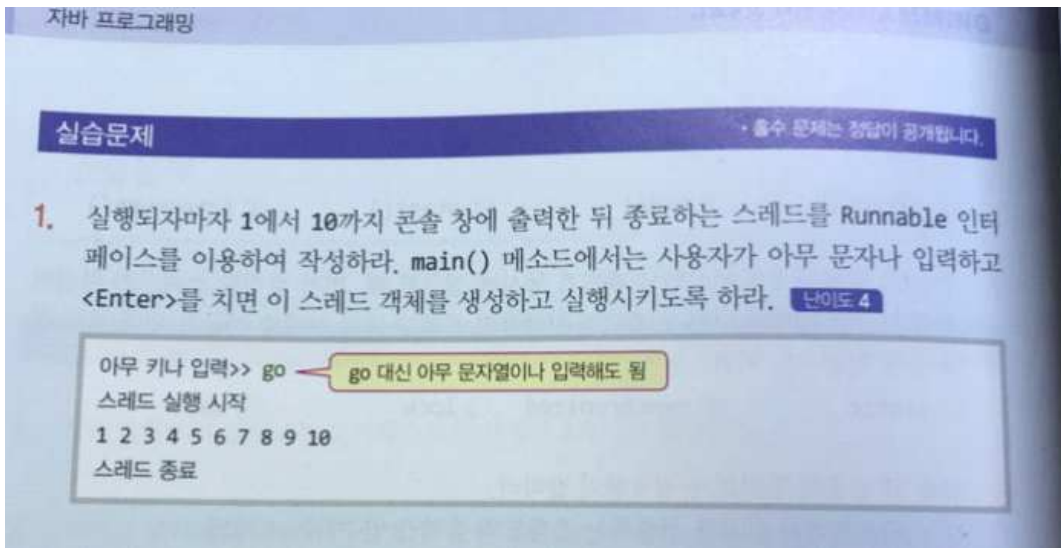


과목명	객체지향프로그래밍	분반	x	담당교수	김화성 교수님
학과	전자통신공학과	학번	2016707079	이름	하상천
과제명: 스레드					

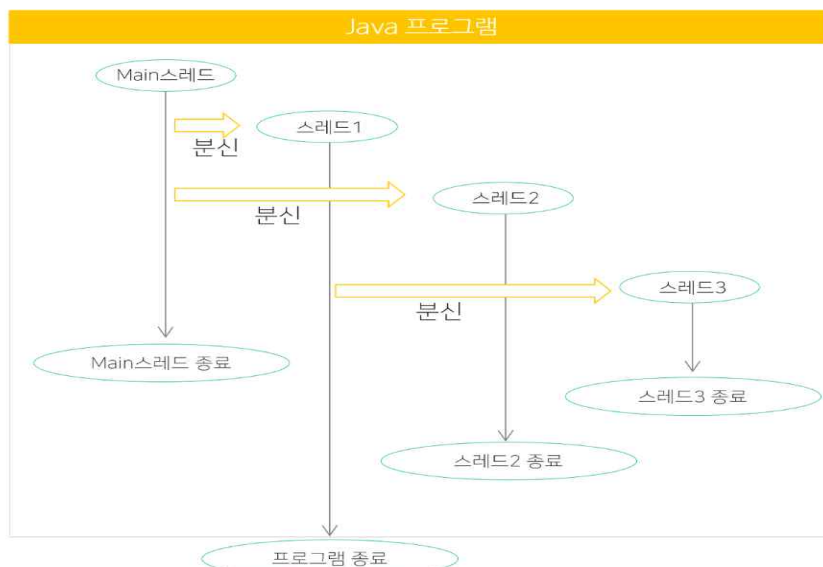
1, 과제설명 (사용자 요구사항 기술: 과제에 대한 설명 및 목표)



2, 사용자 요구사항을 정형적 방법으로 기술 (UML, Pseudo code, 그림등을 이용하여 기술)

이용하는 방법은 다음과 같다.

- (1) 우선 Runnable인터페이스를 구현하는 클래스를 하나 만든다.
- (2) 위에서 만든 클래스의 run() 메소드에 시킬 일을 정의한다.
- (3) 위에서 만든 클래스를 생성하고, Thread를 생성할 때 인자로 넣어서 생성한다.
- (4) Thread의 start() 메소드를 호출하여 새로운 스레드가 만들어져서 실행되도록 한다.



3. 알고리즘 및 자료구조 설계 내용

scanner 객체를 생성하여, 문자열을 입력 받으면 Thread 객체를 생성하도록 하였다. Runnable 인터페이스를 구현상속 하고 있는 TimerRunnable의 객체를 Thread 생성자로 전달하였다. 그리고 start() 메소드를 통해 스레드를 시작하게 하였고, JVM에 의해 run() 메소드가 호출되어 실행되도록 하였다. run() 메소드를 오버라이딩하여 1부터 10까지 출력하도록 하였다.

