Day15 (H1)

20150828

Byte primitive type
Exception Handling
File IO

```java
class MyStudent extends Student {
    int     StID;
    String  Name;

    MyStudent() { super();}
    MyStudent(String s, int i, int x, int y , int z)
    {   super(x, y, z); setName(s); setStID(i); }

    int    getStID() { return StID; }
    String getName() { return Name; }

    void setStID( int    x ) { StID= x; }
    void setName( String x ) { Name= x; }

    void disp() {
        System.out.println( "----------------------");
        System.out.println( "Name= " + Name    );
        System.out.println( "StID= " + StID    );

        System.out.println( "Kor= " + Kor    );
        System.out.println( "Eng= " + Eng    );
        System.out.println( "Math=" + Math   );
        System.out.println( "GPA= " + Avg()  );
    }
}
```

accessor methods

mutator methods

```java
public class Student {
    int Kor;
    int Eng;
    int Math;

    Student(                    ) { setKor(0); setEng(0); setMath(0); }
    Student(int x, int y, int z) { setKor(x); setEng(y); setMath(z); }

    int getKor ()  { return Kor; }
    int getEng ()  { return Eng; }
    int getMath()  { return Math; }

    void setKor ( int x ) { ..
    void setEng ( int x ) { ..
    void setMath( int x ) { ..

    double Avg() { return (Kor+Eng+Math) / 3.0; }

    void disp() {
        System.out.println( "----------------------");
        System.out.println( "Kor= " + Kor    );
        System.out.println( "Eng= " + Eng    );
        System.out.println( "Math=" + Math   );
        System.out.println( "GPA= " + Avg()  );
    }
```

물려주는 클래스

Super class의

disp()

```java
    static void avg_mode( Student[] X, int mode) {..
```

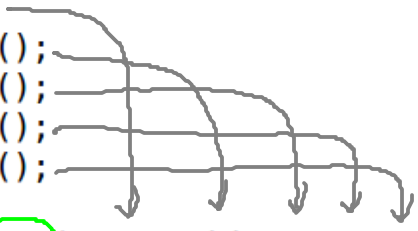
```java
class MyStudent extends Student {
    int    StID;
    String  Name;

    MyStudent() { super();}
    MyStudent(String s, int i, int x, int y , int z)
    {   super(x, y, z); setName(s); setStID(i); }

    int    getStID() { return StID; }
    String getName() { return Name; }

    void setStID( int    x ) { StID= x; }
    void setName( String x ) { Name= x; }

    void disp() {
        System.out.println( "----------------------");
        System.out.println( "Name= " + Name    );
        System.out.println( "StID= " + StID    );

        System.out.println( "Kor= " + Kor    );
        System.out.println( "Eng= " + Eng    );
        System.out.println( "Math=" + Math   );
        System.out.println( "GPA= " + Avg()  );
    }
}
```

물려받는 클래스

Sub class의

disp()

```java
public static void main(String[] args) throws IOException {
    // TODO Auto-generated method stub

    FileInputStream in = new FileInputStream("st.dat");
    Scanner         sn = new Scanner         ( in );

    MyStudent[]      S = new MyStudent[5]; // S[i] : reference var

    int i; String name; int id, x, y, z;

    for (i=0; i<S.length; ++i) {
        name = sn.next();
        id   = sn.nextInt();
        x    = sn.nextInt();
        y    = sn.nextInt();
        z    = sn.nextInt();

        S[i] = new MyStudent( name, id, x, y, z );
    }
```



```java
    S[0].disp();
    S[1].disp();
    S[2].disp();
    S[3].disp();
    S[4].disp();

    Student.avg_mode( S, 0);
    Student.avg_mode( S, 1);
    Student.avg_mode( S, 2);
```

```
young@young-Samsung-NB-System:~/workspace/Day15$
young@young-Samsung-NB-System:~/workspace/Day15$
young@young-Samsung-NB-System:~/workspace/Day15$ ls
bin  src  st.dat
young@young-Samsung-NB-System:~/workspace/Day15$ more st.dat
"Park"  20150001 99 45 50
"Kim"   20150002 88 55 80
"Lee"   20150003 77 65 90
"Baker" 20150004 66 75 80
"John"  20150005 55 85 90
```

format( )

```
void disp() {
    System.out.format( "----------------------\n");
    System.out.format( "Name= \t %8s\n", Name    );
    System.out.format( "StID= \t %8d\n", StID    );

    System.out.format( "Kor=  \t %8d\n", Kor     );
    System.out.format( "Eng=  \t %8d\n", Eng     );
    System.out.format( "Math= \t %8d\n", Math    );
    System.out.format( "GPA=  \t %8.2f \n", Avg()  );
}
```

