기본 API 클래스

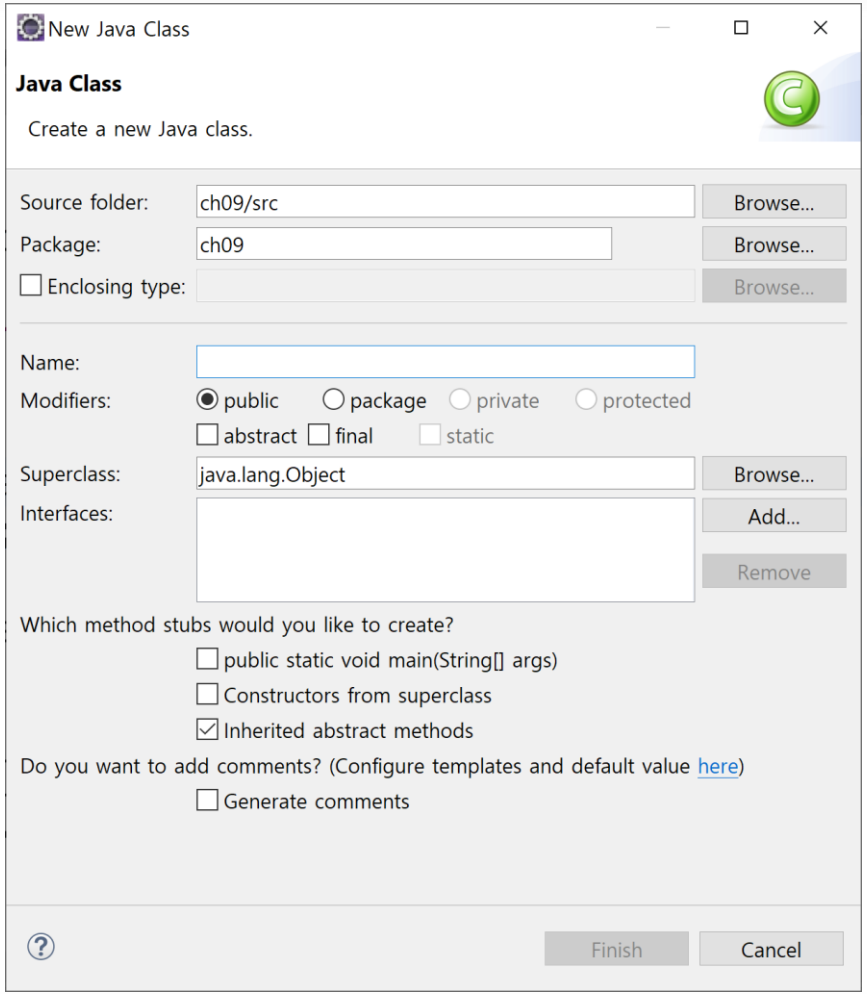
컴퓨터공학전공
박요한

수업내용

- JDK에서 제공하는 기본 API 클래스
  - ✓Object 클래스
  - ✓String, StringBuffer, StringTokenizer 클래스
  - ✓Wrapper 클래스
  - ✓Math, Random 클래스
  - ✓Arrays 클래스

Object 클래스

- 특징
  - ✓java.lang 패키지에 포함
  - ✓모든 클래스의 수퍼 클래스
    - 모든 클래스에 강제 상속
    - 모든 객체가 공통으로 가지는 객체의 속성을 나타내는 메소드 보유



Object 클래스

- 주요 메소드

메소드	설명
boolean equals(Object obj)	obj가 가리키는 객체와 현재 객체를 비교하여 같으면 true 리턴
Class getClass()	현 객체의 런타임 클래스를 리턴
int hashCode()	현 객체에 대한 해시 코드 값 리턴
String toString()	현 객체에 대한 문자열 표현을 리턴
void notify()	현 객체에 대해 대기하고 있는 하나의 스레드를 깨운다.
void notifyAll()	현 객체에 대해 대기하고 있는 모든 스레드를 깨운다.
void wait()	다른 스레드가 깨울 때까지 현재 스레드를 대기하게 한다.

