Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра цифрових технологій в енергетиці

ЗВІТ   
з виконання лабораторної роботи №5  
з дисципліни «Технології Java конструювання програмного забезпечення»

«**КЛАСИ І ЇХНЄ ДОКУМЕНТУВАННЯ**»

Варіант 17

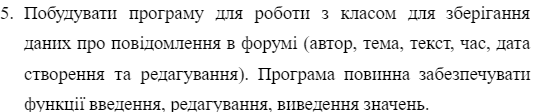
Виконав: студент групи ТР-23

Ровний Г.О.

Перевірив: доцент, к.ф.-м.н.  
Тарнавський Ю.А.

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

# Завдання 1



Інтерфейс:

package org.example;  
  
*/\*\*  
 \* Інтерфейс для роботи з повідомленнями на форумі.  
 \*/*public interface Task5Interface {  
 void setAuthor(String author);  
 void setTopic(String topic);  
 void setText(String text);  
 void setCreationDateTime(String dateTime);  
 void setEditDateTime(String dateTime);  
 String getAuthor();  
 String getTopic();  
 String getText();  
 String getCreationDateTime();  
 String getEditDateTime();  
}

Клас:

package org.example;  
  
import java.util.Objects;  
  
*/\*\*  
 \* Клас для зберігання даних про повідомлення у форумі.  
 \*/*public class Task5Class implements Task5Interface {  
 private String author;  
 private String topic;  
 private String text;  
 private String creationDateTime;  
 private String editDateTime;  
  
 public Task5Class(String author, String topic, String text, String creationDateTime) {  
 this.author = author;  
 this.topic = topic;  
 this.text = text;  
 this.creationDateTime = creationDateTime;  
 this.editDateTime = creationDateTime;  
 }  
  
 @Override  
 public void setAuthor(String author) {  
 this.author = author;  
 }  
  
 @Override  
 public void setTopic(String topic) {  
 this.topic = topic;  
 }  
  
 @Override  
 public void setText(String text) {  
 this.text = text;  
 }  
  
 @Override  
 public void setCreationDateTime(String dateTime) {  
 this.creationDateTime = dateTime;  
 }  
  
 @Override  
 public void setEditDateTime(String dateTime) {  
 this.editDateTime = dateTime;  
 }  
  
 @Override  
 public String getAuthor() {  
 return author;  
 }  
  
 @Override  
 public String getTopic() {  
 return topic;  
 }  
  
 @Override  
 public String getText() {  
 return text;  
 }  
  
 @Override  
 public String getCreationDateTime() {  
 return creationDateTime;  
 }  
  
 @Override  
 public String getEditDateTime() {  
 return editDateTime;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Task5Class that = (Task5Class) o;  
 return Objects.*equals*(author, that.author) &&  
 Objects.*equals*(topic, that.topic) &&  
 Objects.*equals*(text, that.text) &&  
 Objects.*equals*(creationDateTime, that.creationDateTime) &&  
 Objects.*equals*(editDateTime, that.editDateTime);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.*hash*(author, topic, text, creationDateTime, editDateTime);  
 }  
  
 @Override  
 public String toString() {  
 return "ForumMessage{" +  
 "author='" + author + '\'' +  
 ", topic='" + topic + '\'' +  
 ", text='" + text + '\'' +  
 ", creationDateTime='" + creationDateTime + '\'' +  
 ", editDateTime='" + editDateTime + '\'' +  
 '}';  
 }  
  
 public static void main(String[] args) {  
 Task5Class message = new Task5Class(  
 "John Doe",  
 "Sample Topic",  
 "This is a sample forum message.",  
 "2024-11-13 10:00"  
  
 );  
  
 // Вивожу стартові дані  
 System.*out*.println("Original Message:");  
 System.*out*.println(message);  
  
 // Редагування тексту  
 message.setText("This is an updated forum message.");  
 message.setEditDateTime("2024-11-27 12:00");  
  
 // Зміна автора  
 message.setAuthor("Tarnavskyi");  
 System.*out*.println("\nAfter Changing Author:");  
 System.*out*.println(message);  
  