%s String

%d 정수형

%f 소숫점

%8s <-- 8칸 -->

%8d <-- 8칸 -->

%8f <-- 8칸 -->

\t tab (default로 8칸)

\n newline

%8.2f 소숫점 이하 2자리

8칸

```
----------------------
Name=        "Park"
StID=      20150001
Kor=             99
Eng=             45
Math=            50
GPA=          64.67
----------------------
```

2칸

```java
public static void readRecord( MyStudent[] S ) throws IOException {
    FileInputStream in = new FileInputStream("st.dat");
    Scanner         sn = new Scanner         ( in );

    int i;

    for (i=0; i<S.length; ++i) {

        S[i] = new MyStudent(
                    sn.next(),      // name
                    sn.nextInt(),   // id
                    sn.nextInt(),   // Kor
                    sn.nextInt(),   // Eng
                    sn.nextInt()    // Math
                    );
    }


}

public static void main(String[] args) throws IOException {
    // TODO Auto-generated method stub

    MyStudent[]      S = new MyStudent[5]; // S[i] : reference var

    try {
        readRecord( S );
    } catch (IOException e) {
        System.out.println( "st.dat cannot be found...");
    }

    S[0].disp();
    S[1].disp();
    S[2].disp();
    S[3].disp();
    S[4].disp();

    Student.avg_mode( S, 0);
    Student.avg_mode( S, 1);
    Student.avg_mode( S, 2);
}
```

must be specified here

Can be deleted because the exception is handled here

Can throw exceptions

catch here

PrintStream

```java
import java.io.*;

class MySys {
    static PrintStream out  = new PrintStream( System.out );
}

public class PrintTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        MySys.out.println("Hello");
        MySys.out.println( 123 );

    }

}
```

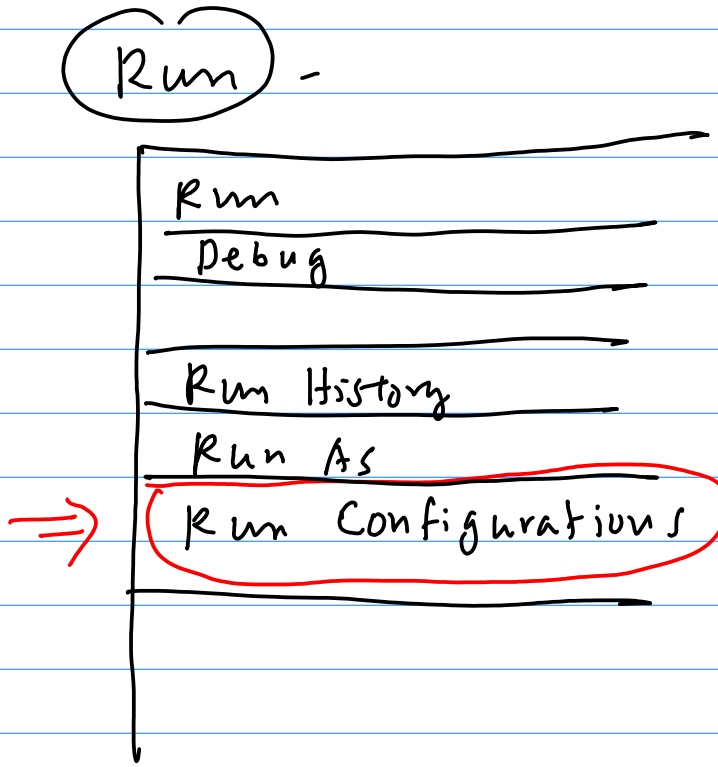
MySys.out — class MySys, static field out

