МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Прктична робота № 1**

з дисципліни « Web-програмування мовою PHP »

*назва дисципліни*

на тему: «ЗНАЙОМСТВО З PHP»

Виконав: студент 3 курсу групи № 632п

освітньої програми

121 інженерія програмного забезпечення

(шифр і назва ОП)

Зайченко Ярослав Ігорович

(прізвище й ініціали студента)

Прийняв: старший викладач

Дем'яненко Владислав Анатолійович

(посада, науковий ступінь, прізвище й ініціали)

Кількість балів:

Харків – 2024

**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 роботи з установки й роботі з одним з основних засобів розробки додатків на мові програмування PHP.

Постановка завдання

Навчитися встановлювати і працювати з середовищем розробки додатків на мові сценаріїв PHP (Середовище на вибір студента – Eclipse PHP).

Порядок виконання роботи

1. Прочитати теоретичний матеріал
2. Вивчити процес інсталювання EclipsePHP/PHPStrom або іншого середовища
3. Навчитися створювати проект в обраному середовищі
4. Навчитися запускати проект PHP
5. Навчитися користуватися коментарями.
6. Навчитися створювати прості програми мовою PHP
7. Зробити висновки по роботі.

Завдання для варіанту 10: Створити об'єкт класу Цуценя, використовуючи класи Тварина, Собака. Методи: вивести на консоль ім'я, подати голос, стрибати, бігати, кусати.

Прочитати теоретичний матеріал

Зміст звіту

1. Постановка завдання.
2. Встановлення завдань, які вирішуються в лабораторній̆ роботі
3. Описати особливості процесу створення і запуску додатка.
4. Коротко описати лексеми і висновок інформації на екран.
5. Висновки по роботі.

Теоретичні відомомості

Для створення програм на мові програмування PHP нам буде потрібно текстовий редактор. Найбільш популярним на сьогоднішній день є програма Notepad ++.

Перейдемо до раніше створеного каталогу ~\localhost, який буде зберігати всі документи сайту. Створимо текстовий файл і назвемо його index.html. Відкриємо його в текстовому редакторі і додамо в нього наступний код:

<!DOCTYPE html>  
<html>  
<head>  
<title>Перший сайт на PHP</title> <meta charset="utf-8">

</head>  
<body>  
<h2>Введіть ваші дані:</h2>  
<form action="display.php" method="POST">  
<p>Введіть ім`я: <input type="text" name="firstname" /></p> <p>Введіть прізвище: <input type="text" name="lastname" /></p> <input type="submit" value="Відправити ">  
</form>  
</body>  
</html>

Код html містить форму з двома текстовими полями. При натисканні на кнопку дані цієї форми відсилаються скрипту display.php, так як він вказаний в атрибуті action.

Тепер створимо цей скрипт, який буде обробляти дані. Додамо в папку ~\localhost новий текстовий файл. Перейменуємо його у display.php. За замовчуванням файли програм на php мають розширення .php. Отже, додамо в файл display.php наступний код:

<!DOCTYPE html>

<html>

<head>

<title>Перший сайт на PHP </title>

<meta charset="utf-8">

</head>

<body>

<?php

name = $\_POST["firstname"];  
$surname = $\_POST["lastname"];  
echo "Ваше ім`я: <b>".$name . " " . $surname . "</b>";

?>

</body>

</html>

Тут вже в розмітці html йдуть вкраплення коду PHP. Для додавання виразів PHP на сторінку використовуються теги <? Php .....?>, Між якими йдуть інструкції на мові PHP. У коді php ми отримуємо дані форми і виводимо їх на сторінку.

Кожне окреме вираз PHP має завершуватися крапкою з комою. В даному випадку у нас три вирази. Два з них отримують передані дані форми, наприклад, $ name = $ \_POST [ "firstname"] ;.

$ Name – це змінна, яка буде зберігати деяке значення. Всі змінні в PHP передує знаком $. І так як форма на сторінці index.html використовує для відправки метод POST, то за допомогою виразу $ \_POST [ "firstname"] ми можемо отримати значення, яке було введено в текстове поле з атрибутом name = "firstname". І це значення потрапляє в змінну $ name.