 // Зміна теми  
 message.setTopic("Updated Topic");  
 System.*out*.println("\nAfter Changing Topic:");  
 System.*out*.println(message);  
  
  
 //  
 System.*out*.println("\nUpdated Message:");  
 System.*out*.println(message);  
 }  
}

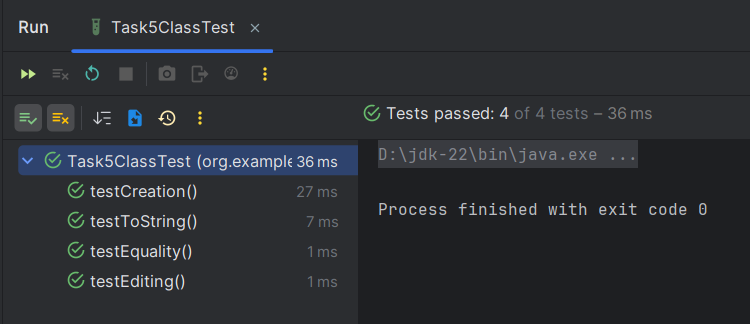
і скріншот:



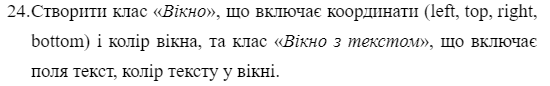
Тести:

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class Task5ClassTest {  
  
 @Test  
 void testCreation() {  
 Task5Class message = new Task5Class("Grisha Rovnyi", "Welcome", "Hello, world!", "2024-11-27 10:00");  
 *assertEquals*("Grisha Rovnyi", message.getAuthor());  
 *assertEquals*("Welcome", message.getTopic());  
 *assertEquals*("Hello, world!", message.getText());  
 *assertEquals*("2024-11-13 10:00", message.getCreationDateTime());  
 }  
  
 @Test  
 void testEditing() {  
 Task5Class message = new Task5Class("Grisha Rovnyi", "Welcome", "Hello, world!", "2024-11-27 10:00");  
 message.setText("Updated text");  
 message.setEditDateTime("2024-11-27 12:00");  
 *assertEquals*("Updated text", message.getText());  
 *assertEquals*("2024-11-27 12:00", message.getEditDateTime());  
 }  
  
 @Test  
 void testEquality() {  
 Task5Class message1 = new Task5Class("Grisha Rovnyi", "Topic", "Text", "2024-11-27 10:00");  
 Task5Class message2 = new Task5Class("Grisha Rovnyi", "Topic", "Text", "2024-11-27 10:00");  
 *assertEquals*(message1, message2);  
 }  
  
 @Test  
 void testToString() {  
 Task5Class message = new Task5Class("John", "Topic", "Text", "2024-11-13 10:00");  
 String expected = "ForumMessage{author='John', topic='Topic', text='Text', creationDateTime='2024-11-13 10:00', editDateTime='2024-11-13 10:00'}";  
 *assertEquals*(expected, message.toString());  
 }  
}

Скріншот:



# Завдання 2



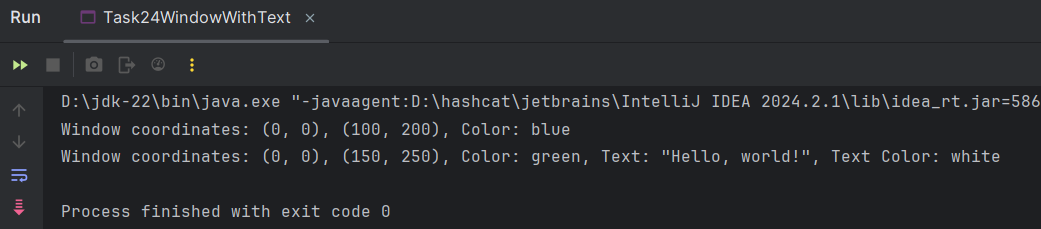
Клас Task24Window:

package org.example;  
  
public class Task24Window {  
 private int left;  
 private int top;  
 private int right;  
 private int bottom;  
 private String color;  
  
