Introduction to Java week#9

12/07/2023

week	Topic
1	JAVA IDE (NetBean) Installation ,Configuration and Compile
2	Basic structure of Java ,Data & Variable type, operator & basic logic
3	Function(Method) create & calling, Input & output
4	Loop statement ,Array variable
5	Object-oriented programming (OOP),Class & Object, Encapsulation
6	Inheritance, Polymorphism, Interfaces
7	Packages, Access Modifiers(Public ,Protected ,Private class)
8	Collections (Array list, HashMap, Stack)
9	Exception
10	Woking with files(Read, Write)
11	Thread Programing

Working with files

Working with files

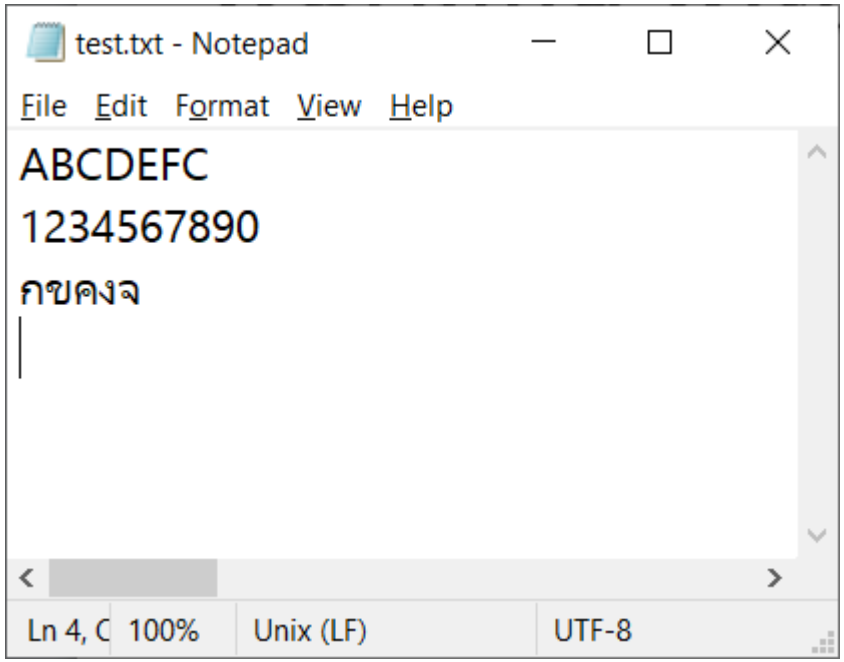
Read & Write Text file

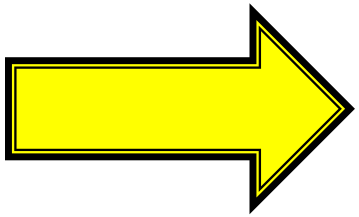
1) **Read** {BufferedReader}

2) **Write** {BufferedWriter}

Working with files

Read Text file





Downloads

Path : /Users/macbook/Downloads

Working with files

Read Text file

```
import java.io.BufferedReader;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class ReadFromFile {
     public static void main(String[] args) {
          try {
               Path file = Paths.get("/User/macbook/Downloads/test.txt");
               BufferedReader reader = Files.newBufferedReader(file ,
               StandardCharsets.UTF_8);
               String line = null;
               while ((line = reader.readLine()) != null) {
                    System.out.println(line);
               }
               reader.close();
          } catch (IOException e) {
                    System.err.println("IOException: " + e.getMessage());
          }
     }
}
```