4. 소스코드 설명 (직접 작성한 소스코드중에 핵심 부분을 발췌하여 설명)

```
Thread th = new Thread(new TimerRunnable());
th.start();
```

Thread 객체를 생성하고, 생성자에 TimerRunnable의 객체를 전달한다. 그리고 start() 메소드를 호출하여 스레드를 시작하게 한다.

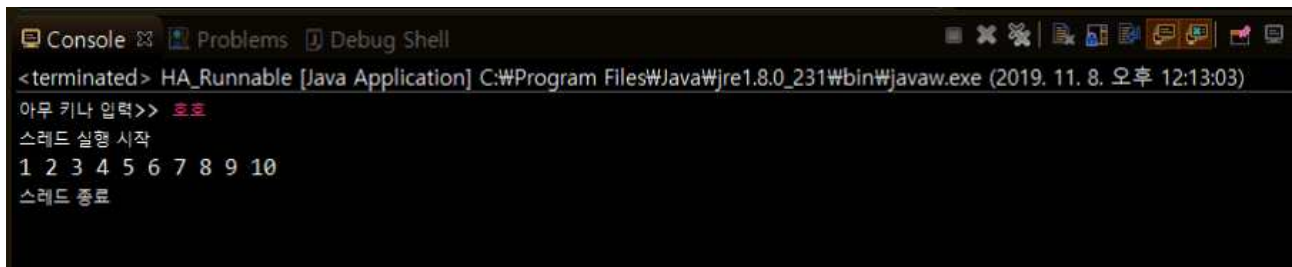
```
Scanner scanner = new Scanner(System.in);
System.out.print("아무키나입력>> ");
scanner.next();
```

문자열을 입력받기 위해 Scanner 클래스의 scanner 객체를 생성한다. scanner.next()로 한 이유는 scanner.nextLine() 으로 하면 문자열을 입력하지 않고 엔터키만 눌러도 넘어가기 때문이다.

```
for(int i =0;i<10;i++) {
    System.out.print(n+ " ");
    n++;
}
```

1부터 10까지 출력하기 위한 반복문이다.

5. 실행결과 및 설명 (실행 결과를 캡처하여 첨부한 후 설명)



(그림을 문서에 포함, 글자처럼 취급 옵션, 잉크 절약과 잘 보이게 하기위해 그림 반전)

작성된 프로그램을 실행한 결과 위와 같은 화면이 출력 되었다.

6. 고찰 (과제를 진행하면서 배운점 이나, 시행 착오 내용, 기타 느낀점)

이번 과제는 Thread를 이용하여 1부터 10까지 출력하는 과제였다. 예전부터 Thread가 중요하다는 말을 많이 들어서, 수업시간에도 집중해서 듣고 했지만 사실 어려운 내용이라 조금 이해가 되지 않았다. 하지만 이번 과제를 통해서 Thread와 조금 더 가까워진 것 같다. 이번 과제는 Runnable 인터페이스를 이용하여 Thread를 만들었지만, 다음에는 Thread 클래스를 상속받아 Thread를 만드는 것도 해보고싶다. 확실히 이론으로 공부하는 것보다 직접 코딩하고, 궁금한 점은 인터넷에서 찾아보면서 코딩하는 것이 이해도 잘 되고 기억에도 오래 남는 것 같다. 이번에는 조금 단순한 스레드 문제 였지만, 개인적으로 동기화를 이용한 스레드 문제들도 몇 개 풀어보는 것이 좋은 것 같다.

7. 전체 소스코드 (글자크기 9에 줄간격을 120%로 유지하고 한 줄이 너무 길지 않게 작성)

```
package HA_homework;
import java.util.Scanner; //Scanner클래스를 사용하기 위한 import

public class HA_Runnable {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); //입력받기 위해 Scanner 클래스의 scanner 객체 생성
        System.out.print("아무키나 입력>> ");
        scanner.next(); //scanner.nextLine()
        으로하면 문자열을 입력하지 않고 엔터만 눌러도 넘어가기 때문이다.
        Thread th = new Thread(new TimerRunnable());
        th.start();
    }
}

class TimerRunnable implements Runnable {
    int n = 1;
    public void run() {
        System.out.println("스레드 실행 시작");
        for(int i = 0; i < 10; i++) { //1부터 10까지 출력하기 위한 반복문
            System.out.print(n + " ");
            n++;
        }
        System.out.println();
        System.out.println("스레드 종료");
    }
}
```

(글자크기는 10으로 유지하고 줄간격도 160%를 유지할 것)