 // Конструктор  
 public Task24Window(int left, int top, int right, int bottom, String color) {  
 this.left = left;  
 this.top = top;  
 this.right = right;  
 this.bottom = bottom;  
 this.color = color;  
 }  
  
 // Гетери та сетери для кожного поля  
 public int getLeft() {  
 return left;  
 }  
  
 public void setLeft(int left) {  
 this.left = left;  
 }  
  
 public int getTop() {  
 return top;  
 }  
  
 public void setTop(int top) {  
 this.top = top;  
 }  
  
 public int getRight() {  
 return right;  
 }  
  
 public void setRight(int right) {  
 this.right = right;  
 }  
  
 public int getBottom() {  
 return bottom;  
 }  
  
 public void setBottom(int bottom) {  
 this.bottom = bottom;  
 }  
  
 public String getColor() {  
 return color;  
 }  
  
 public void setColor(String color) {  
 this.color = color;  
 }  
  
 public String displayInfo() {  
 return "Window coordinates: (" + left + ", " + top + "), (" + right + ", " + bottom + "), Color: " + color;  
 }  
}

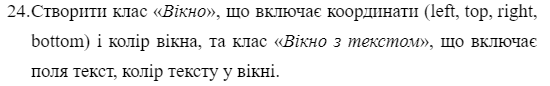
Клас Task24WindowWithText:

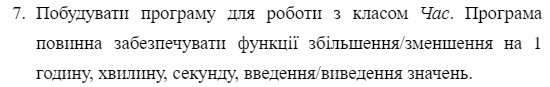
package org.example;  
  
public class Task24WindowWithText extends Task24Window {  
 private String text;  
 private String textColor;  
  
 // Конструктор  
 public Task24WindowWithText(int left, int top, int right, int bottom, String color, String text, String textColor) {  
 super(left, top, right, bottom, color); // Викликаємо конструктор батьківського класу  
 this.text = text;  
 this.textColor = textColor;  
 }  
  
 public String getText() {  
 return text;  
 }  
  
 public void setText(String text) {  
 this.text = text;  
 }  
  
 public String getTextColor() {  
 return textColor;  
 }  
  
 public void setTextColor(String textColor) {  
 this.textColor = textColor;  
 }  
  
 // Перевизначений метод для відображення інформації про вікно з текстом  
 @Override  
 public String displayInfo() {  
 return super.displayInfo() + ", Text: \"" + text + "\", Text Color: " + textColor;  
 }  
 public static void main(String[] args) {  
 // об'єкт класу Task24Window  
 Task24Window window = new Task24Window(0, 0, 100, 200, "blue");  
 System.*out*.println(window.displayInfo());  
  
 //об'єкт класу Task24WindowWithText  
 Task24WindowWithText windowWithText = new Task24WindowWithText(0, 0, 150, 250, "green", "Hello, world!", "white");  
 System.*out*.println(windowWithText.displayInfo());  
 }  
}

Скріншот як працює програма:



# Завдання 3





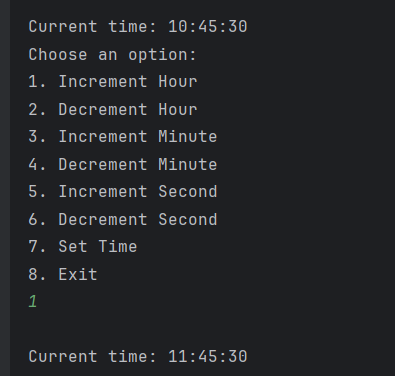
Інтерфейс:

package org.example;  
  
public interface Task7Interface {  
 void incrementHours();  
 void decrementHours();  
 void incrementMinutes();  
 void decrementMinutes();  
 void incrementSeconds();  
 void decrementSeconds();  
 void setTime(int hours, int minutes, int seconds);  
 String displayTime();  
}

Клас:

package org.example;  
import java.util.Scanner;  
  
public class Task7Class implements Task7Interface {  
 private int hours;  
 private int minutes;  
 private int seconds;  
  
 public Task7Class(int hours, int minutes, int seconds) {  
 this.hours = hours;  
 this.minutes = minutes;  
 this.seconds = seconds;  
 }  
  