За допомогою оператора echo можна вивести на сторінку будь-яке значення або текст, які йдуть після оператора. В даному випадку (echo "Ваше ім'я: <b>". $ Name. "". $ Surname. "</ B>") за допомогою знака точки текст в лапках з'єднується зі значеннями змінних $ name і $ surname і виводиться на сторінку .

Основи синтаксису

При створенні першої програми на PHP вже були порушені деякі основні принципи створення скриптів на мові PHP. Тепер, розглянемо їх більш детально.

Програма або скрипт на PHP, як правило, знаходиться в файлі розширенням .php. Хоча розробники можуть також вставляти код php і в файли з розширеннями .html / .htm.

Коли користувач звертається до скрипту в адресному рядку браузера, набираючи, наприклад, http: // localhost: 8080 / display.php, то веб-сервер передає його інтерпретатора PHP. Потім інтерпретатор обробляє код і генерує на його основі html-розмітку. І потім згенерований html-код відправляється користувачеві.

Документ PHP може містити як розмітку html, так і код на мові php. Для переходу від розмітки html до коду php використовуються теги <?php та ?>, Між якими йде код php. Дані теги служать зазначенням інтерпретатора, що їх вміст треба інтерпретувати як код php, а не розмітку html.

Також можна використовувати коротку версію тегів: <? і ?>. Для цього в файлі php.ini треба змінити значення short\_open\_tag = Off на short\_open\_tag = On.

Розглянемо найпростіший скрипт на php:

<html>

<head>

<title>Веб-сайт</title>

</head>  
<body>  
<?php  
echo "<p>Привет мир!</p>"; echo "2 + 2 = " . (2+2); ?>

</body>

</html>

Після обробки файлу інтерпретатор сформує наступну розмітку:

<html>  
<head> <title>Веб-сайт</title> </head>  
<body>  
<p>Привет мир!</p> 2+2= 4  
</body>  
</html>

Тут використані дві інструкції echo "<p> Привіт світ! </ p>" і echo "2 + 2 =". (2 + 2), який виводять певне значення на сторінку. Кожна окрема інструкція в PHP завершується крапкою з комою.

Коментарі

При створенні веб-сайту ми можемо використовувати коментарі. Наприклад, ми можемо прокоментувати будь-яку дію, щоб згодом мати уявлення, що цей код робить:

<?php

echo "<p>Привет мир!</p>"; // вывод сообщения

// echo "2 + 2 = " . (2+2);  
?>

Знак // передує однорядковий коментар, і все що йде після цього знака на одному рядку, буде вважатися коментарем і не буде виконуватися інтерпретатором. При обробці інтерпретатор просто буде пропускати коментарі.

Якщо нам треба закомментувати кілька рядків, то ми можемо використовувати багаторядковий коментар / \* текст коментаря \* /:

<? php  
echo "<p> Привіт світ! </ p>"; // виведення повідомлення  
/\*  
багатостроковий коментарвисновок результату арифметичного виразу  
echo "2 + 2 =". (2 + 2);  
\*/  
?>

Всі рядки всередині коментаря також не обробляються інтерпретатором.

Змінні

Як і в багатьох мовах програмування, в PHP є змінні. Змінні зберігають окремі значення, які можна використовувати у виразах на PHP. Для позначення змінних використовується знак долара $. Наприклад:

<?php  
$a = 10;  
echo $a;  
?>

Тут визначена змінна, яка буде зберігати число 10. Присвоєння значення відбувається за допомогою знака рівності =.

Можна привласнювати значення іншої змінної:

$a = 10;

$b=$a;

echo $b;

PHP є чутливою до регістру мовою, а значить, змінні $ counter і $ Counter представлятимуть дві різні змінні.

