



제가 프로그래밍을 처음 배우기 시작한 이유는 컴퓨터 공학을 완수하거나 그분야의 최고가 되고 싶어서가 아니라

매우 단순한 이유 때문이었어요. 저를 포함한 저희집
남매들이 서로 같이 재밌게 할 수 있는 뭔가를 만들고 싶었거든요.

간단한 프로그램을 짜고 나서, 조금씩 기능들을 추가했어요. 그리고 새로운 걸 배워야되는 상황이 닥치면



Java 프로그래밍 수업내용 정리 및 실습 과제

처음에는 막막한 느낌이지만 시간이 지나면 감이 오거든요.

악기를 연주하거나 스포츠를 하는것과 별로 다르게 없어요.

책을 보거나 인터넷을 찾아보면서 배우고 거기에 저만의 방식을 추가했어요.



(NBA 농구선수 크리스 보쉬) 프로그래밍은 누구나 배울 수 있는 거예요.
물론 처음엔 겁이나죠. 하지만 그것은 비단 프로그래밍에만 해당되는게 아닙니다.



프로그래밍은 천재들만 할 수 있는것이 아니예요. 의지력이 필요할 뿐입니다.



• 2020.04.14.화
• B반 송명훈

더하기 빼기... 그게 거의 전부죠.

목 차

1. 수업 내용 정리
2. 실습화면 캡처
3. 연습문제 6-1
4. 연습문제 6-2
5. 연습문제 6-3
6. 연습문제 6-4



1. 수업 내용 정리

1. 수업 내용 정리 [복수 클래스를 사용한 개발]

소스 파일의 분리

1. 대규모, 팀 단위 개발에서는 혼자 개발하는 것이 아니므로 각자 분담해 소스 코드를 부품화(=모듈화)한다.
2. Java에서는 소스 모듈화를 위해 메소드 뿐 아니라 클래스, 클래스끼리 묶은 패키지(package)로 그룹화해서 분류, 관리되도록 시스템 구성되어 있음.
 - main() 메소드의 라인 수 증가 ⇨ 중복부분을 묶어서 메소드로 분리
 - 메소드 수 증가 ⇨ 특정 요소(특징, 사용패턴 등)를 고려해 클래스로 묶어 분리
 - 클래스 수 증가 ⇨ 패키지로 그룹을 만들어 분리
3. 이름 공간 (name space)
 - Java에서 name space = package 라고 생각해도 무방하다.
 - 자신이 작성한 클래스에 대해 개발자가 자유롭게 이름 지을 수 있음
 - 패키지명 자체의 충돌을 회피하기 위해 보유한(일반적으로 회사) 도메인의 앞 뒤를 바꿔 패키지명으로 사용
Ex] com.example, com.google, com.naver 등
 - Ex] 지역변수는 해당 지역 이름 공간에서 같은 변수명을 사용할 수 없지만 영역이 다르면 변수명이 같을 수 있음 ⇨ 한 메소드에서 사용한 변수이름을 다른 메소드 안에서 중복 사용 가능

Context	이름	이름공간 식별자	지역 이름
경로	/home/user/readme.txt	/home/user (path)	readme.txt (file name)
도메인 네임	www.example.com	example.com (domain)	www (host name)
자바	java.util.Date	java.util	Date

1. 수업 내용 정리 [복수 클래스를 사용한 개발]

Java API에 대하여

1. API (Application Programming Interface) 란?
 - API란 Java 시스템을 제어하기 위해서 Java에서 자체적으로 제공하는 클래스군(명령어)을 의미한다. JDK 제공.
 - 개발자는 자바가 제공한 API를 이용해 프로그래밍을 한다.
 - Java programming은 결국 Java에서 제공하는 패키지들을 조립해서 개발자가 논리 구조를 정의하고 구성하는 과정이라 할 수 있다.
2. Java에는 최소 200개 이상의 패키지, 3500개 이상의 클래스(Java 7 기준)가 라이브러리에 존재
 - Java API는 java. 이나 javax. 으로 시작하는 패키지명 사용
 - API로 준비되어 있는 클래스들 정보는 API 레퍼런스에서 확인 가능
3. Java API에 포함된 대표적인 패키지

java.lang	Java에서 가장 중요한 클래스군 (자동 import됨, System가 여기 소속)
java.util	프로그래밍을 편리하게 해 주는 클래스군 (Scanner가 여기 포함)
java.math	수학에 관한 클래스군
java.net	네트워크 통신 등에 필요한 클래스군
java.io	파일 입출력 등에 필요한 클래스군

1. 수업 내용 정리 [API reference]

Class	Description
Boolean	The Boolean class wraps a value of the primitive type <code>boolean</code> in an object.
Byte	The Byte class wraps a value of primitive type <code>byte</code> in an object.
Character	The Character class wraps a value of the primitive type <code>char</code> in an object.
Character.Subset	Instances of this class represent particular subsets of the Unicode character set.
Character.UnicodeBlock	A family of character subsets representing the character blocks in the Unicode specification.
Class<T>	Instances of the class <code>Class</code> represent classes and interfaces in a running Java application.
ClassLoader	A class loader is an object that is responsible for loading classes.
ClassValue<T>	Lazily associate a computed value with (potentially) every type.
Compiler	Deprecated, for removal: This API element is subject to removal in a future version. <i>JIT compilers and their technologies vary too widely to be controlled effectively by a standardized interface.</i>
Double	The Double class wraps a value of the primitive type <code>double</code> in an object.
Enum<E extends Enum<E>>	This is the common base class of all Java language enumeration types.
Enum.EnumDesc<E extends Enum<E>>	A nominal descriptor for an enum constant.

- ✓ 우리가 셀 수 없이 사용하는 자료형 (Float, Int, Long, Short, String 등) 과제 수행 시 몇십번씩 사용하는 `System.out.println()`의 `System` 모두 Java API가 제공하는 클래스군 `java.lang` 에서 기본적으로 정의 되어 있는 클래스이다.
- ✓ 자동 import 되기 때문에 `import java.lang;` 안 써도 축약해서 사용 가능
- ✓ API reference에서 정확한 정보를 찾아 볼 수 있고, IntelliJ의 Quick documentation (Ctrl + Q) 기능을 사용해서도 해당 부분의 자료를 읽을 수 있다.