Working with files

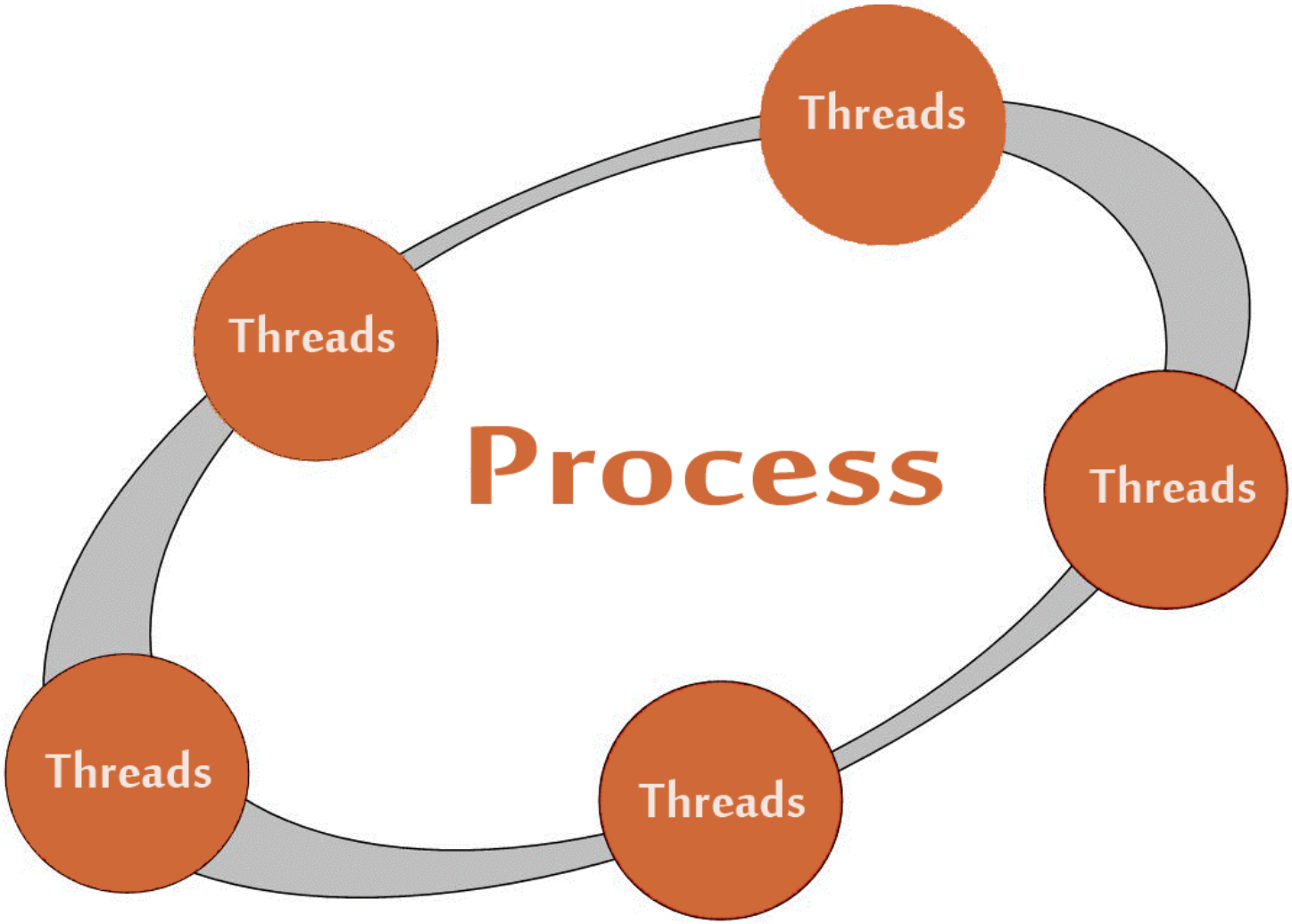
Write Text file

```
import java.io.BufferedWriter;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

public class WriteToFile {
     public static void main(String[] args) {
          try {
               Path file = Paths.get(" ("/User/macbook/Downloads/test2.txt");
               BufferedWriter writer = Files.newBufferedWriter(file,
               StandardCharsets.UTF_8);
               for (int i = 0; i < 10; i++) {
                    writer.write("This is content line " + (i + 1));
                    writer.newLine();
               }
               writer.close();
          } catch (IOException e) {
                    System.err.println("IOException: " + e.getMessage());
          }
     }
}
```

