



(6.17)4일차

강 의 번 호	1주차
복 습	<input checked="" type="checkbox"/>
유 형	변수
자 료	https://www.theatlantic.com/international/archive/2014/12/the-real-story-of-how-america-became-an-economic-superpower/384034/
작 성 일 시	@2021년 6월 29일 오후 12:03

//1교시

[09_44_32_71]

1. 문자열

1) 문자열 표현

✓ 문자열 표현

컴퓨터에서 "기차", "출력하세요"등과 같이 단어나 문장을 문자열이라고 표현하고, ""로 묶여 있으면 문자열로 인식한다.

Java에서는 String 객체를 이용하여 저장한다.

✓ 문자열 초기화

```
String str = "기차";  
String str = new String("기차");  
String str = "기차" + "칙칙폭폭";  
String str = new String("기차" + "칙칙폭폭");  
? { String str = "기차" + 123 + 45 + "출발";  
    String str = 123 + 45 + "기차" + "출발";
```

다른 자료형 + "문자열" → 문자열
"문자열" + 다른 자료형 → 문자열

- 문자열은 ""(더블 쿼테이션)으로 묶고 String 객체를 이용하여 저장하고, 문자는 ''(싱글 쿼테이션)으로 구분 된다.
또한, String은 참조형 자료형이므로 다음과 같이 생성 할 수 있다
Type(Or class) 변수명 = new Type();
String str = new String();

- ex) 'A' → char, "A" → String(참조형)
str ="A"

//2교시

- 문자열 초기화 풀기
Q. String str = "기차"+123+45+"출발"; → A: "기차12345출발"
Q. String str = 123+45+"기차"+"출발"; → A: "168기차출발"

1. 변수의 사이즈(크기) 출력용 메소드

```
package com.test01;

public class Variable04 {
    public static void main(String[] args) {

        //준비
        //클래스명 변수명 = new 클래스명();
        Variable04 test = new Variable04();

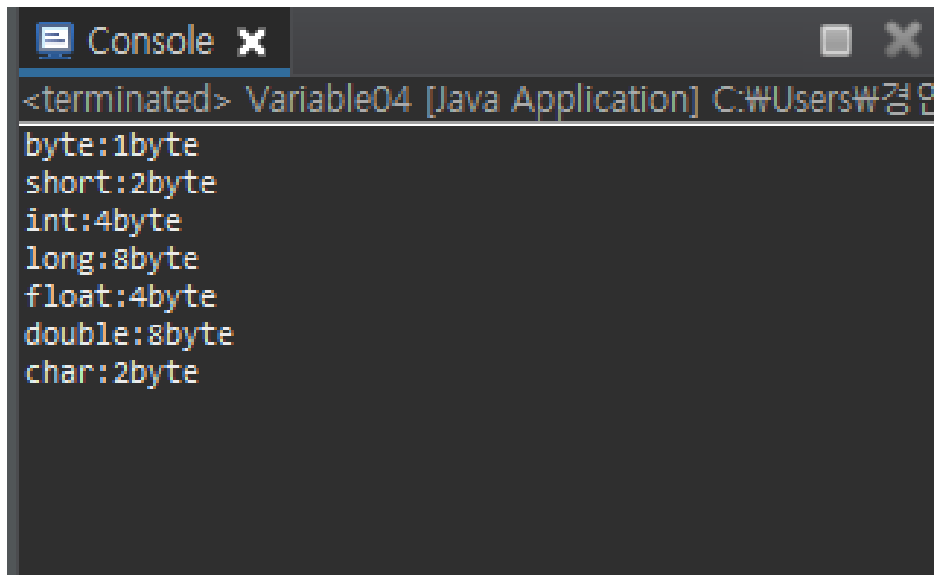
        //실행
        //변수명.메소드();
        test.printVarSize();
    }

    //변수의 사이즈(크기) 출력용 메소드
    public void printVarSize() {
        //정수형
        System.out.println("byte:" + Byte.BYTES+"byte");
        //자료형 타입별로 크기가 BYTES라는 필드로 저장되어있다.
        System.out.println("short:" + Short.BYTES+"byte");
        System.out.println("int:" + Integer.BYTES+"byte");
        System.out.println("long:" + Long.BYTES+"byte");

        //실수형
        System.out.println("float:" + Float.BYTES+"byte");
        System.out.println("double:" + Double.BYTES+"byte");

        //문자형
        System.out.println("char:" + Character.BYTES+"byte");
    }
}
```

실제 작동화면



```
<terminated> Variable04 [Java Application] C:\Users\#경인
byte:1byte
short:2byte
int:4byte
long:8byte
float:4byte
double:8byte
char:2byte
```