2. 실습화면 캡처

2. 실습 화면 캡처

6-1. 계산기 프로그램

```
Ex6_1.java x
1 package Day5;
2
3 public class Ex6_1 {
4     public static void main(String[] args) {
5         int a = 10;
6         int b = 2;
7         int total = add(a, b);
8         int delta = minus(a, b);
9         System.out.println("더하면 " + total + ", 빼면 " + delta);
10    }
11
12    public static int add(int a, int b) {
13        return a + b;
14    }
15
16    public static int minus(int a, int b) {
17        return a - b;
18    }
19 }
```

Run - HighTechJava

Run: Ex6_1 x

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\P
더하면 12, 빼면 8
|
Process finished with exit code 0

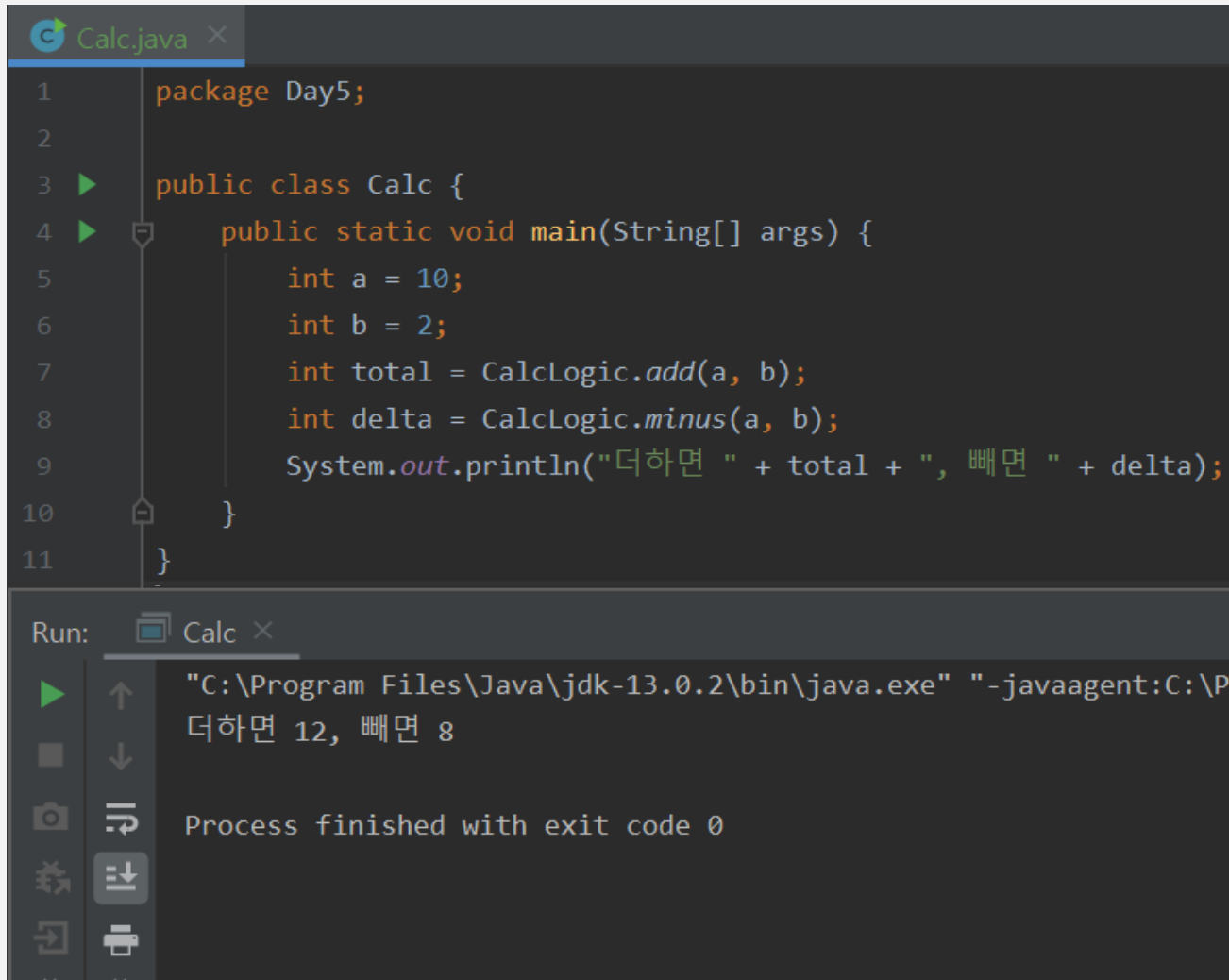
6-2. CalcLogic.java에 계산처리 추가

```
CalcLogic.java x
1 package Day5;
2
3 public class CalcLogic {
4
5     public static int add(int a, int b) {
6         return a + b;
7     }
8
9     public static int minus(int a, int b) {
10        return a - b;
11    }
12 }
```

6-3. Calc.java에 에러 발생

```
CalcLogic.java x Calc.java x
1 package Day5;
2
3 public class Calc {
4     public static void main(String[] args) {
5         int a = 10;
6         int b = 2;
7         int total = add(a, b);
8         int delta = minus(a, b);
9         System.out.println("더하면 " + total + ", 빼면 " + delta);
10    }
11 }
```


2. 실습 화면 캡처



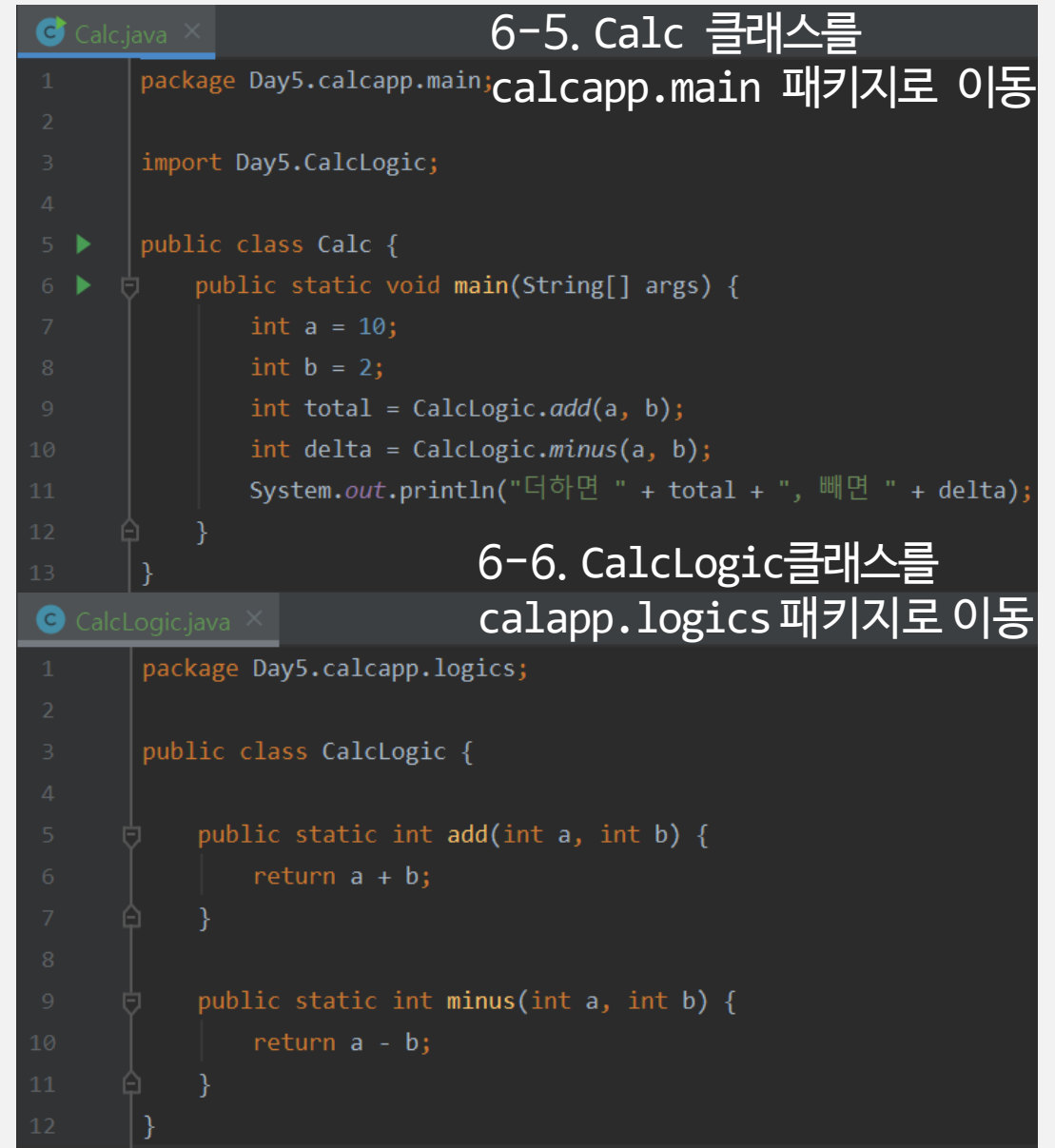
```
1 package Day5;
2
3 public class Calc {
4     public static void main(String[] args) {
5         int a = 10;
6         int b = 2;
7         int total = CalcLogic.add(a, b);
8         int delta = CalcLogic.minus(a, b);
9         System.out.println("더하면 " + total + ", 빼면 " + delta);
10    }
11 }
```

Run: Calc ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\P
더하면 12, 빼면 8

Process finished with exit code 0

6-4. Calc.java 클래스.메소드로 변경하여 에러 해결



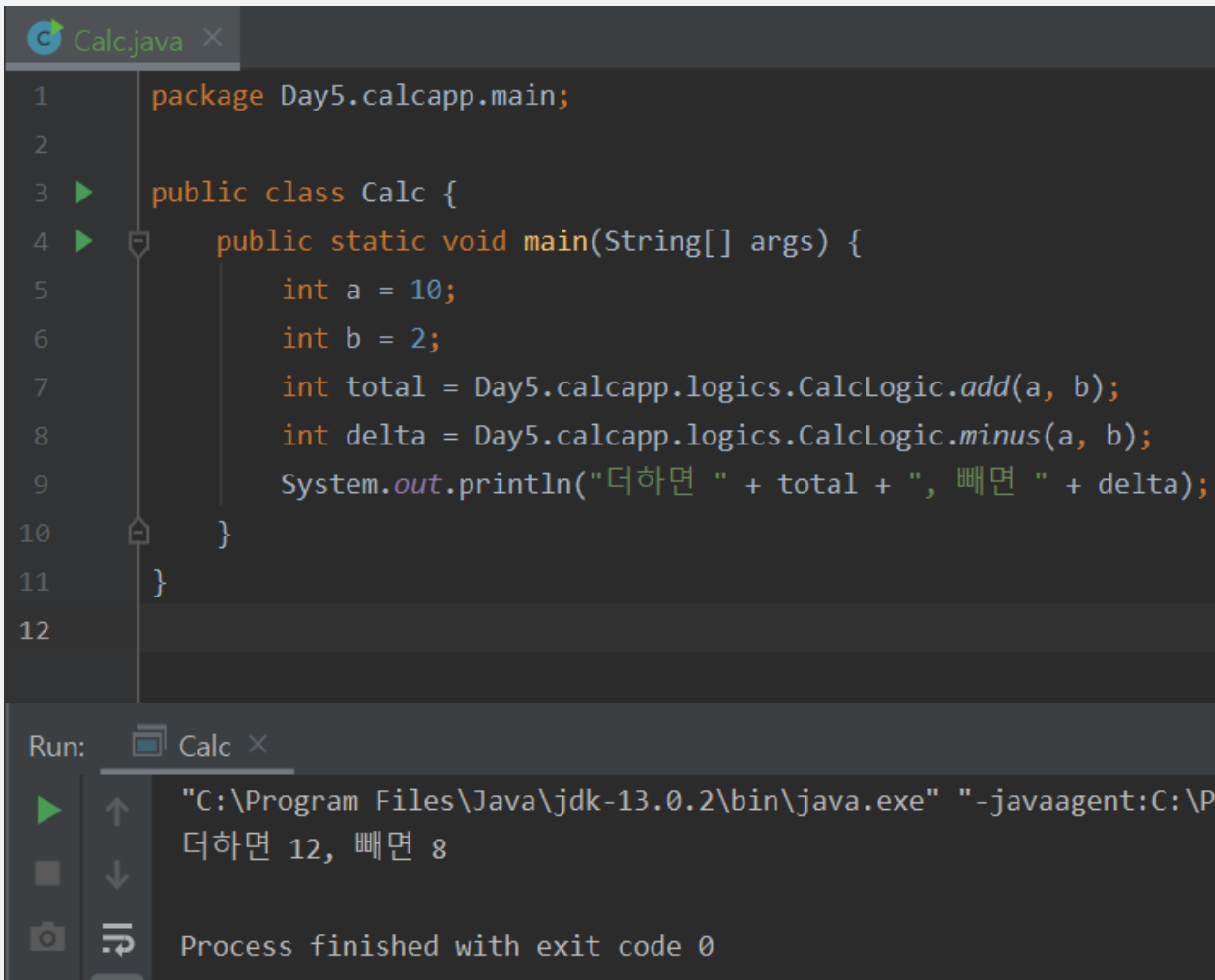
6-5. Calc 클래스를
calapp.main 패키지로 이동

```
1 package Day5.calapp.main;
2
3 import Day5.CalcLogic;
4
5 public class Calc {
6     public static void main(String[] args) {
7         int a = 10;
8         int b = 2;
9         int total = CalcLogic.add(a, b);
10        int delta = CalcLogic.minus(a, b);
11        System.out.println("더하면 " + total + ", 빼면 " + delta);
12    }
13 }
```

6-6. CalcLogic클래스를
calapp.logics 패키지로 이동

```
1 package Day5.calapp.logics;
2
3 public class CalcLogic {
4
5     public static int add(int a, int b) {
6         return a + b;
7     }
8
9     public static int minus(int a, int b) {
10        return a - b;
11    }
12 }
```

2. 실습 화면 캡처



```
1 package Day5.calcapp.main;
2
3 public class Calc {
4     public static void main(String[] args) {
5         int a = 10;
6         int b = 2;
7         int total = Day5.calcapp.logics.CalcLogic.add(a, b);
8         int delta = Day5.calcapp.logics.CalcLogic.minus(a, b);
9         System.out.println("더하면 " + total + ", 빼면 " + delta);
10    }
11 }
12
```

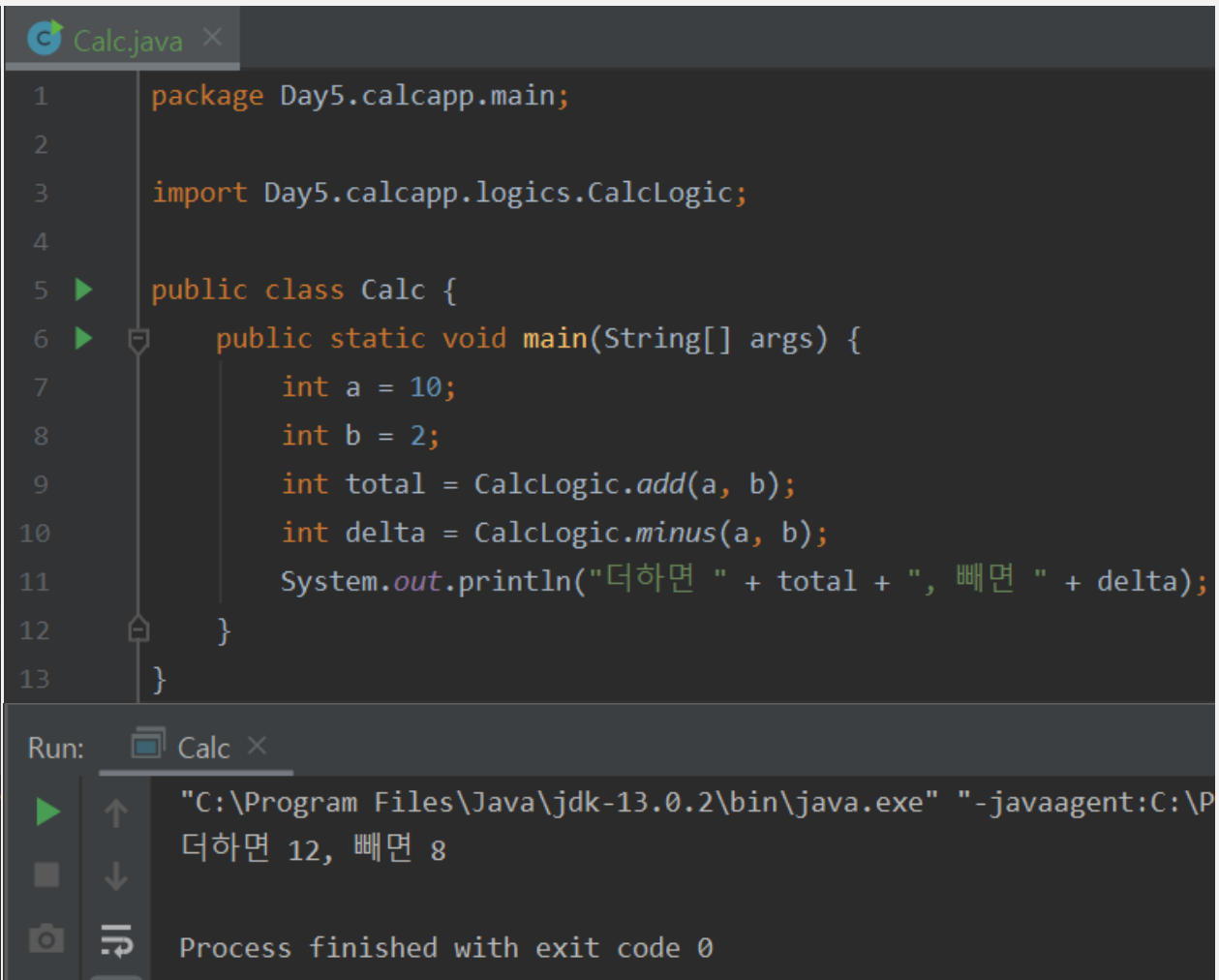
Run: Calc ×

↑ "C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\P
더하면 12, 빼면 8

↓

Process finished with exit code 0

6-7. 다른 패키지에 있는 클래스를 호출



```
1 package Day5.calcapp.main;
2
3 import Day5.calcapp.logics.CalcLogic;
4
5 public class Calc {
6     public static void main(String[] args) {
7         int a = 10;
8         int b = 2;
9         int total = CalcLogic.add(a, b);
10        int delta = CalcLogic.minus(a, b);
11        System.out.println("더하면 " + total + ", 빼면 " + delta);
12    }
13 }
```

Run: Calc ×

↑ "C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\P
더하면 12, 빼면 8

↓

Process finished with exit code 0

6-8. Calc.java 에 import 문을 추가

2. 실습 화면 캡처

```
Ex10_Calc.java ×
1 package Day5.calcapp.main;
2
3 import java.util.Arrays;
4
5 public class Ex10_Calc {
6     public static void main(String[] args) {
7         int[] heights = {172, 149, 152, 191, 155};
8         Arrays.sort(heights);
9         for (int h : heights) {
10             System.out.println(h);
11         }
12     }
13 }
```

Run: Ex10_Calc ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\P

149
152
155
172
191

Process finished with exit code 0

6-10. API 이용의 예

```
Main.java ×
1 package Day5.calcapp.main;
2
3 public class Main {
4     public static void main(String[] args) {
5         System.out.printf("%d 숫자를 나타냄\n", 1);
6         System.out.printf("%s 을 표시\n", "문자열");
7         System.out.printf("%3.2f 정수부분이 3자리, 소수점 아래가 2자리\n", 3.5f);
8         System.out.printf("%2d 2자리의 10진수 숫자 숫자 이외의 부분은 공백\n", 1);
9         System.out.printf("%02d 2자리의 10진수 숫자 숫자 이외의 부분은 0\n", 1);
10        System.out.printf("가\t나\t다 : %s의 예제\n", "탭");
11        System.out.printf("제 이름은 %s입니다. 나이는 %d살 키는 %3.1fcm 입니다", "송명훈", 24, 173.3);
12    }
13 }
14 }
```

Run: Main ×

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDE

1 숫자를 나타냄
문자열 을 표시
3.50 정수부분이 3자리, 소수점 아래가 2자리
1 2자리의 10진수 숫자 숫자 이외의 부분은 공백
01 2자리의 10진수 숫자 숫자 이외의 부분은 0
가 나 다 : 탭의 예제
제 이름은 송명훈입니다. 나이는 24살 키는 173.3cm 입니다
Process finished with exit code 0

출력 포맷



3. 연습문제 6-1

3. 연습문제 6-1

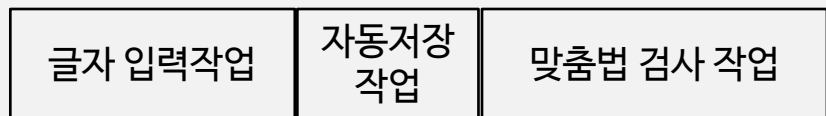
6-1. 문제

: java.lang.Thread 클래스를 조사하여, 프로그램을 3초간 멈추게 하는 프로그램을 완성하시오.
throw Exception에 대해서는 나중에 배우니 일단 무시하시오

```
public class PracProb6_1 {  
    public static void main(String[] args) throws Exception {  
        System.out.println("3초간 기다림!");  
  
        // 3초간 기다림  
  
        System.out.println("끝");  
    }  
}
```

3. 연습문제 6-1

1 개의 프로세스 (Ex] MS Word)



IF (1 개의 Thread 사용) {
1개의 Thread가 순차적 작업 진행
}

1 개의 프로세스 (Ex] MS Word)



IF (3 개의 Thread를 사용) {
각 Thread 개별적 작업
진행
}

⇒ Multithread 작업방식

일반적으로 스레드 (Thread) 란?

사전적 의미

- "프로세스 내에서 실행되는 여러 흐름의 단위"
- 프로세스의 특정 수행 경로
- 프로세스가 할당 받은 자원을 이용하는 실행의 단위

그럼 프로세스(Process) 란?

사전적 의미

- "컴퓨터에서 연속적으로 실행되고 있는 프로그램"
- Ex] MS Word. (다수의 프로세스로 실행되는 프로그램도 존재)
- 운영체제로부터 시스템 자원(CPU 시간, 운영에 필요한 주소공간, 독립된 메모리 영역)을 할당 받는 작업의 단위
- 메모리 영역은 Code, Data, Stack, Heap 구조 (이에 대한 추가적인 설명은 생략)
- 즉, 동작하고 있는 프로그램을 의미
- 기본적으로 프로세스당 최소 1개의 스레드 (메인 스레드)를 가짐

⇒ ∴ 따라서 Thread는 Process의 구성 단위.

3. 연습문제 6-1

자바에서 스레드 란?

- 일반적인 thread와 큰 차이가 없음
- JVM가 운영체제 역할 수행
- java에는 process 존재하지 않고 thread만 존재
- java thread는 JVM에 의해 예정된 실행 단위 코드 블록
- 대부분 JVM이 관리 및 조정
- 개발자는 java thread로 작동할 thread 코드를 작성하고 실행 시작을 JVM에 요청

※ 용어정리

- 인스턴스(instance) : 클래스에 의해 생성된 객체
String strName = new String()
이 때, strName은 String 클래스의 인스턴스이다.

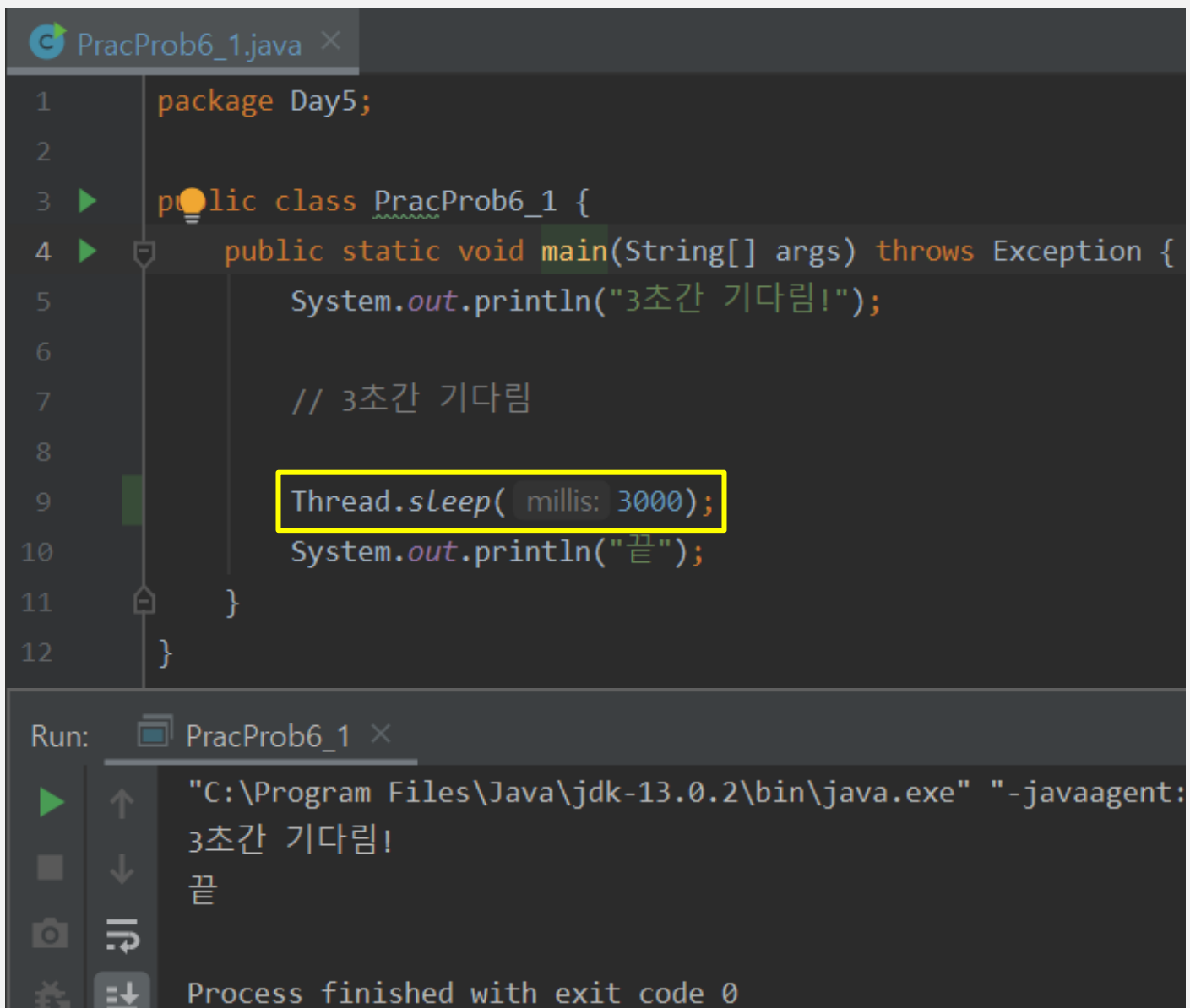
java.lang.Thread class

- Thread class는 thread 사용을 조작하기 위한 코드를 작성할 때 사용하는 class
- 수업 중 사용한 String 자료형은 실제로는 String Class 생성자로 생성된 String 인스턴스
 - 개발자는 java.lang.String class로 String 객체를 생성하고 String class 내부 메소드를 호출하여 기능 사용 가능. (Ex] strName.split(" ") 등)
- String 처럼 Thread class를 이용해 Thread 인스턴스와 메소드를 생성 및 사용 가능.
이를 통해 java에서 thread의 작동 방식을 조절할 수 있는 코드를 작성할 수 있다.

Thread.sleep(long millis)

- Java.lang.Thread class의 메소드 중 하나
- argument로 입력 받은 millis 값에 해당하는 밀리 초(milliseconds, ms)만큼 현재 Thread의 작업을 중지시킴
- 3 초를 정지시키기 위해서는 Thread.sleep 메소드의 argument에 3000을 입력해주면 된다.

3. 연습문제 6-1



```
PracProb6_1.java x
1 package Day5;
2
3 public class PracProb6_1 {
4     public static void main(String[] args) throws Exception {
5         System.out.println("3초간 기다림!");
6
7         // 3초간 기다림
8
9         Thread.sleep( millis: 3000);
10        System.out.println("끝");
11    }
12 }
```

Run: PracProb6_1 x

"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:
3초간 기다림!
끝
Process finished with exit code 0

3. 연습문제 6-1

```
PracProb6_1.java x PracProb6_1revised.java x
1 package Day5.calcapp;
2
3 public class PracProb6_1revised {
4     public static void main(String[] args) throws Exception {
5         System.out.println("3초간 기다림!");
6         long initTime = System.currentTimeMillis();
7         // 3초간 기다림
8         Thread.sleep(3000);
9         long finalTime = System.currentTimeMillis();
10        System.out.println("끝");
11        System.out.println("시작시간: " + initTime + " millisec");
12        System.out.println("종료시간: " + finalTime + " millisec");
13        System.out.println("측정된 타임 슬립: " + (finalTime - initTime) + " millisec");
14    }
15 }
```

Run: PracProb6_1revised x

```
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains
3초간 기다림!
끝
시작시간: 1586956167784millisec
종료시간: 1586956170785millisec
측정된 타임 슬립: 3001 millisec
Process finished with exit code 0
```

✓ 1millisec 오차 발생을 확인
✓ 실제 정지한 시간으로 계산된 시간은 3.001 sec 이다.

- 실제로 3초가 정지되는지 확인하기 위해 정지되는 시간 출력
- `System.currentTimeMillis()`
: 메소드 호출 시 해당하는 시각을
 $\text{millisecond} \left(\frac{1}{1000} \text{ sec} \right)$
단위로 반환한다.
- `Thread.sleep` 메소드 호출 전 후 해당 시각을 `currentTimeMillis` 메소드 사용해 long 타입 변수에 저장
- 마지막에 측정된 시간 정지를 millisecond 단위로 계산해 출력

※ 하지만 `Thread.sleep` 메소드 사용 한 후 `System.current.TimeMillis` 메소드가 작동하는 시간, `finalTime` 변수에 값이 저장되는데 걸린 연산 시간 등의 영향도 존재한다.

※ millisec 단위 정도의 오차를 감안해야 한다.



4. 연습문제 6-2

4. 연습문제 6-2

6-2. 문제

: 구구단을 작성하시오

$2 * 1 = 2$	$3 * 1 = 3$	$4 * 1 = 4$	$5 * 1 = 5$	$6 * 1 = 6$	$7 * 1 = 7$	$8 * 1 = 8$	$9 * 1 = 9$
$2 * 2 = 4$	$3 * 2 = 6$	$4 * 2 = 8$	$5 * 2 = 10$	$6 * 2 = 12$	$7 * 2 = 14$	$8 * 2 = 16$	$9 * 2 = 18$
$2 * 3 = 6$	$3 * 3 = 9$	$4 * 3 = 12$	$5 * 3 = 15$	$6 * 3 = 18$	$7 * 3 = 21$	$8 * 3 = 24$	$9 * 3 = 27$
$2 * 4 = 8$	$3 * 4 = 12$	$4 * 4 = 16$	$5 * 4 = 20$	$6 * 4 = 24$	$7 * 4 = 28$	$8 * 4 = 32$	$9 * 4 = 36$
$2 * 5 = 10$	$3 * 5 = 15$	$4 * 5 = 20$	$5 * 5 = 25$	$6 * 5 = 30$	$7 * 5 = 35$	$8 * 5 = 40$	$9 * 5 = 45$
$2 * 6 = 12$	$3 * 6 = 18$	$4 * 6 = 24$	$5 * 6 = 30$	$6 * 6 = 36$	$7 * 6 = 42$	$8 * 6 = 48$	$9 * 6 = 54$
$2 * 7 = 14$	$3 * 7 = 21$	$4 * 7 = 28$	$5 * 7 = 35$	$6 * 7 = 42$	$7 * 7 = 49$	$8 * 7 = 56$	$9 * 7 = 63$
$2 * 8 = 16$	$3 * 8 = 24$	$4 * 8 = 32$	$5 * 8 = 40$	$6 * 8 = 48$	$7 * 8 = 56$	$8 * 8 = 64$	$9 * 8 = 72$
$2 * 9 = 18$	$3 * 9 = 27$	$4 * 9 = 36$	$5 * 9 = 45$	$6 * 9 = 54$	$7 * 9 = 63$	$8 * 9 = 72$	$9 * 9 = 81$

2	*	1	=	2								3	*	1	=	3
2	*	2	=	4								3	*	2	=	6
2	*	3	=	6								3	*	3	=	9
2	*	4	=	8								3	*	4	=	12
2	*	5	=	10								3	*	5	=	15
2	*	6	=	12								3	*	6	=	18
2	*	7	=	14								3	*	7	=	21
2	*	8	=	16								3	*	8	=	24
2	*	9	=	18								3	*	9	=	27



공백 6칸

4. 연습문제 6-2

```
PracProb6_2.java ×
3 ▶ public class PracProb6_2 {
4   // 구구단을 작성하시오
5 ▶ public static void main(String[] args) {
6     for (int i = 1; i < 10 ; i++) {
7       for (int j = 2; j < 10 ; j++) {
8         if (j < 9)
9           System.out.printf("%d * %d = %2d%6s",j, i, i * j," ");
10        else
11          System.out.printf("%d * %d = %2d\n",j, i, i * j);
12      }
13    }
14  }
15 }
```

```
PracProb6_2 ×
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2019
2 * 1 = 2    3 * 1 = 3    4 * 1 = 4    5 * 1 = 5    6 * 1 = 6    7 * 1 = 7    8 * 1 = 8    9 * 1 = 9
2 * 2 = 4    3 * 2 = 6    4 * 2 = 8    5 * 2 = 10   6 * 2 = 12   7 * 2 = 14   8 * 2 = 16   9 * 2 = 18
2 * 3 = 6    3 * 3 = 9    4 * 3 = 12   5 * 3 = 15   6 * 3 = 18   7 * 3 = 21   8 * 3 = 24   9 * 3 = 27
2 * 4 = 8    3 * 4 = 12   4 * 4 = 16   5 * 4 = 20   6 * 4 = 24   7 * 4 = 28   8 * 4 = 32   9 * 4 = 36
2 * 5 = 10   3 * 5 = 15   4 * 5 = 20   5 * 5 = 25   6 * 5 = 30   7 * 5 = 35   8 * 5 = 40   9 * 5 = 45
2 * 6 = 12   3 * 6 = 18   4 * 6 = 24   5 * 6 = 30   6 * 6 = 36   7 * 6 = 42   8 * 6 = 48   9 * 6 = 54
2 * 7 = 14   3 * 7 = 21   4 * 7 = 28   5 * 7 = 35   6 * 7 = 42   7 * 7 = 49   8 * 7 = 56   9 * 7 = 63
2 * 8 = 16   3 * 8 = 24   4 * 8 = 32   5 * 8 = 40   6 * 8 = 48   7 * 8 = 56   8 * 8 = 64   9 * 8 = 72
2 * 9 = 18   3 * 9 = 27   4 * 9 = 36   5 * 9 = 45   6 * 9 = 54   7 * 9 = 63   8 * 9 = 72   9 * 9 = 81
Process finished with exit code 0
```




5. 연습문제 6-3

5. 연습문제 6-3

6-3. 문제

: 전자시계 프로그램을 작성하시오

1:00	2:00	3:00	4:00	5:00	6:00	7:00	8:00	9:00	10:00	11:00	12:00
1:01	2:01	3:01	4:01	5:01	6:01	7:01	8:01	9:01	10:01	11:01	12:01
1:02	2:02	3:02	4:02	5:02	6:02	7:02	8:02	9:02	10:02	11:02	12:02
1:03	2:03	3:03	4:03	5:03	6:03	7:03	8:03	9:03	10:03	11:03	12:03
1:04	2:04	3:04	4:04	5:04	6:04	7:04	8:04	9:04	10:04	11:04	12:04
1:05	2:05	3:05	4:05	5:05	6:05	7:05	8:05	9:05	10:05	11:05	12:05
...											
1:59	2:59	3:59	4:59	5:59	6:59	7:59	8:59	9:59	10:59	11:59	12:59

5. 연습문제 6-3

```
PracProb6_3.java ×
3 ▶ public class PracProb6_3 {
4 ▶   public static void main(String[] args) {
5     for (int i = 0; i < 60; i++) {
6       for (int j = 1; j < 13; j++) {
7         if (j < 12)
8           System.out.printf("%d:%02ds", j, i, " ");
9         else
10          System.out.printf("%d:%02d\n", j, i);
11       }
12     }
13   }
14 }
```

```
PracProb6_3 ×
"C:\Program Files\Java\jdk-13.0.2\bin\java.exe" "-javaagent:C:\V
1:00 2:00 3:00 4:00 5:00 6:00 7:00 8:00 9:00 10:00 11:00 12:00
1:01 2:01 3:01 4:01 5:01 6:01 7:01 8:01 9:01 10:01 11:01 12:01
1:02 2:02 3:02 4:02 5:02 6:02 7:02 8:02 9:02 10:02 11:02 12:02
1:03 2:03 3:03 4:03 5:03 6:03 7:03 8:03 9:03 10:03 11:03 12:03
1:04 2:04 3:04 4:04 5:04 6:04 7:04 8:04 9:04 10:04 11:04 12:04
1:05 2:05 3:05 4:05 5:05 6:05 7:05 8:05 9:05 10:05 11:05 12:05
1:06 2:06 3:06 4:06 5:06 6:06 7:06 8:06 9:06 10:06 11:06 12:06
1:07 2:07 3:07 4:07 5:07 6:07 7:07 8:07 9:07 10:07 11:07 12:07
1:08 2:08 3:08 4:08 5:08 6:08 7:08 8:08 9:08 10:08 11:08 12:08
1:09 2:09 3:09 4:09 5:09 6:09 7:09 8:09 9:09 10:09 11:09 12:09
1:10 2:10 3:10 4:10 5:10 6:10 7:10 8:10 9:10 10:10 11:10 12:10
1:11 2:11 3:11 4:11 5:11 6:11 7:11 8:11 9:11 10:11 11:11 12:11
1:12 2:12 3:12 4:12 5:12 6:12 7:12 8:12 9:12 10:12 11:12 12:12
```

5. 연습문제 6-3

```
PracProb6_3 ×
↑
↓
↺
↻
🖨
🗑
1:12 2:12 3:12 4:12 5:12 6:12 7:12 8:12 9:12 10:12 11:12 12:12
1:13 2:13 3:13 4:13 5:13 6:13 7:13 8:13 9:13 10:13 11:13 12:13
1:14 2:14 3:14 4:14 5:14 6:14 7:14 8:14 9:14 10:14 11:14 12:14
1:15 2:15 3:15 4:15 5:15 6:15 7:15 8:15 9:15 10:15 11:15 12:15
1:16 2:16 3:16 4:16 5:16 6:16 7:16 8:16 9:16 10:16 11:16 12:16
1:17 2:17 3:17 4:17 5:17 6:17 7:17 8:17 9:17 10:17 11:17 12:17
1:18 2:18 3:18 4:18 5:18 6:18 7:18 8:18 9:18 10:18 11:18 12:18
1:19 2:19 3:19 4:19 5:19 6:19 7:19 8:19 9:19 10:19 11:19 12:19
1:20 2:20 3:20 4:20 5:20 6:20 7:20 8:20 9:20 10:20 11:20 12:20
1:21 2:21 3:21 4:21 5:21 6:21 7:21 8:21 9:21 10:21 11:21 12:21
1:22 2:22 3:22 4:22 5:22 6:22 7:22 8:22 9:22 10:22 11:22 12:22
1:23 2:23 3:23 4:23 5:23 6:23 7:23 8:23 9:23 10:23 11:23 12:23
1:24 2:24 3:24 4:24 5:24 6:24 7:24 8:24 9:24 10:24 11:24 12:24
1:25 2:25 3:25 4:25 5:25 6:25 7:25 8:25 9:25 10:25 11:25 12:25
1:26 2:26 3:26 4:26 5:26 6:26 7:26 8:26 9:26 10:26 11:26 12:26
1:27 2:27 3:27 4:27 5:27 6:27 7:27 8:27 9:27 10:27 11:27 12:27
1:28 2:28 3:28 4:28 5:28 6:28 7:28 8:28 9:28 10:28 11:28 12:28
1:29 2:29 3:29 4:29 5:29 6:29 7:29 8:29 9:29 10:29 11:29 12:29
1:30 2:30 3:30 4:30 5:30 6:30 7:30 8:30 9:30 10:30 11:30 12:30
1:31 2:31 3:31 4:31 5:31 6:31 7:31 8:31 9:31 10:31 11:31 12:31
1:32 2:32 3:32 4:32 5:32 6:32 7:32 8:32 9:32 10:32 11:32 12:32
1:33 2:33 3:33 4:33 5:33 6:33 7:33 8:33 9:33 10:33 11:33 12:33
1:34 2:34 3:34 4:34 5:34 6:34 7:34 8:34 9:34 10:34 11:34 12:34
1:35 2:35 3:35 4:35 5:35 6:35 7:35 8:35 9:35 10:35 11:35 12:35
1:36 2:36 3:36 4:36 5:36 6:36 7:36 8:36 9:36 10:36 11:36 12:36
1:37 2:37 3:37 4:37 5:37 6:37 7:37 8:37 9:37 10:37 11:37 12:37
```

```
PracProb6_3 ×
↑
↓
↺
↻
🖨
🗑
1:37 2:37 3:37 4:37 5:37 6:37 7:37 8:37 9:37 10:37 11:37 12:37
1:38 2:38 3:38 4:38 5:38 6:38 7:38 8:38 9:38 10:38 11:38 12:38
1:39 2:39 3:39 4:39 5:39 6:39 7:39 8:39 9:39 10:39 11:39 12:39
1:40 2:40 3:40 4:40 5:40 6:40 7:40 8:40 9:40 10:40 11:40 12:40
1:41 2:41 3:41 4:41 5:41 6:41 7:41 8:41 9:41 10:41 11:41 12:41
1:42 2:42 3:42 4:42 5:42 6:42 7:42 8:42 9:42 10:42 11:42 12:42
1:43 2:43 3:43 4:43 5:43 6:43 7:43 8:43 9:43 10:43 11:43 12:43
1:44 2:44 3:44 4:44 5:44 6:44 7:44 8:44 9:44 10:44 11:44 12:44
1:45 2:45 3:45 4:45 5:45 6:45 7:45 8:45 9:45 10:45 11:45 12:45
1:46 2:46 3:46 4:46 5:46 6:46 7:46 8:46 9:46 10:46 11:46 12:46
1:47 2:47 3:47 4:47 5:47 6:47 7:47 8:47 9:47 10:47 11:47 12:47
1:48 2:48 3:48 4:48 5:48 6:48 7:48 8:48 9:48 10:48 11:48 12:48
1:49 2:49 3:49 4:49 5:49 6:49 7:49 8:49 9:49 10:49 11:49 12:49
1:50 2:50 3:50 4:50 5:50 6:50 7:50 8:50 9:50 10:50 11:50 12:50
1:51 2:51 3:51 4:51 5:51 6:51 7:51 8:51 9:51 10:51 11:51 12:51
1:52 2:52 3:52 4:52 5:52 6:52 7:52 8:52 9:52 10:52 11:52 12:52
1:53 2:53 3:53 4:53 5:53 6:53 7:53 8:53 9:53 10:53 11:53 12:53
1:54 2:54 3:54 4:54 5:54 6:54 7:54 8:54 9:54 10:54 11:54 12:54
1:55 2:55 3:55 4:55 5:55 6:55 7:55 8:55 9:55 10:55 11:55 12:55
1:56 2:56 3:56 4:56 5:56 6:56 7:56 8:56 9:56 10:56 11:56 12:56
1:57 2:57 3:57 4:57 5:57 6:57 7:57 8:57 9:57 10:57 11:57 12:57
1:58 2:58 3:58 4:58 5:58 6:58 7:58 8:58 9:58 10:58 11:58 12:58
1:59 2:59 3:59 4:59 5:59 6:59 7:59 8:59 9:59 10:59 11:59 12:59

Process finished with exit code 0
```



6. 연습문제 6-4

6. 연습문제 6-4

6-4. 문제 [반지름이 1인 원 안에 다트를 던져서 원주율 구하기]

1. "던질 횟수를 입력해주세요" 를 출력한다.
2. 키보드로부터 long값을 변수 tries에 입력 받는다
3. 정수형 hits 변수를 0으로 초기화 한다
4. 입력 받은 tries의 수 만큼 for 문을 반복하여 아래 a, b를 수행한다
 - A. 다트가 꽂히는 좌표 x, y를 랜덤한 값으로 정하되 범위는 -1 ~ 1 사이의 실수(double)로 한다.
 - Hints: `new Random().nextDouble()`: 0 ~ 1 사이 실수 랜덤하게 리턴
 - B. 다트가 꽂힌 좌표가 원 안에 있을 경우 hits를 증가연산자를 사용하여 1 증가 시킨다
 - Hints: 두 점(x1, y1), (x2, y2) 사이의 거리는
 $\text{Math.sqrt}((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2))$ 를 활용
5. 반복이 끝나면 실수형 변수 piEstimate를 선언과 동시에 PI 값을 계산하여 대입하여 초기화 한다
 - A. `piEstimate = 4 * hits / tries`
6. 마지막에 PI 값의 예상값 piEstimate를 출력한다
7. 3.141592....에 가까운 값이 나오는지 확인한다

6. 연습문제 6-4

```
public static double makeCoordinate() {  
    // 4.a. 다트가 꽂히는 좌표 x, y를 랜덤한 값으로 정하되 범위는 -1 ~ 1 사이의 실수(double)로 한다  
    double var = new java.util.Random().nextDouble();  
    int sign = 1;  
    if (new java.util.Random().nextInt( bound: 2) == 0)  
        sign = -1;  
    return sign * var;  
}  
  
public static void main(String[] args) {  
    // 1. 던질 횟수를 입력해주세요 를 출력한다  
    System.out.println("던질 횟수를 입력해주세요");  
    // 2. 키보드로부터 long값을 변수 tries에 입력 받는다  
    long tries = new java.util.Scanner(System.in).nextLong();  
    // 3. 정수형 hits 변수를 0으로 초기화 한다  
    int hits = 0;  
    // 4. 입력받은 tries의 수 만큼 for 문을 반복하며 아래 a, b를 수행한다  
    for (long i = 0L; i < tries; i++) {  
        // 4.a. 다트가 꽂히는 좌표 x, y를 랜덤한 값으로 정하되 범위는 -1 ~ 1 사이의 실수(double)로 한다  
        double x = makeCoordinate();  
        double y = makeCoordinate();  
        // 4.b. 다트가 꽂힌 좌표가 원 안에 있을 경우 hits를 증가 연산자를 사용하여 1 증가 시킨다  
        if (Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2)) <= 1)  
            hits++;  
    }  
    // 5. 반복이 끝나면 실수형 변수 piEstimate를 선언과 동시에 PI 값을 계산하여 대입하여 초기화 한다.  
    double piEstimate = 4.0 * hits / tries;  
    // 6. 마지막에 PI 값의 예상값 piEstimate를 출력한다.  
    System.out.println("piEstimate = " + piEstimate);  
    // 7. 3.141592... 에 가까운 값이 나오는지 확인한다.  
}
```

makeCoordinate() 메소드

- java.util.Random().nextDouble() 메소드
: [0 ~ 1] 범위 내 임의의 실수를 균등한 확률로 return
- java.util.Random().nextInt(2) 메소드
: 0과 1을 균등한 확률로 결정하여 return
- return sign * var
: [-1 ~ 1] 범위 내 임의의 실수를 균등한 확률로 return

```
PracProb6_4 x  
↑ "C:\Program Files\Java\jdk-13.0.2\  
↓ 던질 횟수를 입력해주세요  
1000000000  
⇨ piEstimate = 3.14164896  
⇩  
Process finished with exit code 0
```

1억번 던졌을 때 piEstimate 결과

```
PracProb6_4 x  
↑ "C:\Program Files\Java\jdk-13.0.2\  
↓ 던질 횟수를 입력해주세요  
10000000000  
⇨ piEstimate = 3.141548732  
⇩  
Process finished with exit code 0
```

10억번 던졌을 때 piEstimate 결과

6. 연습문제 6-4

6-4. 추가

제시된 근사 값 = 3.141592

- tries = 1억, piEstimate = 3.14164896
 - tries = 1억, 오차 (estiError) = $|3.141592 - 3.14164896| = 5.696 \times 10^{-5}$
 - 계산 시간 = 23 ~ 30 sec
- tries = 10억, piEstimate = 3.141691924
 - tries = 10억, 오차 (estiError) = $|3.141592 - 3.141691924| = 9.9924 \times 10^{-5}$
 - 계산시간 = 228 sec
- tries = 100억, piEstimate = 3.14158467
 - tries = 100억, 오차 (estiError) = $|3.141592 - 3.14158467| = 7.33 \times 10^{-6}$
 - 계산 시간 = 3311 sec



tries > 1억 인 경우
계산에 소요되는 시간에 비해
오차 감소가 미미하다.

```
PracProb6_4revised x
i = 1000000000
Total operation time: 228 sec
piEstimate = 3.141691924
estiError = 9.992399999969592E-5
Process finished with exit code 0
```

10억번 던졌을 때 piEstimate 결과

```
PracProb6_4 x
piEstimate = 3.1415861696969696
Current operation time : 3291 sec
i = 100억
Total operation time: 3311 sec
piEstimate = 3.14158467
Process finished with exit code 0
```

100억번 던졌을 때 piEstimate 결과