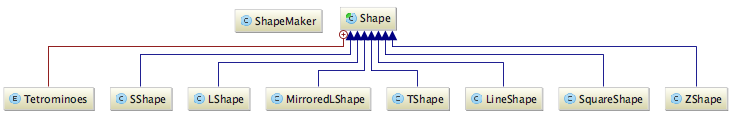
**Обґрунтування використання даного шаблону.** Даний шаблон забезпечує спільну роботу класів з несумісними інтерфейсами шляхом створення спільного об’єкта, через який вони можуть взаємодіяти.

1. **Фабричний метод**

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

**Структура.** Інтерфейс Game створює інтерфейс фрейму, яка створюється при викликанні методу. Конкретні класи Tetris, Proxy унаслідуються від інтерфейсу Game і реалізує кожен свій метод, який створює певний фрейм(в залежності від переданих параметрів). Клас Factory створює та повертає різні (конкретні) фрейми.

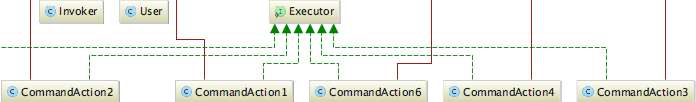
**Обґрунтування використання даного шаблону.** Шаблон було використано тому, що класу наперед невідомо, які будуть об’єкти та яке функціонування.

1. **Лековаговик**

Це структурний шаблон проектування, який структурує об'єкти таким чином, що з них створюється лише обмежений набір екземплярів замість великої множини об’єктів.

**Структура.** Клас Shape визначає інтерфейс, за допомогою якого пристосуванці можуть отримувати зовнішній стан або впливати на нього. Клас SShape,Lshape і тд реалізують різні інтерфейси фігури.Клас ShapeMaker створює об’єкти цих класів і при запиті користувача до нього надає новий екземпляр

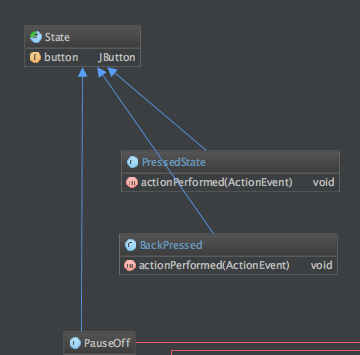
**Обґрунтування використання даного шаблону.** Шаблон використано тому, що потрібно обрати для створення 1 з 7 фігур і поставити її на поле, це дає можливість використовувати велику кількість об’єктів ефективно.

1. **Команда**

Поведінковий шаблон, який інкапсулює різні алгоритми в єдину сутність(об’єкт), завдяки чому можна параметризувати клієнтів різними запитами та вести історію виконаних операцій.

**Структура.** клас Executor визначає інтерфейс для виконання операцій. Клас User Має інформацію про способи виконання операцій, необхідних для задоволення запиту. Клас CommandAction1,2,3 реалізує операцію Execute шляхом виклику відповідних операцій об’єкту User. Клас Invoker Звертається до команди для виконання запиту, є ініціатором

**Обґрунтування використання даного шаблону.** Шаблон використаний для опису певних змін у інтерфейсі програми в залежності від маніпуляцій користувача

1. **Стан**

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

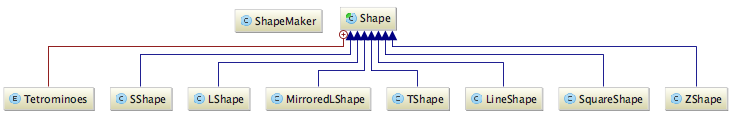
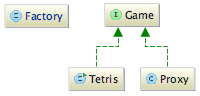
**Структура.**

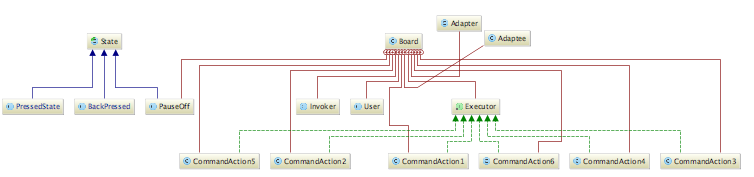
State **–** визначає інтерфейс для інкапсуляції поведінки, для конкретного стану контексту

**PressedState, BackPressedState –**

Реалізує різні стани в залежності від натискання на клавішу

* 1. **Діаграма класів**

****

****

**ВИСНОВКИ**

Метою даної курсової роботи було розроблення гри Тетріс 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».

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 ПЗ дозволяє -користувачу маніпулювання фігурами, можливість зберігати рекорди користувача, можливість користувачем поставити гру на паузу, автоматичне завершення гри при виконанні умови програшу..

Програму створено на основі використання шаблонів проектування: зокрема, до структури ПЗ входить реалізація шести шаблонів, які належать до різних груп.

Для розроблення програмного забезпечення використані мова програмування Java, Java Swing та java.awt.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ**

1. Uses of Design Patterns // Режим доступу: <https://sourcemaking.com/design_patterns>
2. Гамма, Э. Приемы объектно-ориентированого проектирования. Паттерны проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. – СПб: Питер, 2009. – 366 с.
3. Э. Фримен, Э. Фримен, К. Сьерра, Б. Бейтс. Паттерны проектирования. — СПб.: Питер, 2011. — 656 с.: ил.
4. Design Patterns:Elements of Reusable Object-Oriented Software/Erich Gamma, Richard Helm,Ralph Johnson, John M.Vlissides//Шаблони проектування-2012.- С.30-69

**ДОДАТКИ**

**Додаток А**

**Опис інтерфейсу користувача**

При початку роботи програми з’являється вікно входу в систему.

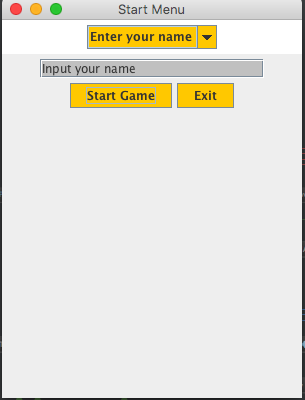


Рис.А.1. Вікно входу в систему

Після успішної авторизації створюється нова панель(самої гри)

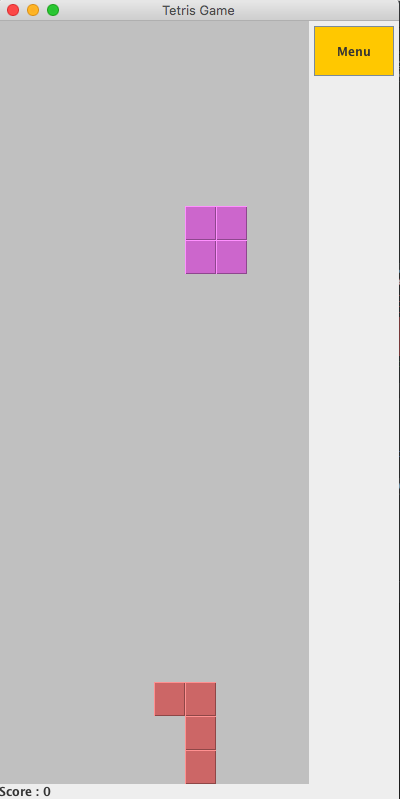


Рис.А.2. Вікно Гри

Приклад запису данних про користувачів в файл

Name  
Score : 8  
Hi  
Score : 4