МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**

з дисципліни « Об'єктно-орієнтоване програмування »

*назва дисципліни*

на тему: «Будівлі »

Виконав: студент 1 курсу групи № 621п

напряму підготовки(спеціальності)

6.121 інженерія програмного забезпечення

(шифр і назва ОП)

Вітковська О.С.

(прізвище й ініціали студента)

Керівник: асистент каф. 603

Дем’яненко Владислав Анатолійович

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала:

Кількість балів:

Члени комісії:

(підпис) (прізвище й ініціали)

(підпис) (прізвище й ініціали)

(підпис) (прізвище й ініціали)

Харків – 2022

ЗМІСТ

[Вступ 3](#_Toc101562507)

[Мета роботи 3](#_Toc101562508)

[Загальна постановка задачі 3](#_Toc101562509)

[1 Детальна постановка задачі 5](#_Toc101562510)

[2 Теоретичний вступ 6](#_Toc101562511)

[3 Архітектура проекту 7](#_Toc101562512)

[4 Опис класів будівель і їх списку 8](#_Toc101562513)

[5 Опис головної програми 9](#_Toc101562514)

[Висновки 10](#_Toc101562515)

# Вступ

## **Мета роботи**

1. Вивчити за літературними/Інтернет джерелами предметну область, уточнити з викладачем постановку задачі та побудувати її інфологічну модель (математичну, логічну, структурну, Use Case тощо).

2. Реалізувати у курсовому проекті основні засади ОВП, згідно

вибраній мові програмування (C++, C#, Java):

• інкапсуляція та методи доступу (закриті, відкриті, захищені);

• статичний чи динамічний поліморфізм;

• просте (або множинне) успадкування;

• прості, абстрактні та поліморфні класи

3. Використовувати текстові або двійкові файли користувача для запису та читання об'єктів

4. Зробити log-файл (повний протокол) роботи з програмою (для команди)

## **Загальна постановка задачі**

1. Перша частина завдання. Згідно зі своїм варіантом створити абстрактний клас з полями та віртуальними методами. На його основі реалізувати не менше 2-х похідних класів із поліморфними методами. Побудувати попередню UML діаграму. Зробити тестовий приклад та фрагменти пояснювальної записки.

2. Друга частина завдання. Створити клас, що містить масив/список/параметризовану колекцію об'єктів цих класів у динамічної пам'яті. Колекція об'єктів та їх поля мають бути отримані з використанням датчика псевдовипадкових чисел або за допомогою зрозумілого ручного коректного введення з клавіатури (з урахуванням значень полів, прийнятих у предметної галузі). Зробити тестовий приклад та фрагменти пояснювальної записки.

3. Третя частина завдання. Написати демонстраційну програму з використанням динамічного поліморфізму, в якій будуть використовуватися всі методи класів, для чого в режимі діалогу має бути організований доступ до будь-якого елементу колекції з обчисленнями, згідно з варіантом. Побудувати уточнену UML діаграму.

4. Передбачити варіант Debug для візуалізації виклику конструкторів та деструкторів, а також проміжних обчислень (за погодженням з викладачем).

5. Вихідні дані вводяться або вручну з перевіркою коректності введення, або генеруються з використанням датчика псевдовипадкових чисел із записом в файл користувача (за запитом в режимі діалогу). Дані повинні відповідати постановці задачі та мати відповідний тип (Арифметичний, символьний, рядковий).

6. Вихідні дані (поля класів) мають бути видані на дисплей (перегляд повний або частковий) або відредаговані (змінені, додані, видалені) запит у режимі діалогу.

7. Обчислення мають бути коректними, для чого мають бути використані методи опрацювання виняткових ситуацій.

8. Вихідна інформація видається на дисплей або записується в користувача файл з ехо-супроводом на дисплей (на запит у режимі діалогу).

# 1 Детальна постановка задачі

Створити абстрактний клас Building (будівля) із полями, що задають кількість кімнат, їх розміри, кількість і дані про людей, що там проживають/працюють, а також віртуальними методами виведення цих даних на екран. На його основі реалізувати класи Residential (житлова будівля), Office (офісне) та Warehouse (склад) з можливістю обчислення загальної площі будівлі, площі та периметра заданих кімнат, а також кількості та ПІБ проживаючих/працюючих там людей.

Створити клас District (група), що містить масив/список/параметризовану колекцію об'єктів цих класів у динамічній пам'яті. Передбачити можливість виведення всіх об'єктів списку та вибірки за конкретним об'єктом. Написати демонстраційну програму, в якій використовуватимуться всі методи класів.

Особливості:

* Програма реалізована мовою Java з використанням інтегрованої

середовища розробки IntelliJ IDEA 2021.

* Вихідні дані генеруються з використанням датчика псевдовипадкових

чисел.

- У програмі задіяно механізм умовної компіляції (Debug/Release)

- При виведенні на екран використовуються рядкові потоки (для +) або інші засоби (вказати КОНКРЕТНО). УКАЖЕМ КОГДА БУДЕМ ДЕЛАТЬ

# 2 Теоретичний вступ

2.1. Опис своєї предметної галузі (математичні формули, геометричні малюнки, посади в установі, співвідношення площ будівель, основні характеристики різних транспортних засобів і т.п.).