 @Override  
 public void incrementHours() {  
 hours = (hours + 1) % 24;  
 }  
  
 @Override  
 public void decrementHours() {  
 hours = (hours - 1 + 24) % 24;  
 }  
  
 @Override  
 public void incrementMinutes() {  
 minutes = (minutes + 1) % 60;  
 if (minutes == 0) {  
 incrementHours();  
 }  
 }  
  
 @Override  
 public void decrementMinutes() {  
 minutes = (minutes - 1 + 60) % 60;  
 if (minutes == 59) {  
 decrementHours(); // Якщо хвилини стали 59 зменшуємо годину  
 }  
 }  
  
 @Override  
 public void incrementSeconds() {  
 seconds = (seconds + 1) % 60;  
 if (seconds == 0) {  
 incrementMinutes();  
 }  
 }  
  
 @Override  
 public void decrementSeconds() {  
 seconds = (seconds - 1 + 60) % 60;  
 if (seconds == 59) {  
 decrementMinutes();  
 }  
 }  
  
 @Override  
 public void setTime(int hours, int minutes, int seconds) {  
 this.hours = hours;  
 this.minutes = minutes;  
 this.seconds = seconds;  
 }  
  
 @Override  
 public String displayTime() {  
 return String.*format*("%02d:%02d:%02d", hours, minutes, seconds); // Форматування  
 }  
  
 public static void main(String[] args) {  
 Task7Class time = new Task7Class(10, 45, 30); // Початковий час  
 Scanner scanner = new Scanner(System.*in*);  
 boolean running = true;  
  
 while (running) {  
 System.*out*.println("\nCurrent time: " + time.displayTime());  
 System.*out*.println("Choose an option:");  
 System.*out*.println("1. Increment Hour");  
 System.*out*.println("2. Decrement Hour");  
 System.*out*.println("3. Increment Minute");  
 System.*out*.println("4. Decrement Minute");  
 System.*out*.println("5. Increment Second");  
 System.*out*.println("6. Decrement Second");  
 System.*out*.println("7. Set Time");  
 System.*out*.println("8. Exit");  
  
 int choice = scanner.nextInt(); // Ввод користувача  
  
 switch (choice) {  
 case 1:  
 time.incrementHours();  
 break;  
 case 2:  
 time.decrementHours();  
 break;  
 case 3:  
 time.incrementMinutes();  
 break;  
 case 4:  
 time.decrementMinutes();  
 break;  
 case 5:  
 time.incrementSeconds();  
 break;  
 case 6:  
 time.decrementSeconds();  
 break;  
 case 7:  
 // Введення нового часу  
 System.*out*.println("Enter new time (hours, minutes, seconds):");  
 int newHours = scanner.nextInt();  
 int newMinutes = scanner.nextInt();  
 int newSeconds = scanner.nextInt();  
 time.setTime(newHours, newMinutes, newSeconds);  
 break;  
 case 8:  
 running = false;  
 break;  
 default:  
 System.*out*.println("Invalid option, please try again.");  
 }  
 }  
 System.*out*.println("Program exited.");  
 scanner.close();  
 }  
}

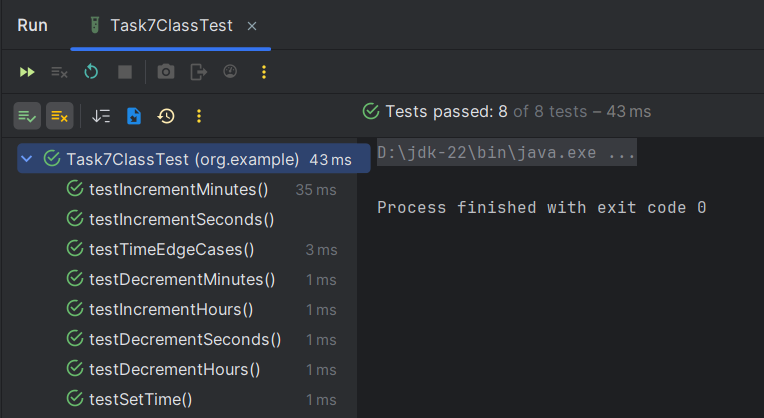
Скріншот



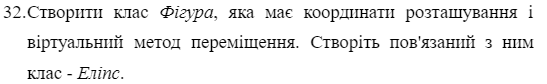
Тести:

package org.example;  
import static org.junit.jupiter.api.Assertions.\*;  
import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.Test;  
  
public class Task7ClassTest {  
  
 private Task7Class time;  
  