2. Overflow...

```
package com.test01;

public class Variable04 {
    public static void main(String[] args) {

        //준비
        //클래스명 변수명 = new 클래스명();
        Variable04 test = new Variable04();

        //실행
        //변수명.메소드();
        //test.printVarSize();

        test.testOverflow();

    }

    // //변수의 사이즈(크기) 출력용 메소드
    // public void printVarSize() {
    //     //정수형
    //     System.out.println("byte:" + Byte.BYTES+"byte");
    //     //자료형 타입별로 크기가 BYTES라는 필드로 저장되어있다.
    //     System.out.println("short:" + Short.BYTES+"byte");
    //     System.out.println("int:" + Integer.BYTES+"byte");
    //     System.out.println("long:" + Long.BYTES+"byte");
    // }
    // //실수형
    // System.out.println("float:" + Float.BYTES+"byte");
    // System.out.println("double:" + Double.BYTES+"byte");
    // }
    // //문자형
    // System.out.println("char:" + Character.BYTES+"byte");
    // }

    public void testOverflow() {
        //byte bnum = 128;

        byte bnum = 127;
    }
}
```

```

    bnum = (byte)(bnum + 1);          //byte + int--> int

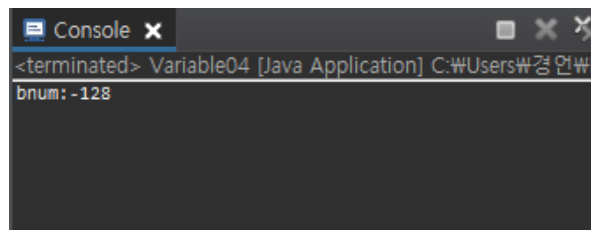
    System.out.println("bnum:" + bnum);

}

}

```

실제 작동화면



1. 형변환 Casting

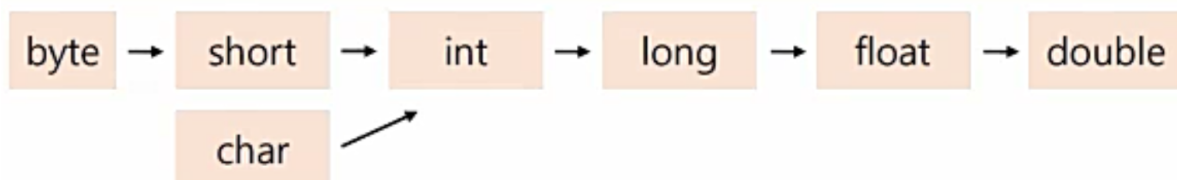
▶ 형변환(casting)

✓ 컴퓨터의 값 처리 원칙

같은 종류 자료형만 대입 가능
 같은 종류 자료형만 계산 가능
 계산의 결과도 같은 종류의 값이 나와야 함

→ 이러한 원칙이 지켜지지 않은 경우에 형 변환이 필요함

1) 자동형변환 : 연산시 컴파일러가 자동으로 형이 변환하는 것



- Byte, short, int, long 까지는 정수, 나머지는 실수(소숫점숫자)
- 화살표대로 바꿀수 있고, 역행은 안됨.

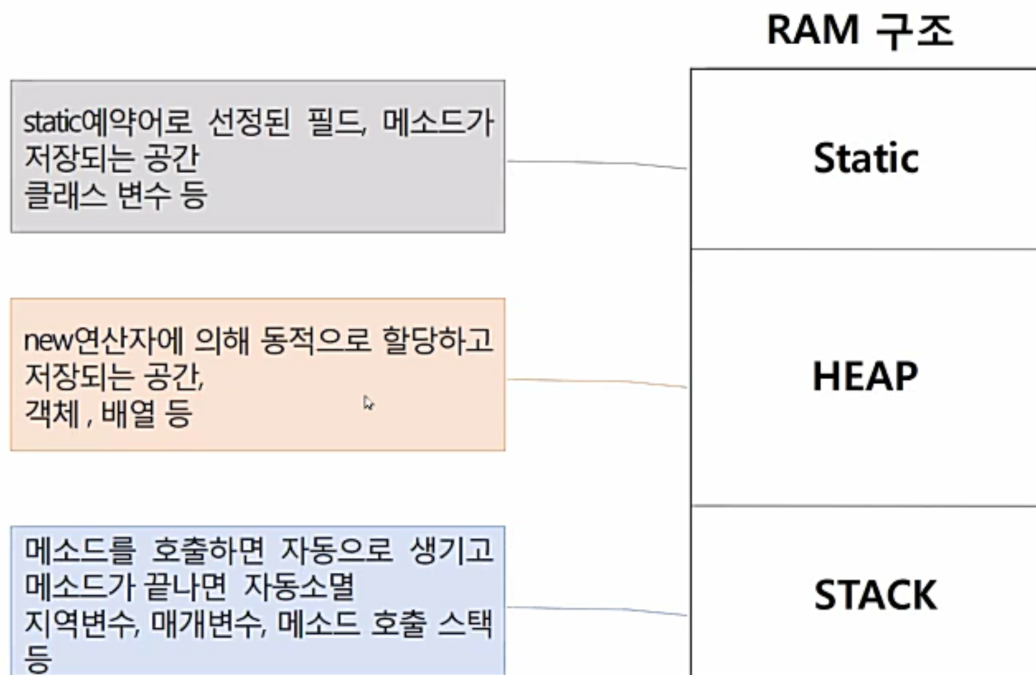
- byte와 short연산 → 결과값이 자동적으로 int값!