System.out — static field of PrintStream class type

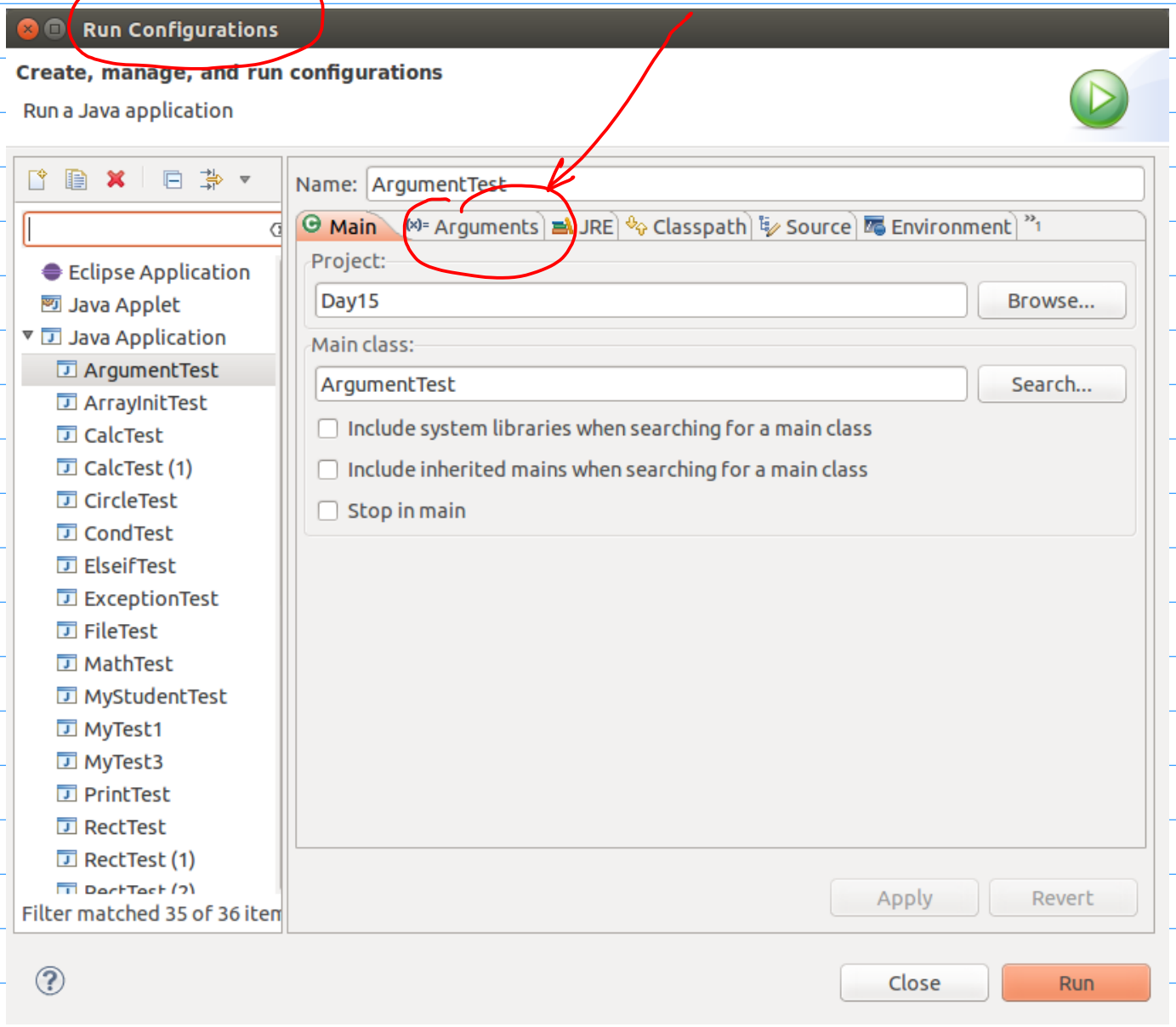
Argument Test

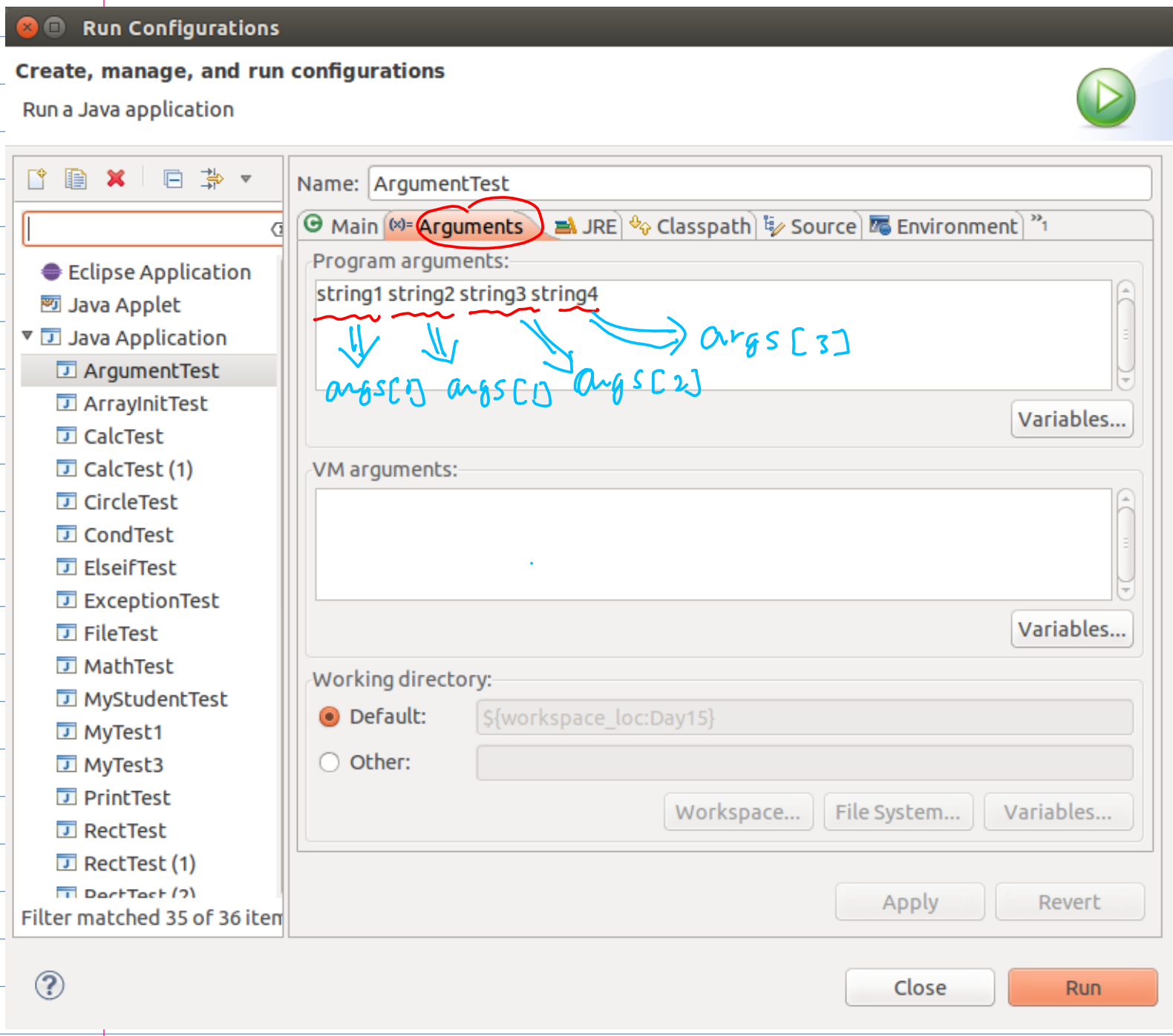
```java
public class ArgumentTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int i;
        for (i=0; i<args.length; ++i) {
            System.out.println( "args[" );
            System.out.println(    i    );
            System.out.println( "]= "    );
            System.out.println( args[i] );
        }
    }

}
```