 @BeforeEach  
 public void setUp() {  
 // 10:45:30  
 time = new Task7Class(10, 45, 30);  
 }  
  
 @Test  
 public void testIncrementHours() {  
 time.incrementHours(); // Збільшуємо годину на 1  
 *assertEquals*("11:45:30", time.displayTime(), "Hour should be incremented by 1.");  
 }  
  
 @Test  
 public void testDecrementHours() {  
 time.decrementHours(); // Зменшуємо годину на 1  
 *assertEquals*("09:45:30", time.displayTime(), "Hour should be decremented by 1.");  
 }  
  
 @Test  
 public void testIncrementMinutes() {  
 time.incrementMinutes(); // Збільшуємо хвилину на 1  
 *assertEquals*("10:46:30", time.displayTime(), "Minute should be incremented by 1.");  
 }  
  
 @Test  
 public void testDecrementMinutes() {  
 time.decrementMinutes(); // Зменшуємо хвилину на 1  
 *assertEquals*("10:44:30", time.displayTime(), "Minute should be decremented by 1.");  
 }  
  
 @Test  
 public void testIncrementSeconds() {  
 time.incrementSeconds(); // Збільшуємо секунду на 1  
 *assertEquals*("10:45:31", time.displayTime(), "Second should be incremented by 1.");  
 }  
  
 @Test  
 public void testDecrementSeconds() {  
 time.decrementSeconds(); // Зменшуємо секунду на 1  
 *assertEquals*("10:45:29", time.displayTime(), "Second should be decremented by 1.");  
 }  
  
 @Test  
 public void testSetTime() {  
 time.setTime(12, 0, 0); // Встановлюємо новий час  
 *assertEquals*("12:00:00", time.displayTime(), "Time should be set to 12:00:00.");  
 }  
  
 @Test  
 public void testTimeEdgeCases() {  
 time.setTime(23, 59, 59); // Встановлюємо час на крайній межі  
 time.incrementSeconds(); // Збільшуємо секунду  
 *assertEquals*("00:00:00", time.displayTime(), "Time should wrap around to 00:00:00.");  
  
 time.setTime(0, 0, 0); // Встановлюємо час на мінімум  
 time.decrementSeconds(); // Зменшуємо секунду  
 *assertEquals*("23:59:59", time.displayTime(), "Time should wrap around to 23:59:59.");  
 }  
}

Скріншот тестів:



# Завдання 4



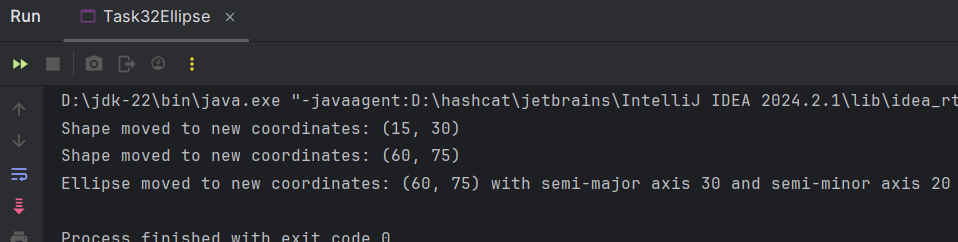
Клас Task32Shape:

package org.example;  
  
public class Task32Shape {  
 protected int x;  
 protected int y;  
  
 public Task32Shape(int x, int y) {  
 this.x = x;  
 this.y = y;  
 }  
  
 // Віртуальний метод переміщення (буде перевизначений в підкласах)  
 public void move(int deltaX, int deltaY) {  
 this.x += deltaX;  
 this.y += deltaY;  
 System.*out*.println("Shape moved to new coordinates: (" + x + ", " + y + ")");  
 }  
  