2) 명시적 형변환

2.

- 변수와 메모리 구조 기억하기!

변수와 메모리 구조



- 변수를 만들면 Static 영역에 생성이 된다.

//3교시

- System.out.pirnt() → 줄바꿈 x , System.out.pirntln() → 줄바꿈 o
- System.out.pirntf(형식)

1. 유니코드 출력하기

```
//6.17 수업
//7.1 복습
package com.test01;

public class Variable05 {

    public static void main(String[] args) {
        //준비
        Variable05 test = new Variable05();

        //실행
        test.test1();

    }

    public void test1() {
        boolean flag = true;
        //flag = (boolean)100; // boolean 자료형은 무조건 true, false만 들어간다(형변환 예외)

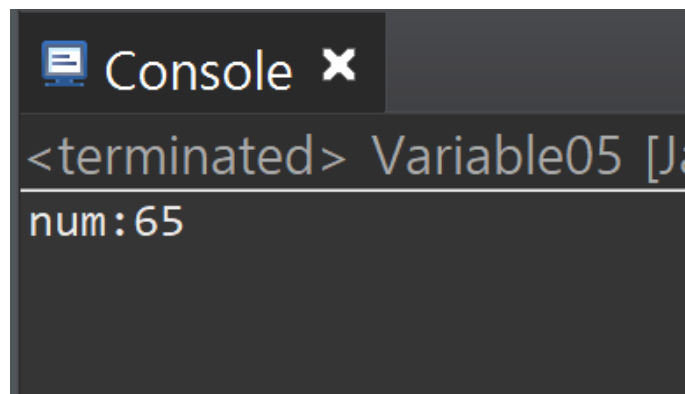
        //자료형이 다르더라도 작은 크기의 변수 타입은 큰 변수 타입으로 저장 가능(자동형변환, 묵시적형변환, upcasting)
        //큰 변수 타입은 작은 변수타입으로 자동 변환이 되지 않아 형변환을 명시해야한다. (강제형변환, 명시적형변환, downcasting)

        int num = 'A';
        System.out.println("num:" + num);
    }

}

```

실제 작동화면



```
//6.17 수업
//7.1 복습
package com.test01;

public class Variable05 {

    public static void main(String[] args) {
        //준비
        Variable05 test = new Variable05();

        //실행
        //test.test1();
    }

}

```

```

        test.test3();

    }

    public void test1() {
        boolean flag = true;
        //flag = (boolean)100; // boolean 자료형은 무조건 true, false만 들어간다(형변환 예외)

        //자료형이 다르더라도 작은 크기의 변수 타입은 큰 변수 타입으로 저장 가능(자동형변환, 묵시적형변환, upcasting)
        //큰 변수 타입은 작은 변수타입으로 자동 변환이 되지 않아 형변환을 명시해야한다. (강제형변환, 명시적형변환, downcasting)

        int num = 'A';
        System.out.println("num:" + num);

        char ch = 98;
        System.out.println("ch:" + ch);

        char ch2 = (char)num;
        System.out.println("ch2:" + ch2);

        int num2 = -98;
        char ch3 = (char)num2;
        System.out.println("ch3:" + ch3);

    }

    public void test2() {

        int inum = 10 ;
        long lnum = 100;

        //문제1 : int isum = inum + lnum; //에러
        //내가 한 풀이
        // isum = long(inum) + lnum;

        //정답 1번방식
        int isum = inum + (int)lnum;
        //2번방식
        int isum = (int)(inum + lnum);
        //3번방식
        long isum = inum + lnum;

        //문제 2
        byte bnum = 1;
        short snum = 2;

        //1번 풀이
        short sum = (short)(bnum+snum);
        //2번 풀이
        int sum2 = bnum + snum;

    }

    public void test3() {
        //문1
        long lnum = 100;

        float fnum = lnum;

        System.out.println("fnum:"+fnum);
    }