2.2 Абстрактний клас, його призначення та основні характеристики.

2.3 Поліморфний клас, його призначення та основні характеристики.

2.4 Покажчики на базовий клас. Їхні властивості.

2.5 Колекція об'єктів та їх обробка у динамічній пам'яті.

2.6 Статичний та динамічний поліморфізм. Поняття про раннє та пізнє зв'язування member-функцій.

2.7. Файлове введення-виведення.

2.8. Текстові та двійкові файли (або інші файли, відповідно до мови реалізації курсового проекту).

# 3 Архітектура проекту

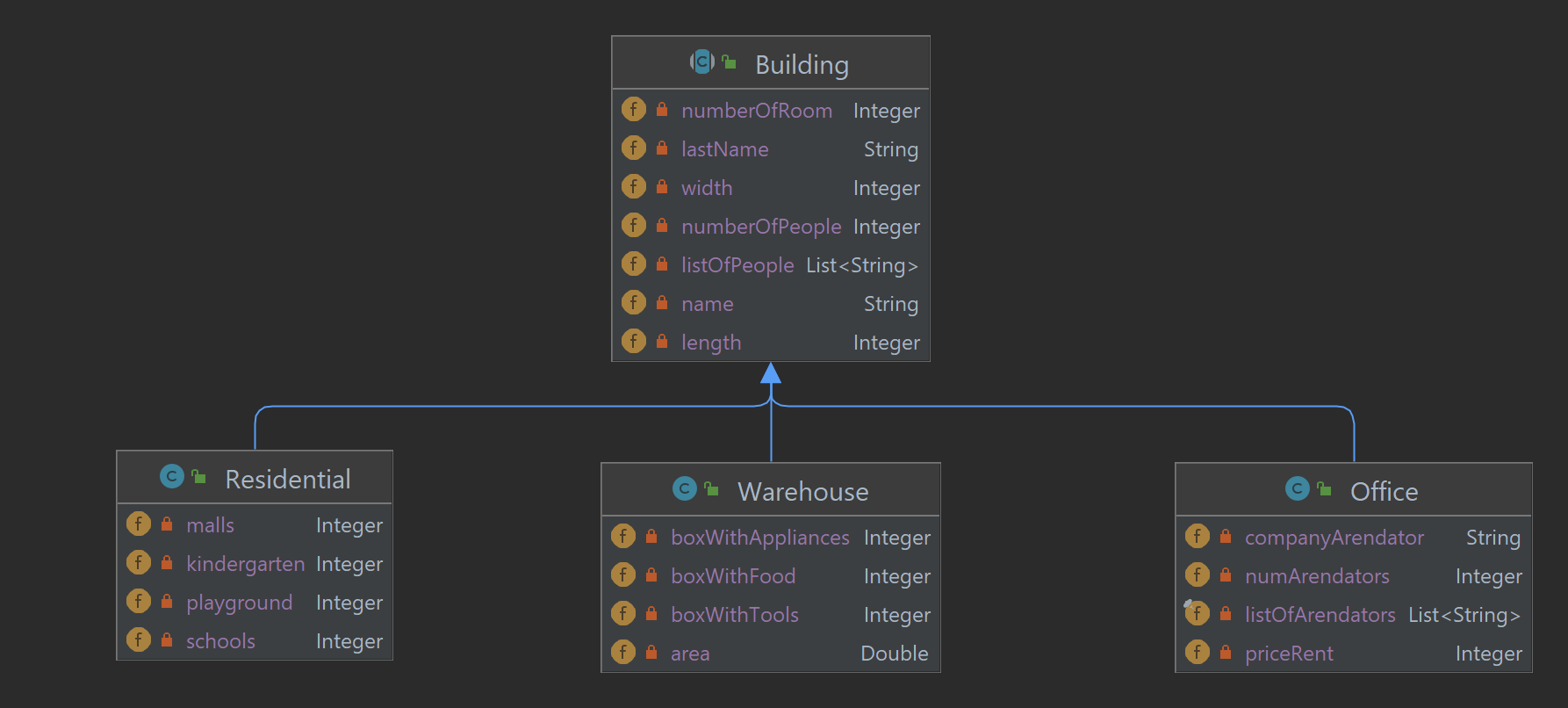


Рисунок 3.1 – Діаграма класів для представлення будівель та їх списку

# 4 Опис класів будівель і їх списку

4.1 **Клас Будівля – Building**

Описание класса как в примере

4.1.1 **Поля класу**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | **Тип** | **Рівень** **доступу** | **Опис** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

4.1.2 Методи класу

4.1.2 Описание методов как в примере

*Рівень доступу*

*Параметри*

*Значення, що повертається*

*Примітки*

# 5 Опис головної програми

5.1 Створення колекції будівель

Описание

Рисунок-алгоритм

Рисунок 5.1 – Алгоритм створення колекції будівель

5.2 Виведення інформації о будівлях в колекції

Описание

Рисунок-алгоритм

Рисунок 5.2 – Алгоритм виведення колекції будівель

5.3 Видалення колекції будівель

Описание

Рисунок-алгоритм

Рисунок 5.1 – Алгоритм видалення колекції будівель

# Висновки