13주차 과제 및 실습

과제 및 실습

- **과제는 문제 및 프로그램 실습으로 나뉘어 있습니다.**
  - 반드시 프로그램 예제를 실행하고 이해한 후 과제를 수행하세요.

- **제출 파일 : 보고서+소스코드**
  - 보고서 : 한글 또는 word 파일
  - 소스코드 : java 파일만 제출
  - 위 두 파일을 학번_이름_실습주차.zip으로 압축하여 제출

- **보고서 내용**
  - 과제 문제 : 해답, 해답의 이유
  - 실습 문제 : 해답소스코드, 프로그램 설명, 결과 화면(결과 캡처)
    - 프로그램 설명 : 작성한 소스코드의 내용, 소스코드 내 주석으로 대체 가능
  - 보고서 파일명 : 학번_이름_실습주차.hwp
    - ex) 2023111010_홍길동_2주차.hwp

- **소스코드**
  - 파일명 : Exec1_학번_문제번호.java

- **제출 : eclass**
  - 과제 연장 제출을 희망할 경우 직접 또는 이메일을 통해 조교에게 요청

프로그램 예제1. 스레드 예제(1) : Thread 클래스

- 스레드 객체를 만들고 시작하기
- 스레드 객체 배열을 만들고 시작하기

→ 스레드는 동시에 실행되므로 순서를 알 수 없다.

```
Thread-2가 실행되었습니다.
Thread-4가 실행되었습니다.
Thread-5가 실행되었습니다.
Thread-3가 실행되었습니다.
Thread-0가 실행되었습니다.
Thread-1가 실행되었습니다.
```

```java
class MyThread extends Thread{
    public void run(){
        String name = this.getName();
        System.out.println(name + "가 실행되었습니다.");
    }
}

public class ThreadTest {
    public static void main(String[] args) {
        MyThread mt = new MyThread();
        mt.start();     //Thread-0

        MyThread[] mArr = new MyThread[5];

        mArr[0] = new MyThread();
        mArr[1] = new MyThread();
        mArr[2] = new MyThread();
        mArr[3] = new MyThread();
        mArr[4] = new MyThread();

        mArr[0].start();    //Thread-1
        mArr[1].start();    //Thread-2
        mArr[2].start();    //Thread-3
        mArr[3].start();    //Thread-4
        mArr[4].start();    //Thread-5
    }
}
```

프로그램 예제2.

- Runnable 인터페이스로 스레드 만들기

```
Thread 15가 시작되었습니다.
Thread 14가 시작되었습니다.
Thread 14 : 10
Thread 15 : 1000
Thread 14의 값 : 20
Thread 15의 값 : 1010
Thread 15의 값 : 1020
Thread 14의 값 : 30
Thread 14의 값 : 40
Thread 15의 값 : 1030
Thread 14의 값 : 50
Thread 15의 값 : 1040
Thread 15의 값 : 1050
Thread 14의 값 : 60
```

```java
class Common{
    int value = 0;
    Common(int value){
        this.value = value;
    }
    public int getValue() {
        return value;
    }
    public void setValue(int value) {
        this.value = value;
    }
}

class MyThread implements Runnable{
    Common c;
    MyThread(int value){
        c = new Common(value);
    }

    public void run(){
        int id = (int)Thread.currentThread().getId();

        System.out.println("Thread " + id + "가 시작되었습니다.");
        System.out.println("Thread " + id + " : " + c.getValue());

        try {
            for(int i = 0; i< 5; i++) {
                Thread.sleep(10);
                c.setValue(c.getValue()+10);
                System.out.println("Thread " + id + "의 값 : " + c.getValue());
            }
        }catch(InterruptedException e) {
            System.out.println(e);
        }
    }
}

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

        Thread t1 = new Thread(new MyThread(10));
        Thread t2 = new Thread(new MyThread(1000));
        t1.start();
        t2.start();
    }
}
```

과제문제 : 다음의 문제를 풀고 정답의 이유도 함께 작성하세요.

1. 프로세스와 스레드의 설명으로 옳지 않은 것은?

① 운영체제에 의해 실행중인 하나의 프로그램을 프로세스라고 한다.

② 하나의 프로세스는 내부에 최소 한 개 이상의 스레드를 가질 수 있다.

③ 프로세스 내의 스레드들은 운영체제에 의해 독립적으로 실행된다.

④ 두가지 이상의 작업을 동시에 처리하는 것을 멀티 태스킹이라고 한다.

⑤ 멀티 프로세스는 동시에 여러 개의 프로세스가 실행되고 있는 것을 말한다.

2. 자바의 멀티 태스킹에 대한 설명 중 틀린 것은?

① 자바에서는 다수의 스레드를 가진 멀티 스레드를 지원한다.

② 자바에서는 다수의 프로세스를 가진 멀티프로세스를 지원한다.

③ 자바에서는 하나의 JVM은 오직 하나의 응용프로그램만 사용한다.

④ 자바에서는 스레드의 생명주기는 JVM에 의해 관리된다.

과제문제 : 다음의 문제를 풀고 정답의 이유도 함께 작성하세요.

3. 스레드를 만들기 위해 Thread를 상속받아 오버라이딩해야 하는 메서드는 무엇인가?

① run()　② start()　③ sleep()　④ wait()　⑤ runnable()

4. 스레드를 생성하기 위해서 Thread 클래스와 Runnable 인터페이스를 사용하는 두가지의 방법이 있다. 이유는 무엇인가?

실습 문제1

Exec1) 은행 계좌를 만들어 입금과 출금을 한다. 스레드로 아래와 같이 입출금을 진행하라.

- 계좌 클래스 Account를 만들고 입금, 출금을 위한 메서드를 만들어라.
- 은행 거래 클래스 Bank를 스레드로 만들고 맴버로 계좌를 갖는다.
- Bank클래스에서 은행거래는 Random으로 0~1000사이의 값을 생성한다.
- 위 Random 값 만큼 정지(sleep)하고, 이후 위 Random 값이 짝수값이면 입금하고, 홀수 값이면 출금한다.

- 은행 거래 클래스로 객체를 생성하고 스레드를 동작시킨다.

- 실행 결과

```
계좌를 만들었습니다. 현재 잔액 : 0
>>576원을 입금 요청하였습니다.
현재 잔액 : 576
>>154원을 입금 요청하였습니다.
현재 잔액 : 730
>>85원을 출금 요청하였습니다.
>> 출금 : 85
현재 잔액 : 645
>>715원을 출금 요청하였습니다.
>>70원 잔액이 부족하여 출금하지 못합니다.
현재 잔액 : 645
>>429원을 출금 요청하였습니다.
>> 출금 : 429
현재 잔액 : 216
```

```java
class Account{//계좌 클래스

}

class Bank extends Thread { //은행 클래스
    public void run() {

    }
}

public class BankEx {
    public static void main(String args[]) {
        Bank b = new Bank(0);    //0원으로 계좌생성
        b.start();
    }
}
```