Wrapper 클래스

- 자바의 기본 타입을 클래스화 한 8개 클래스

기본 타입	byte	short	int	long	char	float	double	boolean
Wrapper 클래스	Byte	Short	Integer	Long	Character	Float	Double	Boolean

✓ 이름이 Wrapper인 클래스는 존재하지 않음

✓ 용도

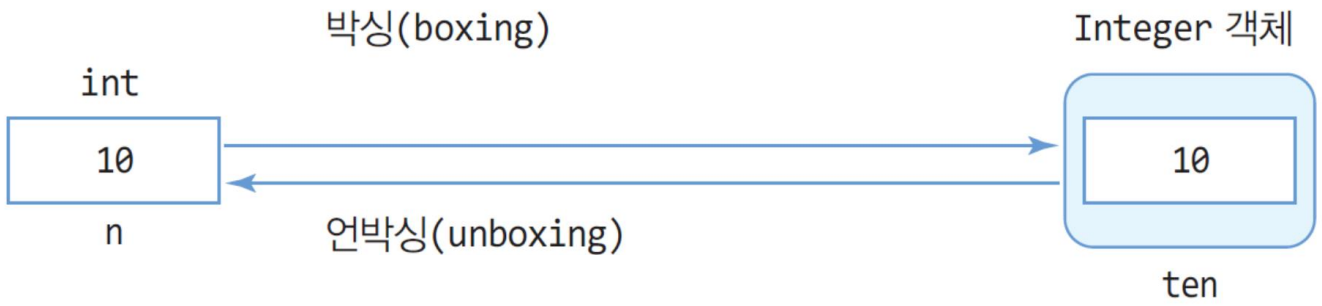
- 기본 자료형의 값을 감싸는 클래스 => 기본 타입의 값을 객체로 다룰 수 있게 함
- 자바는 객체 지향 언어, 객체만 다루는 클래스들이 존재함
- 따라서, **기본 타입의 값을 객체**로 만들 필요가 있음

```
public static void showData(Object obj)
{
    System.out.println(obj);
}
```

박싱과 언박싱

- 박싱(boxing)
  - ✓ 기본 타입의 값을 Wrapper 객체로 변환

- 언박싱(unboxing)
  - ✓ Wrapper 객체에 들어 있는 기본 타입의 값을 빼내는 것



Wrapper 객체 생성

- 기본 타입의 값으로 Wrapper 객체 생성

```
Integer i = Integer.valueOf(10);
Character c = Character.valueOf('c');
Double f = Double.valueOf(3.14);
Boolean b = Boolean.valueOf(true);
```

- 문자열로 Wrapper 객체 생성

```
Integer I = Integer.valueOf("10");
Double d = Double.valueOf("3.14");
Boolean b = Boolean.valueOf("false");
```

- Float 객체는 double 타입의 값으로 생성 가능

```
Float f = Float.valueOf((double) 3.14);
```

주요 메소드

- ✓Wrapper 객체들은 거의 유사, 많은 메소드가 static 타입
- ✓Integer 클래스의 주요 메소드

메소드	설명
static int bitCount(int i)	정수 i의 이진수 표현에서 1의 개수 리턴
float floatValue()	float 타입으로 값 리턴
int intValue()	int 타입으로 값 리턴
long longValue()	long 타입으로 값 리턴
short shortValue()	short 타입으로 값 리턴
static int parseInt(String s)	문자열 s를 10진 정수로 변환한 값 리턴
static int parseInt(String s, int radix)	문자열 s를 지정된 진법의 정수로 변환한 값 리턴
static String toBinaryString(int i)	정수 i를 이진수 표현으로 변환한 문자열 리턴
static String toHexString(int i)	정수 i를 16진수 표현으로 변환한 문자열 리턴
static String toOctalString(int i)	정수 i를 8진수 표현으로 변환한 문자열 리턴
static String toString(int i)	정수 i를 문자열로 변환하여 리턴
static Integer valueOf(int i)	정수 i를 담은 Integer 객체 리턴
static Integer valueOf(String s)	문자열 s를 정수로 변환하여 담고 있는 Integer 객체 리턴