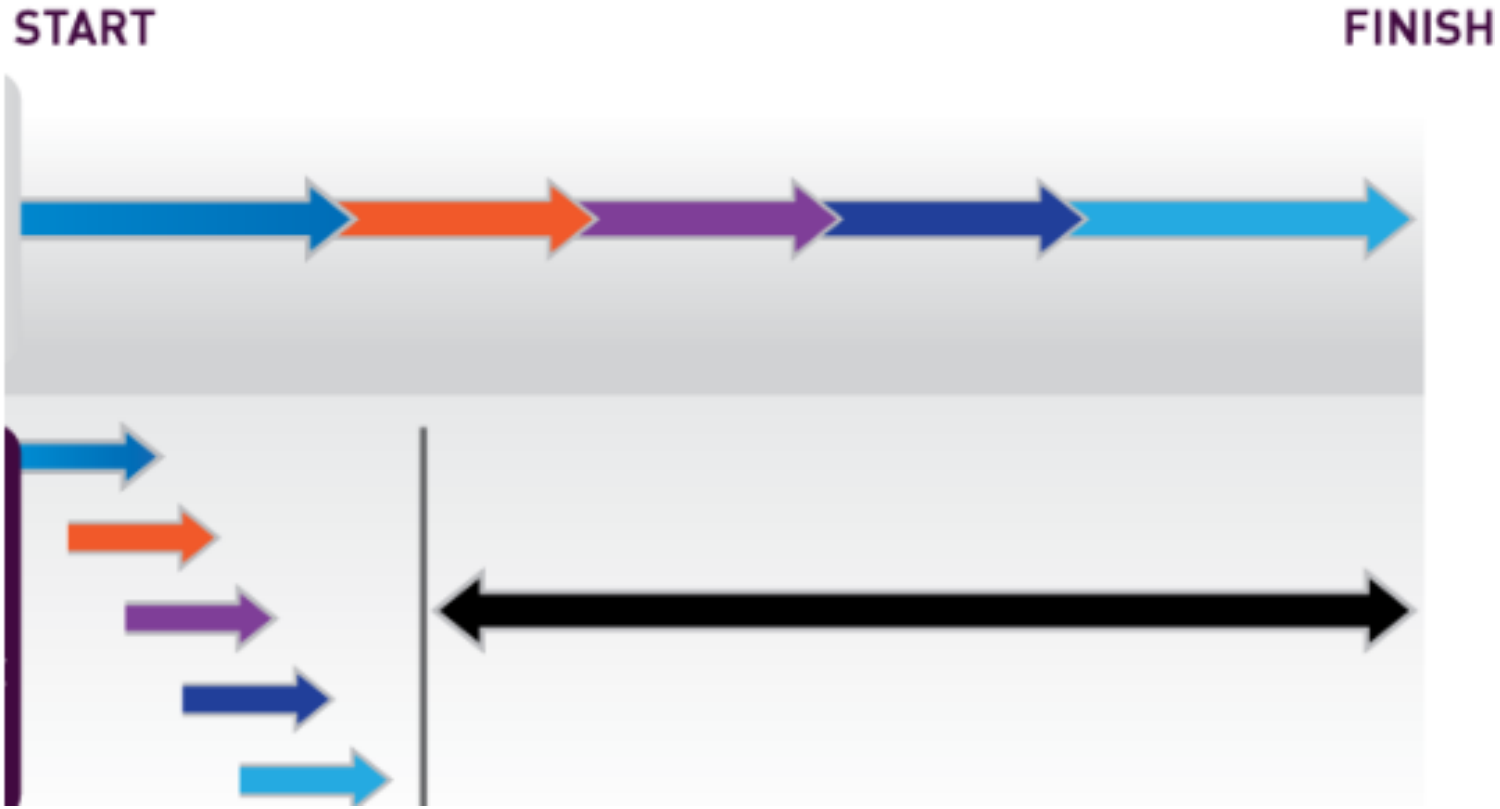
Write Text file –Append to text file

```
import java.io.BufferedWriter;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;

public class WriteToFile {
     public static void main(String[] args) {
          try {
               Path file = Paths.get(" ("/User/macbook/Downloads/test2.txt");

               BufferedWriter writer = Files.newBufferedWriter(file,
               StandardCharsets.UTF_8, StandardOpenOption.APPEND);

               for (int i = 0; i < 10; i++) {
                    writer.write("This is content line " + (i + 1));
                    writer.newLine();
               }
               writer.close();
          } catch (IOException e) {
                    System.err.println("IOException: " + e.getMessage());
          }
     }
}
```

