МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Лабораторна робота № 2**

з дисципліни « ООП на Java »

*назва дисципліни*

на тему: «УСПАДКУВАННЯ КЛАСІВ У МОВІ JAVА»

Виконав: студент 2 курсу групи № 622п

освітньої програми

121 інженерія програмного забезпечення

(шифр і назва ОП)

Зайченко Ярослав Ігорович

(прізвище й ініціали студента)

Прийняв: доцент

Симбірський Генадій Дмитрович

(посада, науковий ступінь, прізвище й ініціали)

Кількість балів:

Харків – 2023

**ЗМІСТ**

[Постановка завдання 3](#_Toc158672321)

[Зміст звіту 3](#_Toc158672322)

[Порядок виконання роботи 3](#_Toc158672323)

[Виконання роботи 5](#_Toc158672324)

[Висновок 10](#_Toc158672325)

[Контрольні питання 11](#_Toc158672326)

Мета роботи: отримання практичних навичок із застосування класів та методів при проектуванні об'єктно-орієнтованих програмних додатків мовою Java.

Постановка завдання

Розробити та реалізувати мовою Java програмний додаток для моделювання властивостей, поведінки та взаємодії об'єктів. Предметною областю для цього додатка можуть бути об'єкти, наведені у п. 3 цієї роботи, або аналогічні об'єкти (наприклад з попередньої лабораторної роботи. Для заданої предметної області створити ієрархію класів. У додатку продемонструвати використання механізмів успадкування, абстрактних класів та методів, перевизначення методів у підкласах, а також застосування механізмів висхідних і низхідних перетворень типів об'єктів та поліморфізму.

Зміст звіту

1. Постановка завдання.
2. Встановлення завдань, які вирішуються в лабораторній роботі
3. Опис концептуальної моделі для обраної категорії об’єктів.
4. Відображення структурних відносин між класами за допомогою діаграми класів UML.
5. Реалізація класів.
6. Реалізація класу для тестування додатку.
7. Ручне тестування додатку.
8. Результат створення документації на розроблений додаток з допомогою **javadoc.exe.**
9. Висновки по роботі.

Порядок виконання роботи

1. Вивчити теоретичний матеріал, пов'язаний з темою даної лабораторної роботи, використовуючи конспект лекцій, літературу, що рекомендується, і документацію, що входить до складу пакету Java SE JDK.
2. Розробити концептуальну модель для заданої категорії об'єктів у вигляді словесного опису їх основних властивостей та поведінки.
3. Спроектувати абстрактний клас, у якому буде визначено найбільш загальні характерні властивості заданої категорії об'єктів.
4. Визначити найбільш характерні риси поведінки заданої категорії об'єктів і подати їх у вигляді набору методів, визначених в абстрактному класі.
5. Розробити підкласи для об'єктів, які успадковують клас узагальненого об'єкта (абстрактний) та реалізують поведінку (методи). Ці методи мають бути задекларовані у заздалегідь розроблених абстрактних класах або суперкласах.
6. У розроблених підкласах перевизначити методи equals(), hashCode() та toString(), які визначені у класі Object.
7. Відобразити структурні відносини між спроектованими класами та інтерфейсами за допомогою діаграми класів UML.
8. Реалізувати програмний код класів на мові Java. При реалізації програмного коду програми обов'язково використовувати коментарі та дескриптори документування для опису класів, конструкторів, методів та полів.
9. Відкомпілювати розроблену програму та запустити на виконання.
10. Протестувати розроблений додаток.
11. Отримати документацію на розроблений додаток за допомогою утиліти javadoc.exe.
12. Зробити висновки щодо роботи.
13. Оформити звіт з лабораторної роботи відповідно до вимог, що пред'являються.

Завдання для варіанту 10:

Транспорті засоби (автобуси, трамваї, тролейбуси).

Властивості: тип моделі, фірма-виробник, номер, маршрут, місткість, закріплений водій.

Поведінка: визначення та зміна властивостей, визначення стану тощо.

