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. Файлове введення-виведення.

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

# 3 Архітектура проекту

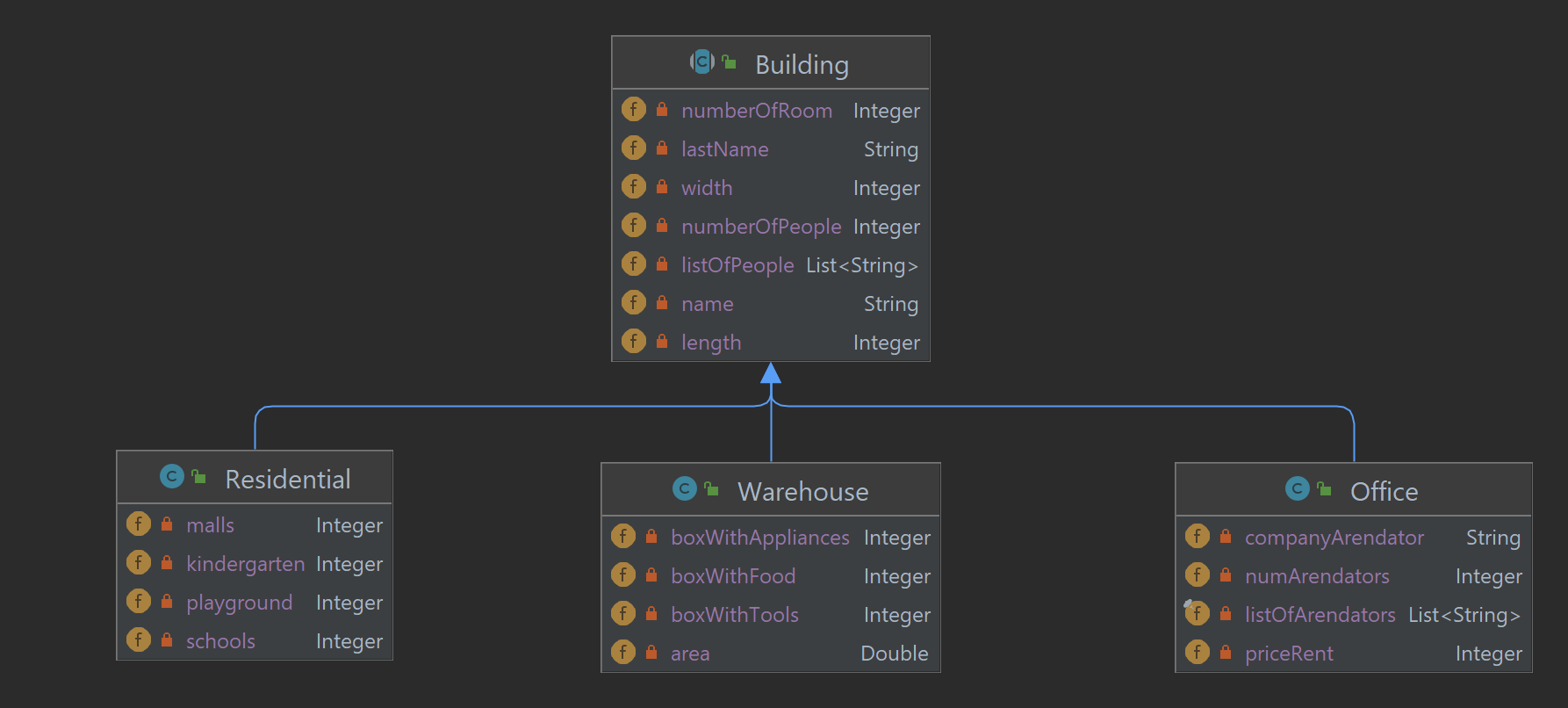


Рисунок 3.1 – Діаграма класів для представлення будівель та їх списку

# 4 Опис класів будівель і їх списку

4.1 **Клас Будівля – Building**

Абстрактний клас з полями, в яких вказується кількість кімнат, їх розміри, кількість і дані про людей, що живуть / там працюють, а також віртуальні методи відображення цих даних на екрані.

public abstract class Building {  
  
 private Integer numberOfRoom;  
 private Integer length;  
 private Integer width;  
  
 private List<String> listOfPeople = new ArrayList<>();  
 private String name;  
 private String lastName;  
 private Integer numberOfPeople;  
  
 public Integer getNumberOfRoom() {}  
  
 public void setNumberOfRoom(Integer numberOfRoom) {}  
  
 public Integer getLength() {}  
  
 public void setLength(Integer length) {}  
  
 public Integer getWidth() {}  
  
 public void setWidth(Integer width) {}  
  
 public String getName() {}  
  
 public void setName(String name) {}  
  
 public String getLastName() {}  
  
 public void setLastName(String lastName) {}  
  
 public Integer getNumberOfPeople() {}  
  
 public void setNumberOfPeople(Integer numberOfPeople) {}

public List<String> getListOfPeople() {}

@Override  
 public String toString() {}  
  
  
 public Integer areaOfRoom() {}  
  
 public Integer perimOfRoom() {}  
  
 public Integer areaOfBuilding() {}

4.1.1 **Поля класу**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | **Тип** | **Рівень** **доступу** | **Опис** |
| numberOfRoom | Integer | private | Кількість кімнат |
| length | Integer | private | Довжина кімнати |
| width | Integer | private | Висота кімнати |
| listOfPeople | List<String> | private | Список мешканців |
| name | String | private | Ім’я мешканця |
| lastName | String | private | Призвіще мешканця |
| numberOfPeople | Integer | private | Кількість мешканців |

4.1.2 Методи класу

4.1.2.1 public Integer getNumberOfRoom()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

numberOfRoom. Кількість кімнат.