```

```

//문2
double dnum = 12.3;
//int inum = dnum; -> 타입 미스매치

//풀이 )
int inum = (int)dnum;
System.out.println("inum:"+inum);
}
}

```

//4교시

1. MethodTest01

```

//7.1 복습 시작

package com.test01;

public class MethodTest01 {
    public static void main(String[] args) {
        //
        MethodTest01.publicMethod();
        MethodTest01.protectedMethod();
        MethodTest01.defaultMethod();
        privateMethod();

        //
        MethodTest01 mt01 = new MethodTest01();
        mt01.NonStaticMethod();
    }

    //public : 접근제한자로써 어디서나 접근 가능 ( + )
    public static void publicMethod() {
        System.out.println("public method");
    }

    // 어디에서 사용이 가능하냐?
    //1) 상속일 경우 상속된 곳에서 (#)
    //2) 상속이 아닌경우에는 같은 패키지 안에서 사용가능
    protected static void protectedMethod() {
        System.out.println("protected method");
    }

    //같은 패키지 내에서
    static void defaultMethod() {
        System.out.println("defalut method");
    }

    //현재 클래스 안에서만 (-)
    private static void privateMethod() {
        System.out.println("private method");
    }

    public void NonStaticMethod() {
        System.out.println("non static method");
    }
}

```


2. MethodTest02

```
package com.test01;

public class MethodTest02 {

    public static void main(String[] args) {
        //method 호출방법
        //1. static: class.nethod();
        MethodTest01.publicMethod();
        MethodTest01.protectedMethod();
        MethodTest01.defaultMethod();
        //MethodTest01.privateMethod();

        //2.non static :
        // class 변수명 = new class();
        //변수명.method();
        MethodTest01 method01 = new MethodTest01();
        method01.NonStaticMethod();
    }
}
```

3. MethodTest03

```
package com.test02;

import com.test01.MethodTest01;

public class MethodTest03 {
    public static void main(String[] args) {
        MethodTest01.publicMethod();
        //MethodTest01.protectedMethod();
        //MethodTest01.defaultMethod();
    }
}
```

//6교시

[연습 1]

- 주어진 것
public void printUniCode(){}
키보드로 문자하나를 입력받아, 그 문자의 유니코드를 출력되게 함
입력예시) 문자 입력 : A
출력예시) A is unicode : 65
- 풀이
1) Example

```
package com.example2;

import java.util.Scanner;
```

```

public class CastingSample {
    public void printUnicode() {

        Scanner sc = new Scanner(System.in);

        System.out.println("문자를 하나 입력하세요: ") ;
        char ch = sc.next().charAt(0);

        int unicode = ch;

        System.out.println(ch + " is unicod: " + unicode);
    }
}

```

2) 메인 메소드

```

package com.example2;

public class TestMain {
    public static void main(String[] args) {
        CastingSample cs = new CastingSample();
        cs.printUnicode();
    }
}

```

[연습 2]

- 주어진 것
 - ⇒ 메소드명 : public void calculatorScore(){}
 - ⇒ 실행내용 : 실수형으로 국어, 영어, 수학 세 과목의 점수를 입력받아 총점과 평균을 계산해서 출력함. 총점과 평균은 정수형으로 처리함

입력예 :

국어 : 90.0

영어 : 90.0

수학 : 90.0

출력예 :

총점 : 270

평균 : 90

- 풀이
 - Example

```

package com.example2;

import java.util.Scanner;

public class CastingSample {

    public void calculatorScore() {
        Scanner sc = new Scanner(System.in);
    }
}

```

```

        System.out.println("국어 : ") ;
        double kor = sc.nextDouble();

        System.out.println("영어 : ") ;
        double eng = sc.nextDouble();

        System.out.println("수학 : ") ;
        double math = sc.nextDouble();

        int sum = (int)(kor + eng + math);
        int avg = sum/3;

        System.out.println("sum: " + sum);

        System.out.println("avg: " + avg);

    }
}

```

2) 메인 메소드

```

package com.example2;

public class TestMain {
    public static void main(String[] args) {
        CastingSample cs = new CastingSample();
        //cs.printUnicode();
        cs.calculatorScore();
    }
}

```

3) 실제 작동 방식