Виконання роботи

Загальні відомості

Мова програмування – Java

Операційна система – Windows 11 Pro

Процесор – Apple Silicon M1 Pro with 6 CPU cores

Тип компілятора – IntelliJ IDEA 2024

Концептуальна модель об'єктів

Властивості:

* Тип моделі: рядок або перерахування (enum), що вказує на тип транспортного засобу, наприклад, "автобус", "трамвай", "тролейбус".
* Фірма-виробник: Рядок, що вказує на компанію, яка виготовила транспортний засіб.
* Номер: Унікальний ідентифікатор транспортного засобу, може бути числом або рядком, залежно від системи ідентифікації.
* Маршрут: Рядок або перерахування, що вказує на маршрут, за яким рухається транспортний засіб.
* Місткість: Ціле число, що вказує на кількість пасажирів, які можуть перевозитися транспортним засобом.
* Закріплений водій: Це може бути об'єкт або посилання на об'єкт, що відповідає водію, який керує транспортним засобом.

Поведінка:

* Визначення та зміна властивостей: Методи для встановлення та зміни властивостей об'єкта, таких як тип моделі, фірма-виробник, номер, маршрут, місткість та закріплений водій.
* Визначення стану: Методи для визначення поточного стану транспортного засобу, наприклад, чи знаходиться він у русі, скільки пасажирів знаходиться на борту, водій за кермом, на якій зупинці.

Абстрактний клас, у якому визначено найбільш загальні характерні властивості заданої категорії об'єктів

Створений абстрактний клас з загальними характерними властивостями та поведінкою поданий в Додатку А.

Розроблені підкласи з об'єктами, які успадковують абстрактний клас та реалізують поведінку.

Створені класи транспорту автобус, тролейбус та трамвай подані в Додатках Б, В, Г.

При створенні підкласів об’єктів були перевизначені методи класу Object, а саме: equals(), hashCode() та toString(). Їх реалізацію можна переглянути у відповідних класах об’єктів.

Структурні відносини між спроектованими класами за допомогою діаграми класів UML.

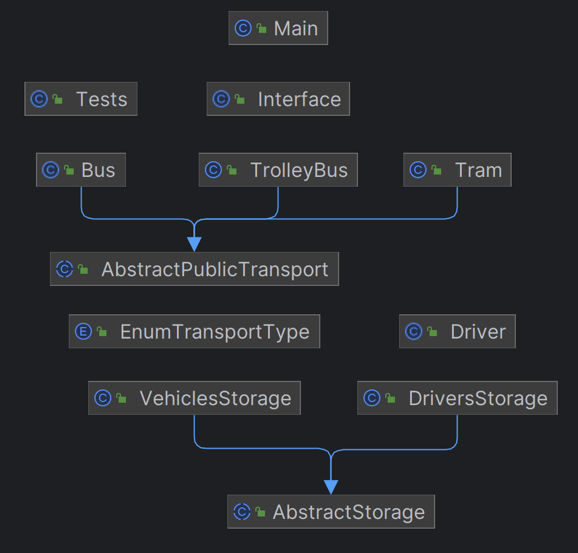


Рисунок 1 – згенерована діаграма у середовищі розробки

Реалізувати програмний код класів на мові Java.

При написанні коду були виконані умови: абстрактний клас з характерними властивостями та рисами поведінки. Підкласи об’єктів, які успадковують абстрактний клас та реалізовують поведінку об’єкту, перевизначені методи класу Object в кожному класі окремо. Фрагменти фунціоналу коду описані коментарями. Програмний код класів можна переглянути в додатках А-І.

Тестування програми

Тестувати будемо правильними та неправильними значеннями. Всі тестування були виконані за допомогою ручного введення даних. Результат на рисунку 4-8.

Рисунок 4 – виконання всіх тестових випадків

Генерація документації на розроблений додаток

Для цього необхідно скористатись командою **javadoc**. В даному випадку вона буде написана таким чином: ***javadoc -d docs Main.java***. Результат виконання команди для створення документації зображено на рисунку 5.