Також при найменуванні змінних нам треба враховувати наступні правила:

* Імена змінних повинні починатися з алфавітного символу або з підкреслення
* Імена змінних можуть містити тільки символи: a-z, A-Z, 0-9, і знак підкреслення
* Імена змінних не повинні включати в себе прогалини

Перевірка існування змінної. оператор isset

Якщо змінна оголошена, але їй спочатку не присвоєно ніякого значення

(інакше кажучи вона не ініциалізованна), то нам буде проблематично її використовувати. наприклад:

<?php  
$a;

echo $a;

?>

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

**Клас Тварина**:

Властивості:

* Ім'я (name)

Поведінка:

* Виведення імені на консоль (void printName())
* Подати голос (void makeSound())

**Клас Собака** (наслідується від класу Тварина):

Поведінка:

* Стрибати (void jump())
* Бігати (void run())
* Кусати (void bite())

**Клас Цуценя** (наслідується від класу Собака):

* Зміна реалізації методу для подання звуку
* Метод створення рядка з повною інформацією про об’єкт

**Структурні відносини між спроектованими класами за допомогою діаграми класів UML.**

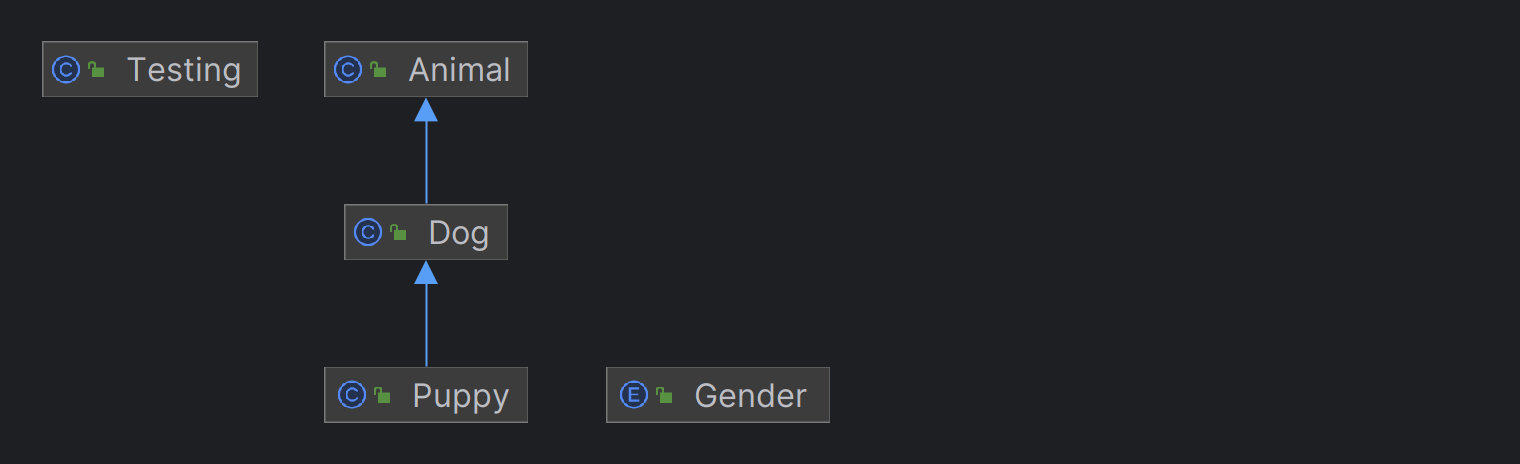


Рисунок 1 – згенерована діаграма у середовищі розробки

Реалізувати програмний код класів на мові Java.

При написанні коду були виконані умови: створення в класах 2 конструкторів, поля класів були створені з модифікатором **private**. Також для закритих полів створено методи гетерів та сетерів (**get/set**), у кожному класі присутні декілька різних методів, які моделюють поведінку об’єктів та змінюють їх стан. Фрагменти фунціоналу коду описані коментарями. Програмний код класів можна переглянути в додатках А-Д.

Клас для тестування

Створений клас демонструє створення різних екземплярів класу **Puppy** з використанням різноманітних конструкторів, та виклику методів класів.

