**Практичне завдання № 1.1**

**Приклад 1. Унаслідування класу*java.lang.Thread***

public class MyThread extends Thread {

public void run() {

System.out.println("Important job running in " + getName());

}

}

public class MyThreadDemo {

public static void main(String[] args) {

MyThread myThread = new MyThread();

myThread.start();

}

}

#### Приклад 2. Перевизначення методу run()

public class MyThread extends Thread {

public void run() {

System.out.println("Important job running in MyThread");

}

public void run(String s) {

System.out.println("String in run is " + s);

}

}

#### Приклад 3. Реалізація інтерфейсу java.lang.Runnable

public class MyRunnable implements Runnable {

public void run() {

System.out.println("Important job running");

}

}

public class MyRunnableDemo {

public static void main(String[] args) {

MyRunnable myRunnable = new MyRunnable();

Thread thread1 = new Thread(myRunnable);

Thread thread2 = new Thread(myRunnable);

Thread thread3 = new Thread(myRunnable);

thread1.start();

thread2.start();

thread3.start();

}

}

#### Приклад 4. Отримання назви потоку

public class NameRunnable implements Runnable {

public void run() {

System.out.println("NameRunnable running");

System.out.println("Run by "

+ Thread.currentThread().getName());

}

}

public class NameThreadDemo {

public static void main(String[] args) {

NameRunnable nr = new NameRunnable();

Thread tread1 = new Thread(nr);

tread1.setName("First Thread");

tread1.start();

Thread tread2 = new Thread(nr, "Second Thread");

tread2.start();

}

}

#### Приклад 5. Використання методу Thread.sleep()

public class SleepRunnable implements Runnable {

public void run() {

for (int x = 1; x < 4; x++) {

System.out.println("Run by "

+ Thread.currentThread().getName()

+ ", x is " + x);

try {

Thread.sleep(1000);

} catch (InterruptedException ex) {

ex.printStackTrace();

}

}

}

}

public class SleepRunnableDemo {

public static void main(String[] args) {

SleepRunnable sleepRunnable = new SleepRunnable();

Thread one = new Thread(sleepRunnable);

one.setName("Fred");

Thread two = new Thread(sleepRunnable);

two.setName("Lucy");

Thread three = new Thread(sleepRunnable);

three.setName("Ricky");

one.start();

two.start();

three.start();

}

}

#### Приклад 6. Використання методу Thread.join()

public class JoinRunnable extends Thread {

public JoinRunnable(String name) {

super(name);

}

@Override

public void run() {

String currentThreadName = Thread.currentThread().getName();

for (int i = 0; i < 10; i++) {

System.out.println(currentThreadName + " is running!" + i);

try {

Thread.sleep(500);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println(currentThreadName + " completed");

}

}

public class JoinDemo {

public static void main(String[] args) {

JoinRunnable a = new JoinRunnable("A");

JoinRunnable b = new JoinRunnable("B");

JoinRunnable c = new JoinRunnable("C");

a.start();

try {

a.join();

} catch (InterruptedException e) {

System.out.println(e.getMessage());

}

b.start();

c.start();

}

}

#### Приклад 7. Використання методу Thread.isAlive()

public class IsAliveDemo {

public static void main(String[] args) throws InterruptedException {

MyRunnable myRunnable = new MyRunnable();

Thread thread = new Thread(myRunnable);

System.out.println("Before starting: " + thread.isAlive());

thread.start();

System.out.println("After starting: " + thread.isAlive());

thread.join();

System.out.println("After thread completed: " + thread.isAlive());

}

}

**Завдання**

1. Протестувати всі приклади. (0.3 б)
2. Створити клас NewThread, що успадковує Thread. Перевизначити метод run (). У циклі for вивести на консоль символ 100 разів. Створити екземпляр класу і запустити новий потік. (0.7 б)
3. Створити клас, який реалізує інтерфейс Runnable. Перевизначити run () метод. Створити цикл for. У циклі виводимо значення від 0 до 100, що діляться на 10 без залишку. Використовуємо статичний метод Thread.sleep () щоб зробити паузу. Створити три потоки, що виконують завдання виведення значень.(1 б)