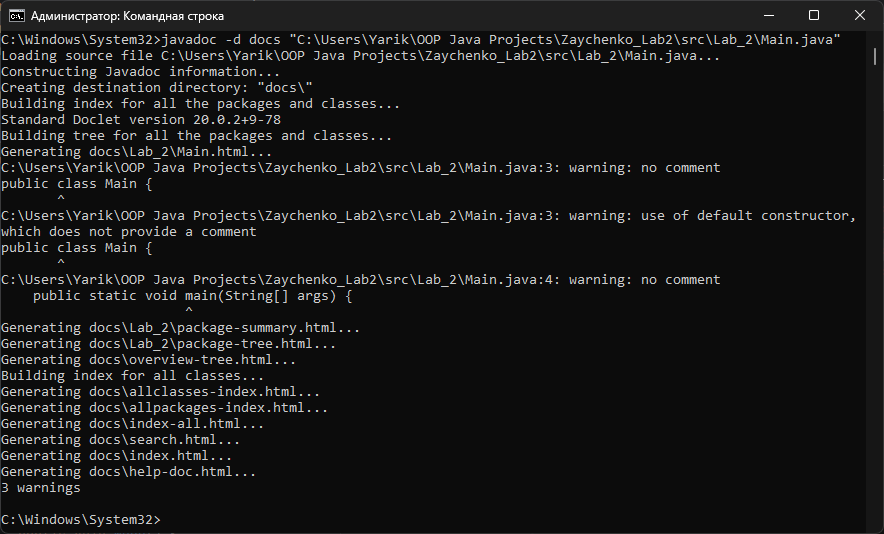


Рисунок 5 – створення документації додатку стандартними командами JRE

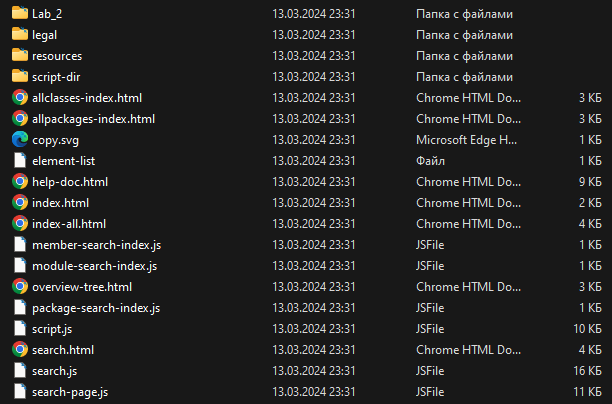


Рисунок 6 – отримані файли документації на додаток

Висновок

В ході даної лабораторної роботи було проведено дослідження теоретичного матеріалу, пов'язаного з програмуванням на мові Java та використанням його засобів для моделювання об'єктів та їх взаємодії. Згідно з вимогами завдання, спочатку була розроблена концептуальна модель для заданої предметної області, описуючи основні властивості та поведінку об'єктів. На основі цієї моделі було спроектовано абстрактний клас, що містив у собі найбільш загальні характеристики цих об'єктів.

Визначено основні риси поведінки об'єктів та представлено їх у вигляді набору методів, визначених у абстрактному класі. Після цього були реалізовані підкласи, які успадковували абстрактний клас та реалізували відповідну поведінку через перевизначення абстрактних методів.

У розроблених підкласах були перевизначені методи `equals()`, `hashCode()` та `toString()`, що дозволило правильно порівнювати об'єкти за їх вмістом, визначати унікальний ідентифікатор об'єкта та отримувати рядкове представлення об'єкту в зручному для виведення форматі.

Для відображення структурних відносин між класами та інтерфейсами була створена діаграма класів UML, що спрощує розуміння структури програми та взаємозв'язків між її складовими.

Після реалізації програмного коду була проведена компіляція та запуск програми для перевірки її працездатності. Розроблений додаток було протестовано, що дозволило переконатися в правильності його функціонування.

Завершальним етапом було отримання документації на розроблений додаток за допомогою утиліти **javadoc.exe**, що дозволяє автоматично генерувати документацію на основі коментарів у програмному коді. В результаті вдалої роботи над проектом було успішно досягнуто поставлених цілей та вимог завдання.

Контрольні питання

1. Який клас називають абстрактним?

2. Навіщо застосовуються абстрактні класи?

3. Який синтаксис має оголошення абстрактного класу?

4. Який синтаксис має оголошення підкласу?

5. Які модифікатори використовуються для оголошення класу?

6. Який метод оголошується як абстрактний?

7. Який синтаксис має оголошення абстрактного методу?