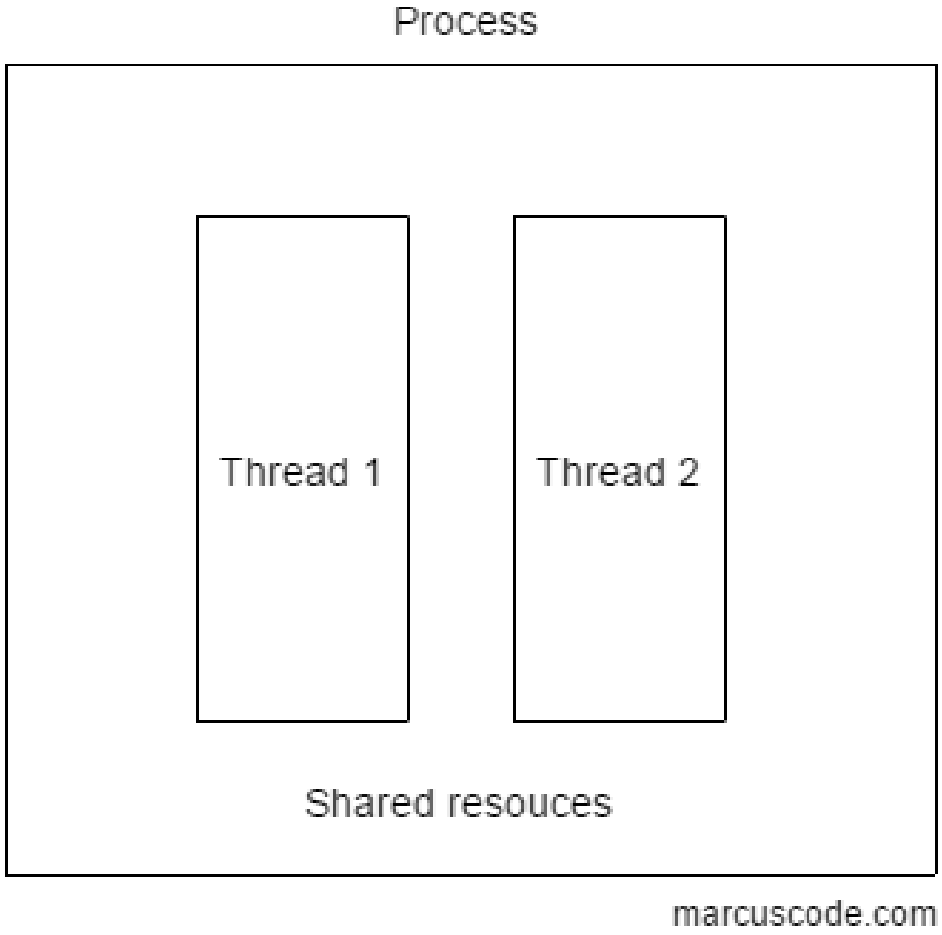
Thread Programing



Thread Programing



Thread Programing



Thread Programing

```
public class ThreadExample {
   public static void main(String[] args){

      Thread t1 = new Thread(new MyThread());
      t1.start();
   }
}

class MyThread implements Runnable {
   @Override
   public void run() {
      System.out.println("Thread is running...");
   }
}
```

Thread Programing

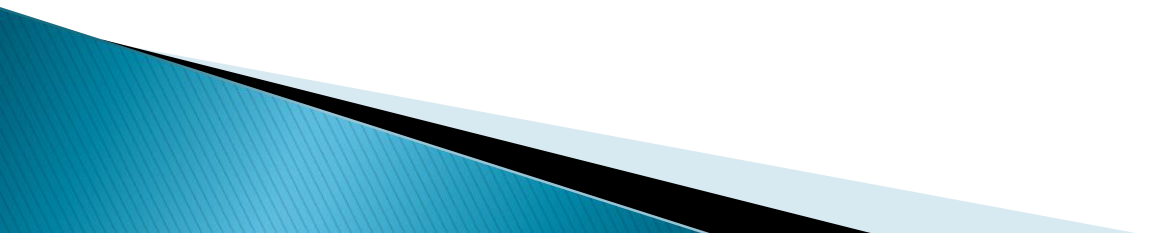
Multi-threaded program

```
class HelloMessage implements Runnable {

    public int threadNum;

    public HelloMessage(int threadNum) {
        this.threadNum = threadNum;
    }

    @Override
    public void run() {
        System.out.println("Hello from thread " + threadNum);
    }

}
```