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();

}

}

}

Компіляція коду

Для того, щоб відкомпілювати розроблені класи програми за допомогою компілятора javac.exe необхідно відкрити командний рядок (cmd) і використати команду **javac**, яка поставляється разом з JDK (Java Development Kit). Файли проєкту зберігаються в папці src, ми можемо скомпілювати їх такою командою:

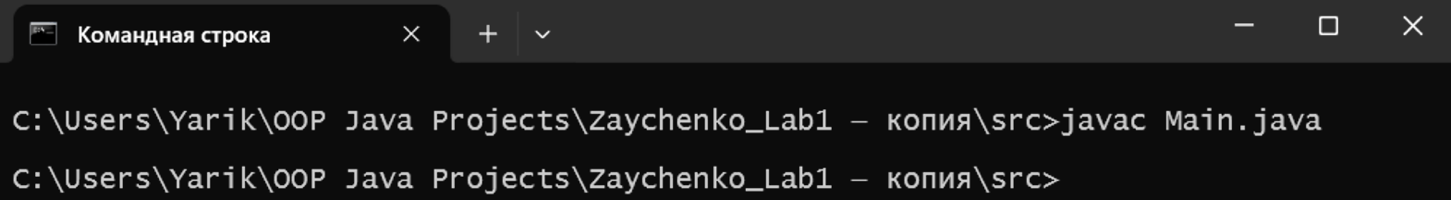
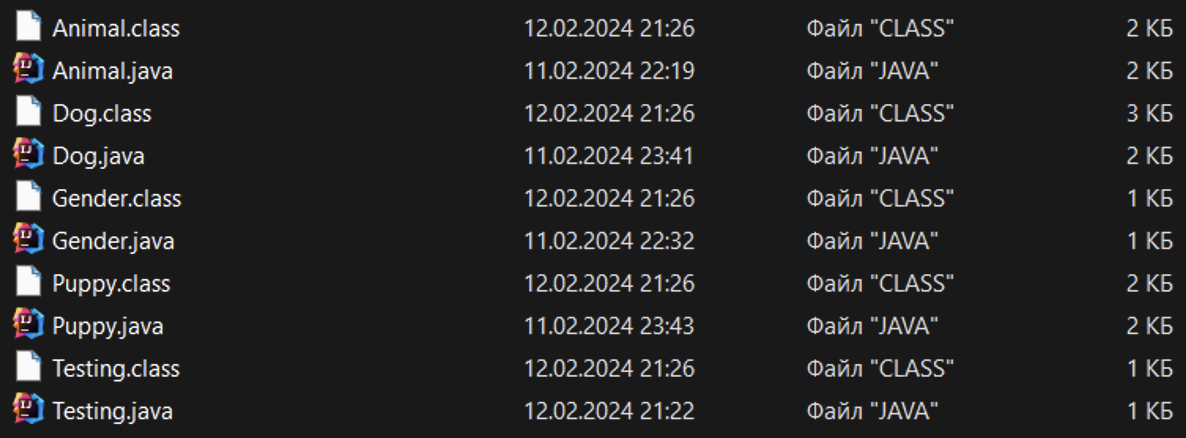


Рисунок 2 – використання команди компіляції файлів \*.java



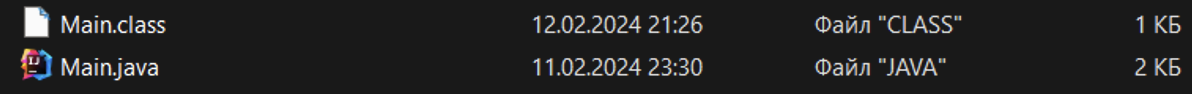


Рисунок 3 – результат використання команди для компіляції

Тестування програми

Тестувати будемо правильними та неправильними значеннями. Всі тестування були виконані за допомогою окремого класу, кожен тест підписаний. Результат на рисунку 4.