8. Які модифікатори використовуються для оголошення абстрактного методу?

9. Який метод називають перевантаженим?

10. Який метод називають перевизначеним?

11. Для чого використовують перевизначення методів?

12. У чому полягає відмінність між методами, що перевантажуються і перевизначаються?

13. Для яких цілей застосовують змінну super?

14. Що таке висхідне перетворення типу об'єкту?

15. Що таке низхідне перетворення типу об'єкту?

16. Що може зберігатися в змінній, яка має тип суперкласу?

17. Що може зберігатися в змінній, яка має тип підкласу?

18. Для яких цілей застосовують оператор instanceof?

*Додаток А*Машинний лістинг файлу **Main.java**

import Classes.Testing;

import java.util.Scanner;

public class Main { // клас мейн

public static void main(String[] args) { // початок виконання програми в методі мейн

// Створення об'єкта Scanner для зчитування введеного тексту з консолі

Scanner scanner = new Scanner(System.in);

boolean end = false;

while (!end) {

try {

// Виведення повідомлення, щоб користувач ввів текст

System.out.print("Введіть ім'я цуценя:");

String inputName = scanner.nextLine(); // Зчитування введеного тексту

Testing.StartTest(inputName); //виклик методу з класу для тестів

end = true; //все пройшло без помилок - можна виходити

} catch (IllegalArgumentException ex) {

System.out.println(ex.getMessage()); // виведення тексту помилки

}

}

Testing.AllTest();

}

}

*Додаток Б*Машинний лістинг файлу **AbstractPublicTransport.java**

package Lab\_2.Classes.Transport;

import java.util.Arrays;

public abstract class AbstractPublicTransport {

// Властивості

protected EnumTransportType type;

protected String modelType; // модель

protected String manufacturer; // иробник

protected String number; // номер маршруту

protected String[] route = new String[2]; // маршрут

protected int capacity; // місткість

protected Driver driver; // закріплений водій

protected String stopped; // зупинка

// Конструктор

public AbstractPublicTransport(EnumTransportType tt, String modelType, String manufacturer, String number, String route\_start, String route\_end, int capacity) {

this.type = tt;

setModelType(modelType);

setManufacturer(manufacturer);

setNumber(number);

setRoute(route\_start, route\_end);

setCapacity(capacity);

this.stopped = route\_end;

}

// Методи доступу до властивостей

public String getModelType() {

return modelType;

}

public void setModelType(String modelType) {

this.modelType = modelType;

}

public String getManufacturer() {

return manufacturer;

}

public void setManufacturer(String manufacturer) {

this.manufacturer = manufacturer;

}

public String getNumber() {

return number;

}

public void setNumber(String number) {

this.number = number;

}

public String[] getRoute() {

return route;

}

public void setRoute(String route1, String route2) {

this.route[0] = route1;

this.route[1] = route2;

}

public int getCapacity() {

return capacity;

}

public void setCapacity(int capacity) {

this.capacity = capacity;

}

public Driver getAssignedDriver() {

return driver;

}

public void setAssignedDriver(Driver assignedDriver) {

this.driver = assignedDriver;

}

// Методи поведінки

public void Movement() // метод руху

{

if (this.stopped == this.route[0]) {

this.stopped = this.route[1];

} else if (this.stopped == this.route[1]) {

this.stopped = this.route[0];

}

}

public abstract void assignDriver(Driver driver);

public abstract void assignModel();

@Override

public String toString() {

return "Model: " + this.type.toString() + ", " + this.modelType + ", Manufacturer: " + this.manufacturer + ", Number: " + this.number + ", Route: " + Arrays.toString(this.route) + ", Capacity: " + this.capacity + ", Driver: " + this.driver + ", Stopped at: " + this.stopped;

}

}

*Додаток В*Машинний лістинг файлу **Dog.java**

package Classes;

import Classes.Puppy;

import javax.swing.\*;

public class Testing {

public static void StartTest(String puppyName){

// Створення об'єкта класу Цуценя

Puppy TestMYpuppy = new Puppy(puppyName);

Actions(TestMYpuppy);

Diff\_Constructors();

}

private static void Actions(Puppy object)