 public int getX() {  
 return x;  
 }  
  
 public int getY() {  
 return y;  
 }  
  
 public void setX(int x) {  
 this.x = x;  
 }  
  
 public void setY(int y) {  
 this.y = y;  
 }  
}

Клас Task32Ellipse:

package org.example;  
  
public class Task32Ellipse extends Task32Shape {  
 private int semiMajorAxis;  
 private int semiMinorAxis;  
  
 public Task32Ellipse(int x, int y, int semiMajorAxis, int semiMinorAxis) {  
 super(x, y);  
 this.semiMajorAxis = semiMajorAxis;  
 this.semiMinorAxis = semiMinorAxis;  
 }  
  
 // Метод для переміщення еліпса (перевизначаємо метод move)  
 @Override  
 public void move(int deltaX, int deltaY) {  
 super.move(deltaX, deltaY);  
 System.*out*.println("Ellipse moved to new coordinates: (" + x + ", " + y + ") with semi-major axis " + semiMajorAxis + " and semi-minor axis " + semiMinorAxis);  
 }  
  
 public int getSemiMajorAxis() {  
 return semiMajorAxis;  
 }  
  
 public int getSemiMinorAxis() {  
 return semiMinorAxis;  
 }  
  
 public void setSemiMajorAxis(int semiMajorAxis) {  
 this.semiMajorAxis = semiMajorAxis;  
 }  
  
 public void setSemiMinorAxis(int semiMinorAxis) {  
 this.semiMinorAxis = semiMinorAxis;  
 }  
 public static void main(String[] args) {  
 Task32Shape shape = new Task32Shape(10, 20);  
 shape.move(5, 10);  
  
 Task32Ellipse ellipse = new Task32Ellipse(50, 60, 30, 20);  
 ellipse.move(10, 15);  
 }  
}

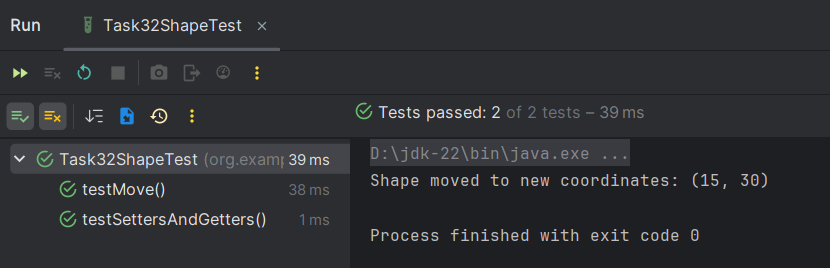
Скріншот як це працює +-



Тести для Task32Shape

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class Task32ShapeTest {  
  
 @Test  
 public void testMove() {  
 Task32Shape shape = new Task32Shape(10, 20);  
 shape.move(5, 10);  
 *assertEquals*(15, shape.getX());  
 *assertEquals*(30, shape.getY());  
 }  
  
 @Test  
 public void testSettersAndGetters() {  
 Task32Shape shape = new Task32Shape(10, 20);  
 shape.setX(25);  
 shape.setY(35);  
 *assertEquals*(25, shape.getX());  
 *assertEquals*(35, shape.getY());  
 }  
}

Скріншот



Тести для Task32Ellipse

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class Task32EllipseTest {  
  
 @Test  
 public void testMove() {  
 Task32Ellipse ellipse = new Task32Ellipse(50, 60, 30, 20);  
 ellipse.move(10, 15);  
 *assertEquals*(60, ellipse.getX());  
 *assertEquals*(75, ellipse.getY());  
 }  
  
 @Test  
 public void testSettersAndGetters() {  
 Task32Ellipse ellipse = new Task32Ellipse(50, 60, 30, 20);  
 ellipse.setSemiMajorAxis(40);  
 ellipse.setSemiMinorAxis(25);  
 *assertEquals*(40, ellipse.getSemiMajorAxis());  
 *assertEquals*(25, ellipse.getSemiMinorAxis());  
 }  
}

Скріншот:

