

Computer Programming

컴퓨터 네트워크 연구실
유현우, 김기연

2016hycp@gmail.com

실습 계획

- Java Naming Convention
- 변수 (Variable) 선언
- 입출력 (Input / Output)
- 조건문 (Conditionals)

실습에 앞서

- 영어와 친해지자
 - 우수한 검색력은 컴공의 덕목
 - 대부분의 자료는 영어(구글신)
 - 변수?? Variable??
 - 조건문?? Conditional??
 - 다 같은 말이다.
 - 단어단위로라도 알아두자.

Java Naming Convention

- 좋은 코드란?
 - 짧은코드?
 - 긴코드?
 - 빠른코드?

Java Naming Convention

```
public class A {  
  
    public static void main(String[] args) {  
  
        int a = 1;  
  
        int b = 2;  
  
        int d;  
  
        d = c(a, b)  
  
        System.out.println(d);  
  
    }  
  
    int c(int a, int b) {  
  
        return a + b;  
  
    }  
  
}
```

```
public class SumCalculator {  
  
    public static void main(String[] args) {  
  
        int first_value = 1;  
  
        int second_value = 2;  
  
        int result;  
  
        result = addTwoInput(first_value, second_value);  
  
        System.out.println( result );  
  
    }  
  
    int addTwoInput(int input_1, int input_2) {  
  
        return input_1 + input_2;  
  
    }  
  
}
```

둘은 같은 일을 하는 코드, 어떻게 읽기 좋은가요?

Java Naming Convention

- 남이 읽기에 이해하기 쉬운코드가 좋은 코드
- 클래스 이름, 변수이름, 메서드이름,,
- Naming Convention을 꼭 지키라는 법은 없지만, 더 쉽게 읽히게 할 수 있는 법이 떠오르지 않는다면, 지키면 좋다.

Java Naming Convention

- 식별자 (Identifier)
 - 변수, 상수, 메서드, 클래스에 붙여주는 이름
- 클래스 (Class)
 - 대문자로 시작하는 명사로 시작.
 - ex) ClassName, AccountManager...
- 메서드 (Method)
 - 소문자 동사로 시작.
 - doSomething, getValue ...

Java Naming Convention

- 변수 (Variable)
 - 소문자 명사
 - value, variable, apple, book...
- 상수 (Constant Value)
 - 대문자의 명사
 - CONSTANT_VALUE...

변수 (Variable) 선언

```
/*
 * 소스 파일 : Hello.java
 */
public class Hello {

    public static int sum(int n, int m) {
        return n + m;
    }

    // main() 메소드에서 실행 시작
    public static void main(String[] args) {
        int i = 20;
        int s;
        char a;

        s = sum(i, 10); // sum() 메소드 호출
        a = '?';
        System.out.println(a); // 문자 '?' 화면 출력
        System.out.println("Hello"); // "Hello" 문자열 화면 출력
        System.out.println(s); // 정수 s 값 화면 출력
    }
}
```

클래스

메소드

메소드

이부분!

?
Hello
30

변수 (Variable) 선언

Source Code

```
1 | int level, size;
```

Source Code

```
1 | int level;  
2 | int size;
```

윗코드보다 아래처럼 작성,
선언은 한줄에 하나씩 하는게
보기좋다.

입출력 (Input, Output)

객체? 일단 넘어가자
나중에,,,

```
import java.util.Scanner; // импорт 문 필요  
...  
Scanner a = new Scanner(System.in); // Scanner 객체 생성
```

import -> 책이 많이 꽂혀 있는 책장에서 필요한 책을 가져오겠다.

입력(Input)을 처리할 수 있는
Scanner라는 형태(타입)을 담을수 있는 공간에
a라는 이름을 붙여서 만들고,
그 공간에 new(새로운) Scanner를 집어 넣겠다.

입출력 (Input, Output)

예제

```
import java.util.Scanner;

public class ScannerEx {
    public static void main(String args[]) {
        System.out.println("이름, 도시, 나이, 체중, 독신 여부를 빈칸으로 분리하여 입력하세요");

        Scanner scanner = new Scanner(System.in);
        String name = scanner.next();
        System.out.println("당신의 이름은 " + name + "입니다.");
        String city = scanner.next();
        System.out.println("당신이 사는 도시는 " + city + "입니다.");
        int age = scanner.nextInt();
        System.out.println("당신의 나이는 " + age + "살입니다.");
        double weight = scanner.nextDouble(); // 실수 토큰 읽기
        System.out.println("당신의 체중은 " + weight + "kg입니다.");
        boolean single = scanner.nextBoolean();
        System.out.println("당신은 독신 여부는 " + single + "입니다.");

        scanner.close();
    }
}
```

빨간줄 단위로
실행시켜 봅시다.

입출력 (Input, Output)

```
System.out.println( );
```

코딩을 할때 가장 많이 호출할 메서드

단순히 출력 메서드를 넘어서
디버깅(Debug : 오류수정)에
기초가 된다.

조건문 (Conditional) - if

Source Code

```
01  if (condition) {  
02      statements;  
03  }  
04  
05  if (condition) {  
06      statements;  
07  } else {  
08      statements;  
09  }  
10  
11  if (condition) {  
12      statements;  
13  } else if (condition) {  
14      statements;  
15  } else {  
16      statements;  
17  }
```

if (괄호 안의 조건이 만족되면) {

중괄호 내부의 내용을 실행한다.

} else {

괄호안의 조건 이외의 조건에는 이
부분의 내용을 실행한다.

}

조건문 (Conditional) - if

Source Code

```
01 // 아래와 같은 들여쓰기는 사용하지 말자.
02 if ((condition1 && condition2)
03     || (condition3 && condition4)
04     || !(condition5 && condition6)) { // 좋지 않은 들여쓰기
05     doSomethingAboutIt();           // 메서드 본문 시작이 명확하지가 않다.
06 }
07
08 // 대신에 아래와 같은 들여쓰기를 사용한다.
09 if ((condition1 && condition2)
10     || (condition3 && condition4)
11     || !(condition5 && condition6)) {
12     doSomethingAboutIt();
13 }
14
15 // 또는 아래와 같은 들여쓰기를 사용한다.
16 if ((condition1 && condition2) || (condition3 && condition4)
17     || !(condition5 && condition6)) {
18     doSomethingAboutIt();
19 }
```

피해야 할 코드

```
if (condition)
    statement;
```

if문은 반드시 중괄호를
포함해야한다. (에러의 가능성)

조건문 (Conditional) - if

예제

```
import java.util.Scanner;
public class Twenties {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("나이를 입력하시오:");
        int age = scanner.nextInt();
        if((age>=20) && (age<30)) {    // age가 20~29 사이인지 검사
            System.out.print("20대입니다. ");
            System.out.println("20대라서 행복합니다!");
        }
        else ← 중괄호 해야합니다!
            System.out.println("20대가 아닙니다.");

        scanner.close();
    }
}
```


조건문 (Conditional) - switch

```
switch(식) {  
    case 값1: // 식의 결과가 값1과 같을 때  
        실행 문장 1;  
        break;  
    case 값2: // 식의 결과가 값2와 같을 때  
        실행 문장 2;  
        break;  
    ...  
    case 값m:  
        실행 문장 m; // 식의 결과가 값m과 같을 때  
        break;  
    default: // 어느 것과도 같지 않을 때  
        실행 문장 n;  
}
```

switch 문에 식의 결과에 맞는
case의 statement를 실행.

break을 하지 않으면 break이
나오기 전까지 실행된다.

default는 선택사항.

조건이 아주 많지 않은이상,

if문을 더 많이 쓴다.

조건문 (Conditional) – switch

```
import java.util.Scanner;
public class Season {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("월(1~12)을 입력하시오:");
        int month = scanner.nextInt(); // 정수로 월 입력
        switch(month) {
            case 3:
            case 4:
            case 5:
                System.out.println("봄입니다.");
                break;
            case 6: case 7: case 8:
                System.out.println("여름입니다.");
                break;
            case 9: case 10: case 11:
                System.out.println("가을입니다.");
                break;
            case 12: case 1: case 2:
                System.out.println("겨울입니다."); break;
            default:
                System.out.println("잘못된 입력입니다.");
        }
        scanner.close();
    }
}
```

실습 과제

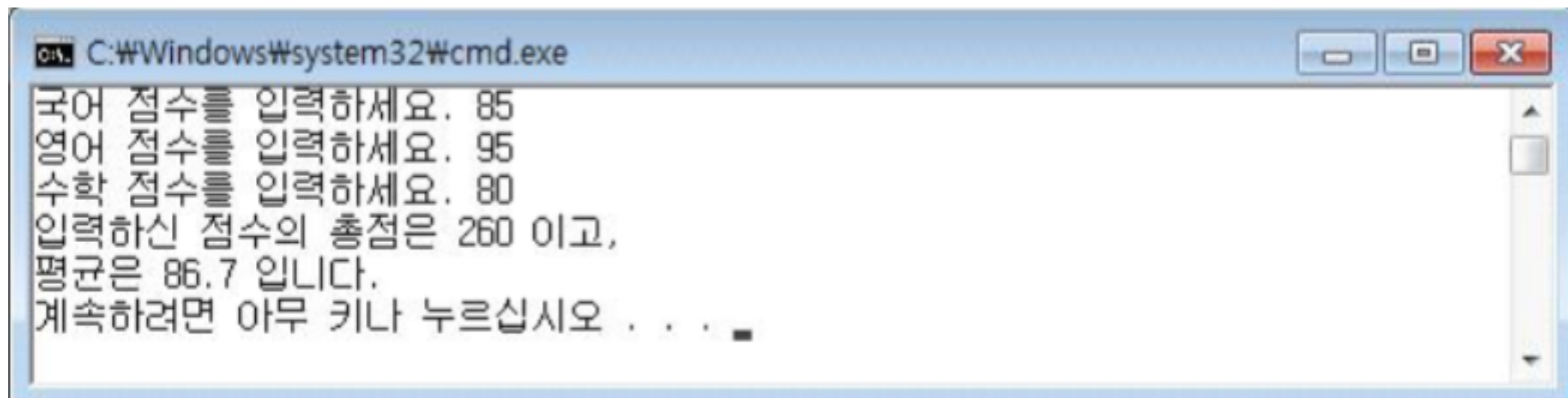
실습 1

국어, 영어, 수학 점수를 입력받아 이 점수의 총점과 평균을 계산하여 출력하라.

단, 총점 = 국어점수 + 영어점수 + 수학점수, 평균 = 총점 / 3.0 으로 계산하라.

변수는 다음과 같이 사용하라.

```
int kor;           // 국어점수
int eng;           // 영어점수
int math;          // 수학점수
int total;         // 총점
double average;    // 평균점수
```



```
C:\Windows\system32\cmd.exe
국어 점수를 입력하세요. 85
영어 점수를 입력하세요. 95
수학 점수를 입력하세요. 80
입력하신 점수의 총점은 260 이고,
평균은 86.7 입니다.
계속하려면 아무 키나 누르십시오 . . .
```

실습 과제

실습 2

파일 용량을 메가바이트 단위로 입력받고, USB 포트가 2.0인지 아닌 지를 'Y'또는 'N'으로 입력받아 이에 따라 파일 전송 시간을 초 단위로 계산하여 출력하라.

단, 계산방법은 다음과 같다.

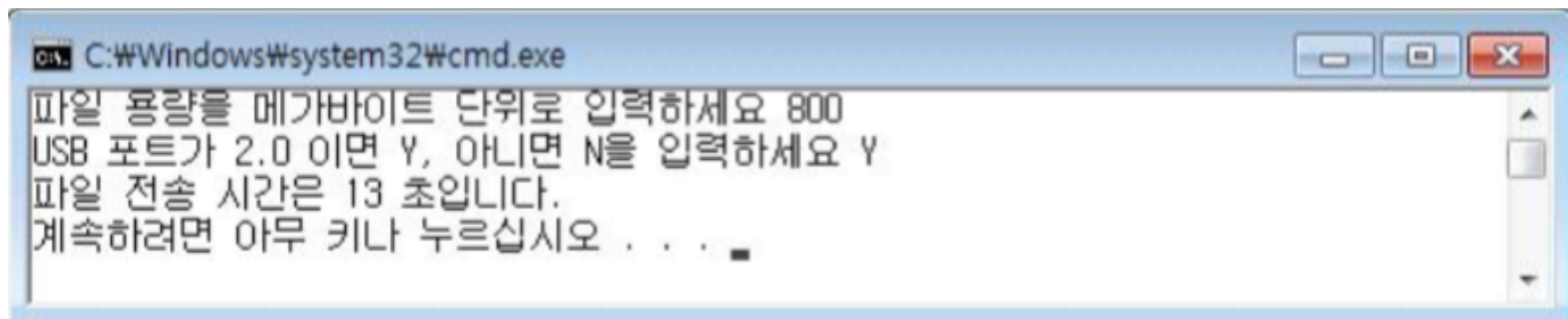
바이트 수 = 메가바이트 수 * 1024 * 1024

USB 1.1 전송 속도 = 1,500,000바이트 / 초

USB 2.0 전송 속도 = 60,000,000바이트 / 초

변수는 다음과 같이 사용하라.

int megabytes;	// 용량(메가바이트 단위)
long bytes;	// 용량(바이트 단위)
String usb2;	// USB 2.0 사용여부 (Y: 예, N: 아니요)
int time;	// 전송시간(초 단위)



```
C:\Windows\system32\cmd.exe
파일 용량을 메가바이트 단위로 입력하세요 800
USB 포트가 2.0 이면 Y, 아니면 N을 입력하세요 Y
파일 전송 시간은 13 초입니다.
계속하려면 아무 키나 누르십시오 . . .
```

실습 과제

실습 3

숫자를 3개 입력받은 후, 이 숫자 중에서 가장 큰 수, 가장 작은 수를 출력하라.

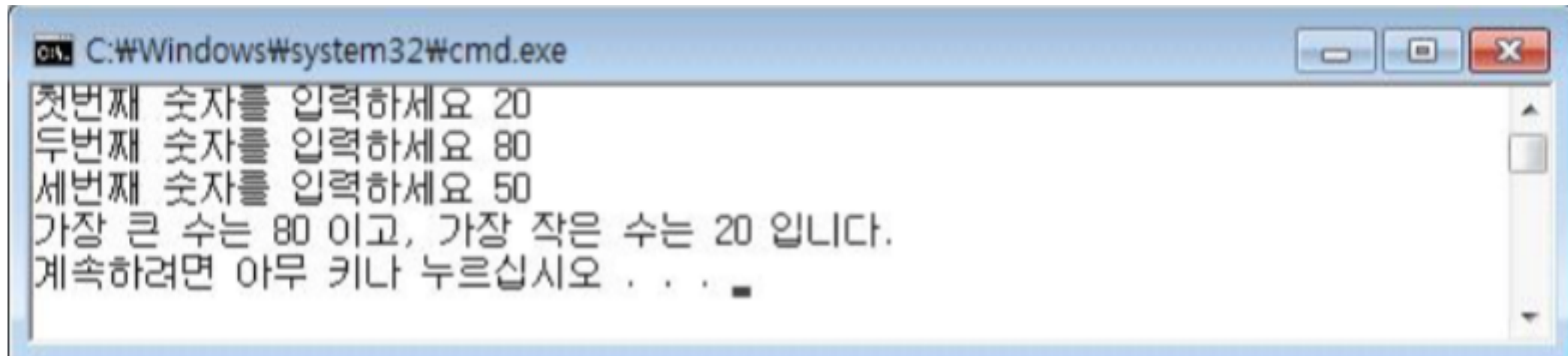
변수는 다음과 같이 사용하라.

```
int num1, num2, num3;
```

```
// 첫 번째 숫자, 두 번째 숫자, 세 번째 숫자
```

```
int max_num, min_num;
```

```
// 가장 큰 숫자, 가장 작은 숫자
```



```
C:\Windows\system32\cmd.exe
첫번째 숫자를 입력하세요 20
두번째 숫자를 입력하세요 80
세번째 숫자를 입력하세요 50
가장 큰 수는 80 이고, 가장 작은 수는 20 입니다.
계속하려면 아무 키나 누르십시오 . . .
```

중요!!!

제출 방법

1. 프로젝트파일
2. 코드 스크린샷
3. 결과화면 스크린샷
4. 코드에관한 설명 pdf

압축해서

2016hycp@gmail.com

화요일 1시~3시 => A반

화요일 3시~5시 => B반

2층 실습실 => 1반

4층 실습실 => 2반

메일제목 형식 : 학번

이름

반

실습번호

[2015000000_홍길동_A-1반_Lab01]