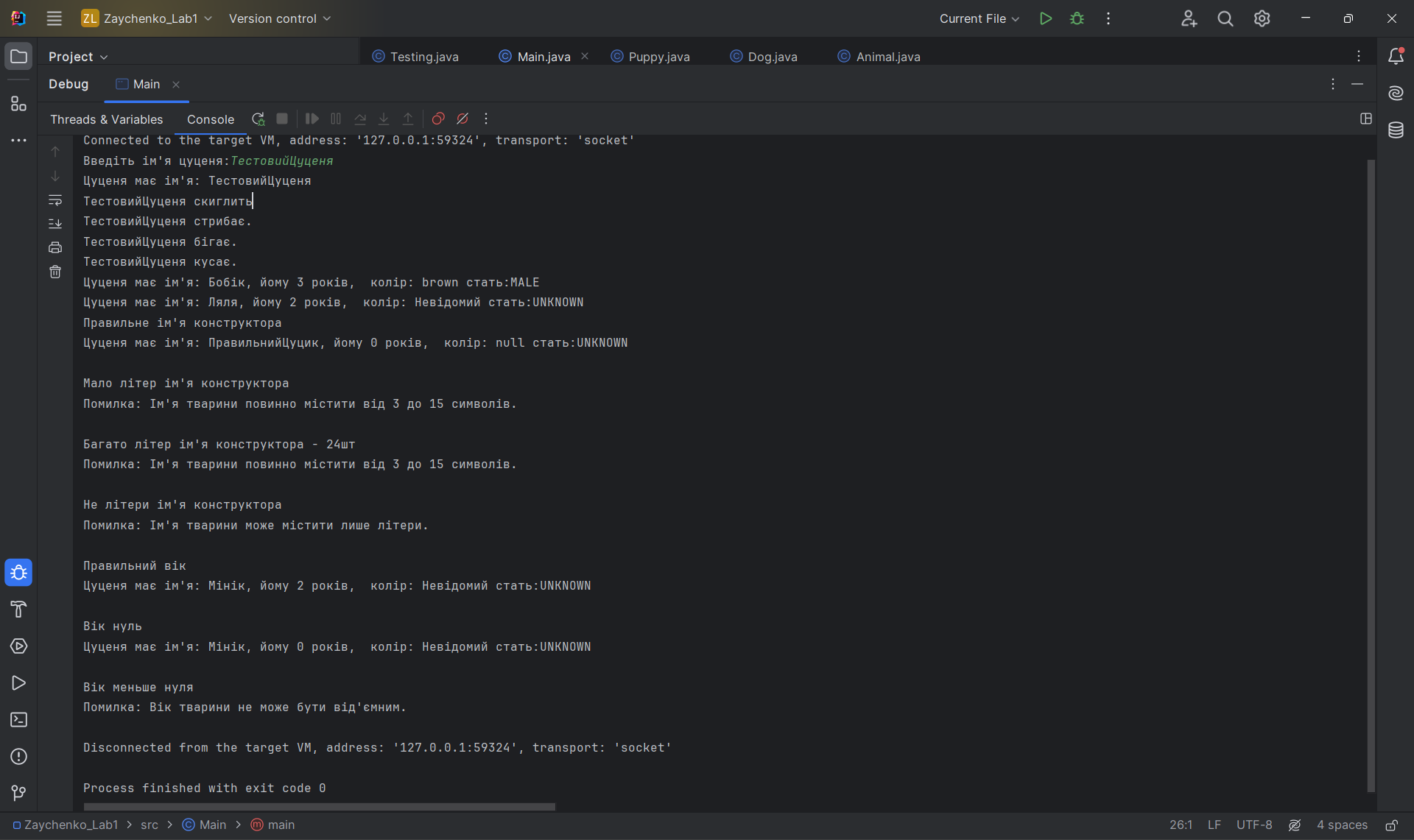


Рисунок 4 – виконання всіх тестових випадків

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.