Thread Programing

Multi-threaded program

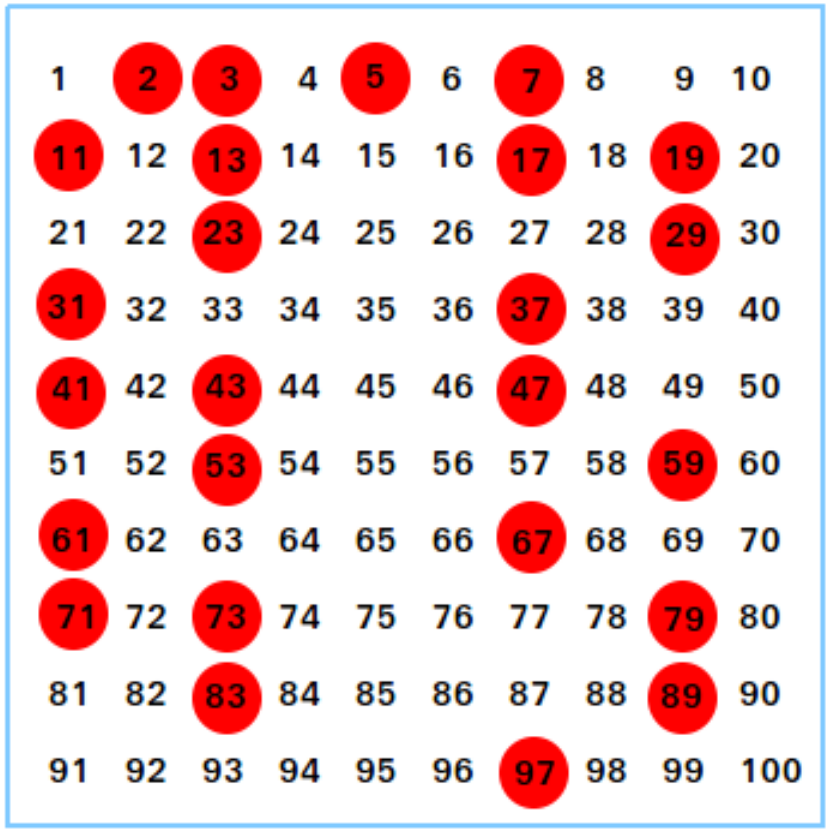
```
public static void main(String[] args){

   for (int i = 0; i < 4; i++) {

      Thread t1 = new Thread(new HelloMessage(i + 1));
      t1.start();

   }

}
```



Thread Programing

Multi-threaded program to find Prime numbers

Prime numbers from 1 to 100



```
class PrimeThread implements Runnable {

     public int threadNum;
     public static int number = 1;
     public static int MAX_NUMBER = 100;

     public PrimeThread(int threadNum) {
          this.threadNum = threadNum;
     }

     @Override
     public void run() {
          while (true) {
          int n = ++number;
          if (n <= MAX_NUMBER) {
          if(IsPrime(n)) {
          System.out.println("Thread " + threadNum + ": " + n + " is prime
          number");
          }
          } else { break; }
     }
}

public boolean IsPrime(int number) {
     for (int i = 2; i < number; i++) {
          if (number % i == 0 && i != number)
                    return false;
     }
     return true;
}
```

Thread Programing

Multi-threaded program to find Prime numbers

```
public static void main(String[] args){

   final int NUM_THREAD = 4;

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

      Thread t1 = new Thread(new PrimeThread(i + 1));

      t1.start();

   }
}
```

- Assignments

ให้ลองสร้างโปรแกรมที่แยก Thread ทำการทำงานเพื่อหาค่า prime number แล้วทำการบันทึกลงใน text file แยกคนละไฟล์กันในแต่ละthread

Thank you