*Примітки*

Нема.

4.1.2.2 public void setNumberOfRoom(Integer numberOfRoom)

*Рівень доступу*

public.

*Параметри*

Integer numberOfRoom. Кількість кімнат.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.1.2.3 public Integer getLength()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

length. Довжина кімнати

*Примітки*

Нема.

4.1.2.4 public void setLength (Integer length)

*Рівень доступу*

public.

*Параметри*

Integer length. Довжина кімнати.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.1.2.5 public Integer getWidth ()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

width. Ширина кімнати.

*Примітки*

Нема.

4.1.2.6 public void setWidth (Integer width)

*Рівень доступу*

public.

*Параметри*

Integer width. Ширина кімнати.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.1.2.7 public String getName ()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

name. Ім’я мешканця.

*Примітки*

Нема.

4.1.2.8 public void setName (String name)

*Рівень доступу*

public.

*Параметри*

String name. Ім’я мешканця.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.1.2.9 public String getLastName ()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

lastName. Призвіще мешканця.

*Примітки*

Нема.

4.1.2.10 public void setLastName (String lastName)

*Рівень доступу*

public.

*Параметри*

String lastName. Призвіще мешканця.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.1.2.11 public Integer getNumberOfPeople()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

numberOfPeople. Кількість мешканців.

*Примітки*

Нема.

4.1.2.12 public void setNumberOfPeople(Integer numberOfPeople)

*Рівень доступу*

public.

*Параметри*

Integer numberOfPeople. Кількість мешканців.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.1.2.13 public List<String> getListOfPeople()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

listOfPeople. Кількість мешканців.

*Примітки*

Нема.

4.1.2.14 public String toString()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

Рядок з інформацією про поля класу.

*Примітки*

Нема.

4.1.2.15 public Integer areaOfRoom()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

Площа кімнати.

*Примітки*

Нема.

4.1.2.16 public Integer perimOfRoom()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

Периметр кімнати.

*Примітки*

Нема.

4.1.2.17 public Integer areaOfBuilding()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

Площа будівлі.

*Примітки*

Нема.

4.1.2.18 public void addPeople()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.2 **Клас Будівля – Office**

Клас з можливістю розрахувати загальну площу будівлі, площу і периметр зазначених приміщень, а також кількість і ПІБ людей, що проживають / працюють там.

public class Office extends Building {  
 private String companyArendator;  
 private Integer numArendators;  
 private Integer priceRent;  
  
 private final List<String> listOfArendators = new ArrayList<>();  
  
 public Integer costOfRent() {}  
  
 public Integer getPriceRent() {}  
  
 public void setPriceRent(Integer priceRent) {}  
  
 public String getCompanyArendator() {}  
  
 public void setCompanyArendator(String companyArendator) {}  
  
 public List<String> getListOfArendators() {}  
  
 public Integer getNumArendators() {}  
  
 public void setNumArendators(Integer numArendators) {}  
  
 @Override  
 public String toString() {}

4.2.1 **Поля класу**

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** | **Тип** | **Рівень** **доступу** | **Опис** |
| numArendators | Integer | private | Кількість арендаторів |
| priceRent | Integer | private | Ціна ренти |
| companyArendator | String | private | Назва компанії |
| listOfArendators | List<String> | private | Список арендаторів |

4.2.2 Методи класу

4.2.2.1 public Integer getPriceRent()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

priceRent. Ціна ренти.

*Примітки*

Нема.

4.2.2.2 public void setPriceRent(Integer priceRent)

*Рівень доступу*

public.

*Параметри*

Integer priceRent. Ціна ренти.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.2.2.3 public Integer getNumOfArendators()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

priceRent. Кількість арендаторів.

*Примітки*

Нема.

4.2.2.4 public void setNumOfArendators(Integer numOfArendators)

*Рівень доступу*

public.

*Параметри*

Integer priceRent. Кількість арендаторів.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.2.2.5 public String getCompanyArendator ()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

companyArendator. Назва компанії.

*Примітки*

Нема.

4.2.2.6 public void setCompanyArendator (String companyArendator)

*Рівень доступу*

public.

*Параметри*

String companyArendator. Назва компанії.

*Значення, що повертається*

Нема.

*Примітки*

Нема.

4.2.2.7 public List<String> getListOfArendators()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

listOfArendators. Список арендаторів.

*Примітки*

Нема.

4.2.2.8 public String toString()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

Рядок з інформацією про поля класу.

*Примітки*

Нема.

4.2.2.9 public Integer getCostRent()

*Рівень доступу*

public.

*Параметри*

Нема.

*Значення, що повертається*

Ціна ренти.

*Примітки*

Нема.

# 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