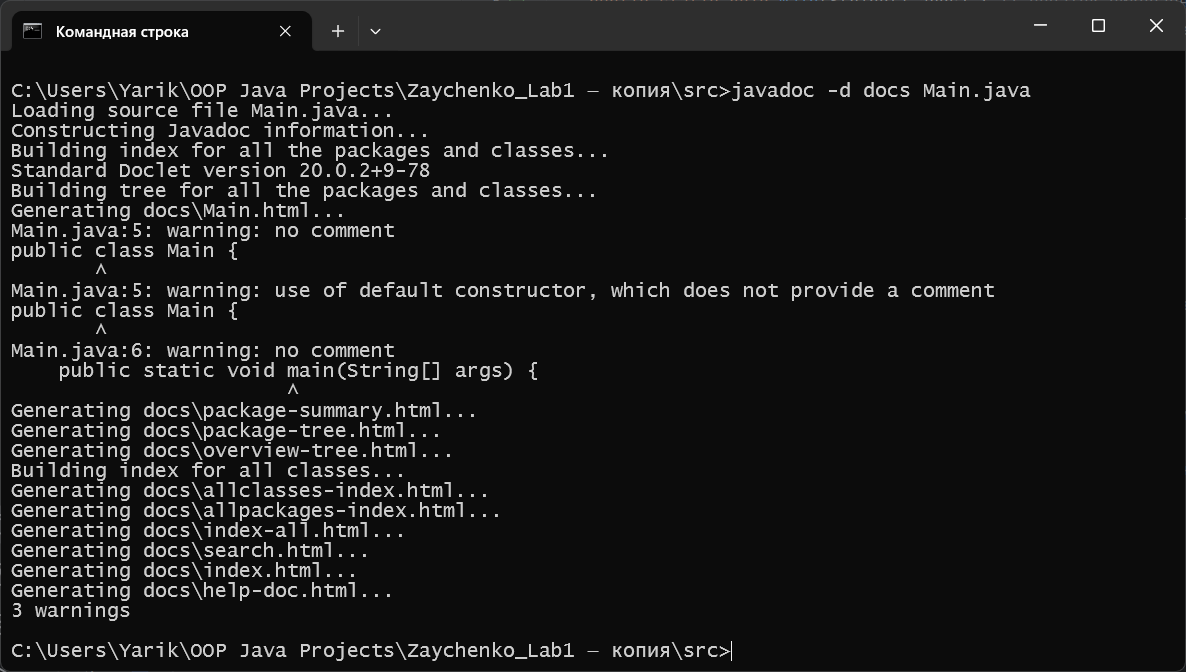


Рисунок 5 – створення документації додатку стандартними командами JRE

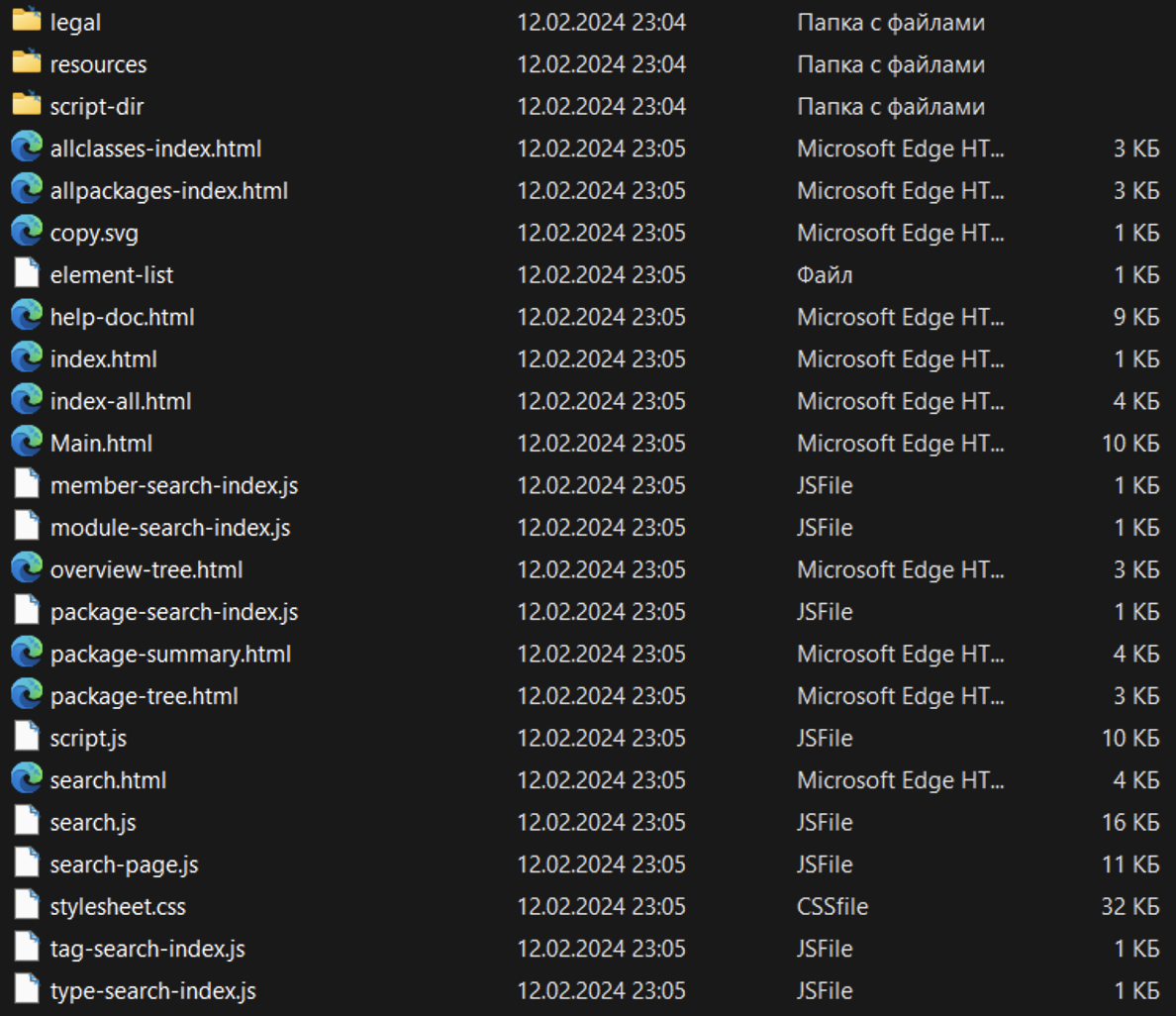


Рисунок 6 – отримані файли документації на додаток

Висновок

У ході виконання лабораторної роботи було розроблено програмний додаток мовою Java для моделювання властивостей, поведінки та взаємодії об'єктів. Було використано концепти об'єктно-орієнтованого програмування, такі як класи, конструктори, методи, геттери та сеттери.

Під час розробки програмного коду було враховано постановку завдання, в якій вказано необхідність реалізації конструкторів та методів для зміни поведінки та стану об'єктів. Кожен клас був ретельно розроблений з урахуванням його властивостей та поведінки.

Для підтримки зрозумілості та документації коду, було додано коментарі та дескриптори документування для опису класів, конструкторів, методів та полів. Це сприяє зручнішому розумінню функціоналу кожного класу та його взаємодії з іншими.

Також було реалізовано тестуючий клас для демонстрації створення об'єктів класів та виклику методів, що змінюють їх властивості та поведінку. Цей етап дозволив перевірити коректність роботи розробленого додатку та виявити та виправити можливі помилки.

Крім того, було створено документацію на розроблений додаток за допомогою утиліти javadoc.exe. Це дозволяє отримати зрозумілу та структуровану інформацію про функціонал програми та класів, що її складають.