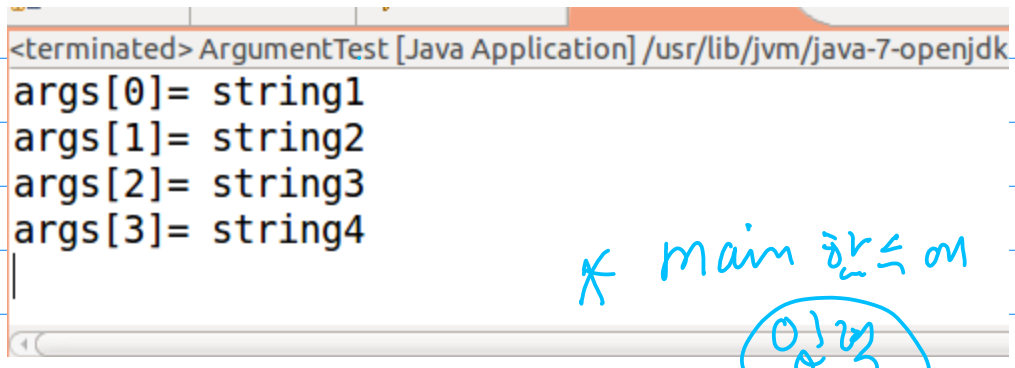
Menu Bar의 Run -



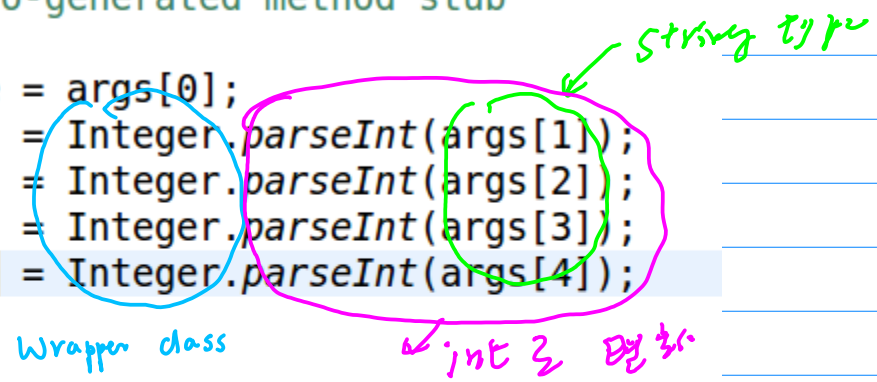
Select Arguments tab




Run Configurations
Create, manage, and run configurations
Run a Java application
Name: ArgumentTest
Main
Arguments
JRE
Classpath
Source
Environment
Program arguments:
string1 string2 string3 string4
args[3]
args[0]
args[1]
args[2]
Variables...
VM arguments:
Variables...
Working directory:
Default:
${workspace_loc:Day15}
Other:
Workspace...
File System...
Variables...
Apply
Revert
Close
Run
Eclipse Application
Java Applet
Java Application
ArgumentTest
ArrayInitTest
CalcTest
CalcTest (1)
CircleTest
CondTest
ElseifTest
ExceptionTest
FileTest
MathTest
MyStudentTest
MyTest1
MyTest3
PrintTest
RectTest
RectTest (1)
Filter matched 35 of 36 item


<terminated> ArgumentTest [Java Application] /usr/lib/jvm/java-7-openjdk
args[0]= string1
args[1]= string2
args[2]= string3
args[3]= string4

```java
public static void main(String[] args) {
    // TODO Auto-generated method stub

    String name = args[0];
    int    id   = Integer.parseInt(args[1]);
    int    kor  = Integer.parseInt(args[2]);
    int    eng  = Integer.parseInt(args[3]);
    int    math = Integer.parseInt(args[4]);

    System.out.println( "name= " + name   );
    System.out.println( "id  = " + id     );
    System.out.println( "kor = " + kor    );
    System.out.println( "eng = " + eng    );
    System.out.println( "math= " + math   );

}
```



Wrapper class

Primitive	Wrapper Class	Constructor Argument
boolean	Boolean	boolean or String
byte	Byte	byte or String
char	Character	char
int	Integer	int or String
float	Float	float, double or String
double	Double	double or String
long	Long	long or String
short	Short	short or String

Method	Purpose
parseInt(s)	returns a signed decimal integer value equivalent to string s
toString(i)	returns a new String object representing the integer i
byteValue()	returns the value of this Integer as a byte