{

// Виклик методів для демонстрації

object.askName();

object.makeSound();

object.jump();

object.run();

object.bite();

}

public static void Diff\_Constructors(){

// Створення об'єкта собаки з використанням різних значень статі

Puppy p1 = new Puppy("Бобік", 3, "brown", Gender.MALE);

Puppy p2 = new Puppy("Ляля", 2);

System.out.println(p1.getInfo());

System.out.println(p2.getInfo());

}

public static void AllTest() {

//перевірка імені

try {

System.out.println("Правильне ім'я конструктора");

Puppy a = new Puppy("ПравильнийЦуцик");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Мало літер ім'я конструктора");

Puppy a = new Puppy("По");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Багато літер ім'я конструктора - 24шт");

Puppy a = new Puppy("ПоПоПоПоПоПоПоПоПоПоПоПо");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Не літери ім'я конструктора");

Puppy a = new Puppy("12\_34");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

//перевірка віку

try {

System.out.println("Правильний вік");

Puppy a = new Puppy("Мінік",2);

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Вік нуль");

Puppy a = new Puppy("Мінік",0);

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Вік меньше нуля");

Puppy a = new Puppy("Мінік",-2);

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

}

}

*Додаток Г*Машинний лістинг файлу **Gender.java**

package Classes;

// Оголошення перерахування для статі тварини

public enum Gender {

MALE,

FEMALE,

UNKNOWN;

}

*Додаток Ґ*Машинний лістинг файлу **Puppy.java**

package Classes;

//властивості та поведінкa цуценяти.

public class Puppy extends Dog {

private String color;

public Puppy(String name)

{

super(name);

}

// Конструктор з параметрами ім'я, вік, колір та стать

public Puppy(String name, int age, String color, Gender gender) {

super(name, age, gender);

this.color = color;

}

// Конструктор з параметрами ім'я та вік (колір і стать встановлюються за замовчуванням)

public Puppy(String name, int age) {

super(name, age, Gender.UNKNOWN);

this.color = "Невідомий";

}

// Метод для отримання інформації про ім'я та колір цуценяти

public String getInfo() {

String result = ("Цуценя має ім'я: " + getName() +", йому "+getAge()+" років, "+" колір: " + color + " стать:"+getGender().toString());

return result;

}

public void askName() {

System.out.println( "Цуценя має ім'я: " + super.getName());

}

@Override

public void makeSound() {

System.out.println(super.getName() +" скиглить");

}

}

*Додаток Д*Машинний лістинг файлу **Testing.java**

package Classes;

import Classes.Puppy;

import javax.swing.\*;

public class Testing {

public static void StartTest(String puppyName){

// Створення об'єкта класу Цуценя

Puppy TestMYpuppy = new Puppy(puppyName);

Actions(TestMYpuppy);

Diff\_Constructors();

}

private static void Actions(Puppy object)

{

// Виклик методів для демонстрації

object.askName();

object.makeSound();

object.jump();

object.run();

object.bite();

}

public static void Diff\_Constructors(){

// Створення об'єкта собаки з використанням різних значень статі

Puppy p1 = new Puppy("Бобік", 3, "brown", Gender.MALE);

Puppy p2 = new Puppy("Ляля", 2);

System.out.println(p1.getInfo());

System.out.println(p2.getInfo());

}

public static void AllTest() {

//перевірка імені

try {

System.out.println("Правильне ім'я конструктора");

Puppy a = new Puppy("ПравильнийЦуцик");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Мало літер ім'я конструктора");

Puppy a = new Puppy("По");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Багато літер ім'я конструктора - 24шт");

Puppy a = new Puppy("ПоПоПоПоПоПоПоПоПоПоПоПо");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Не літери ім'я конструктора");

Puppy a = new Puppy("12\_34");

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

//перевірка віку

try {

System.out.println("Правильний вік");

Puppy a = new Puppy("Мінік",2);

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Вік нуль");

Puppy a = new Puppy("Мінік",0);

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

try {

System.out.println("Вік меньше нуля");

Puppy a = new Puppy("Мінік",-2);

System.out.println(a.getInfo());

System.out.println();

} catch (IllegalArgumentException e) {

System.out.println("Помилка: " + e.getMessage());

System.out.println();

}

}

}