Отже, виконання лабораторної роботи над даним проектом дало можливість отримати практичні навички з проектування та реалізації об'єктно-орієнтованих програмних додатків мовою Java.

Контрольні питання

*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 **Animal.java**

package Classes;

//загальні властивості тварин.

public class Animal {

private String name;

private int Age;

// конструктор з параметром ім'я.

public Animal(String NAME) {

setName(NAME);

}

public Animal(String NAME, int AGE) {

setName(NAME);

setAge(AGE);

}

// отримання імені тварини.

public String getName() {

return this.name;

}

public int getAge() {

return this.Age;

}

// Метод-сеттер для імені тварини з перевіркою на відповідність заданим критеріям

public void setName(String name) throws IllegalArgumentException {

// Розбиваємо рядок на символи і перевіряємо кожен символ

for (char c : name.toCharArray()) {

if (!Character.isLetter(c)) {

throw new IllegalArgumentException("Ім'я тварини може містити лише літери.");

}

}

if (name.length() < 3 || name.length() > 20) {

throw new IllegalArgumentException("Ім'я тварини повинно містити від 3 до 15 символів.");

}

this.name = name;

}

public void setAge(int AGE) throws IllegalArgumentException {

if (AGE < 0) throw new IllegalArgumentException("Вік тварини не може бути від'ємним.");

this.Age=AGE;

}

// метод звуку

public void makeSound() {

System.out.println("Тварина видає звук.");

}

}

*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();

}

}

}