Wrapper 활용

- Wrapper 객체로부터 기본 타입 값 알아내기

```
Integer i = Integer.valueOf(10);
int ii = i.intValue(); // ii = 10

Character c = Character.valueOf('c' );
char cc = c.charValue(); // cc = 'c'
```

```
Double f = Double.valueOf(3.14);
double dd = d.doubleValue(); // dd = 3.14

Boolean b = Boolean.valueOf(true);
boolean bb = b.booleanValue(); // bb = true
```

- 문자열을 기본 데이터 타입으로 변환

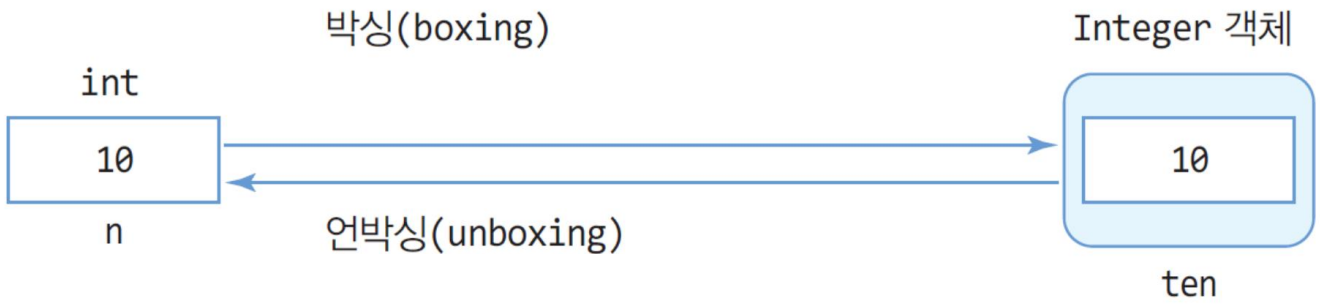
```
int i = Integer.parseInt("123");                    // i = 123
boolean b = Boolean.parseBoolean("true");           // b = true
double f = Double.parseDouble("3.14" );             // d = 3.14
```

- 기본 타입을 문자열로 변환

```
String s1 = Integer.toString(123);          // 정수 123을 문자열 "123" 으로 변환
String s2 = Integer.toHexString(123);       // 정수 123을 16진수의 문자열 "7b"로 변환
String s3 = Double.toString(3.14);          // 실수 3.14를 문자열 "3.14"로 변환
String s4 = Charater.toString('a');         // 문자 'a'를 문자열 "a"로 변환
String s5 = Boolean.toString(true);         // 불린 값 true를 문자열 "true"로 변환
```

자동 박싱과 자동언박싱

- 자동 박싱과 자동 언박싱




```
Integer ten = 10;          // 자동 박싱. Integer ten = Integer.valueOf(10);로 자동 처리
int n = ten;               // 자동 언박싱. int n = ten.intValue();로 자동 처리
```

자동 박싱과 자동언박싱

```
public static void main(String[] args)
{
    Integer num1=10;
    Integer num2=20;

    num1++;
    System.out.println(num1);

    num2+=3;
    System.out.println(num2);

    int addResult=num1+num2;
    System.out.println(addResult);

    int minResult=num1-num2;
    System.out.println(minResult);
}
```

Auto Boxing, Auto Unboxing 동시 발생

Auto Boxing, Auto Unboxing 동시 발생

Math 클래스

- 산술 연산 메소드 제공, java.lang.Math
  - ✓모든 메소드는 static 타입 : 클래스 이름으로 바로 호출해야 함

메소드	설명
static double abs(double a)	실수 a의 절댓값 리턴
static double cos(double a)	실수 a의 cosine 값 리턴
static double sin(double a)	실수 a의 sine 값 리턴
static double tan(double a)	실수 a의 tangent 값 리턴
static double exp(double a)	e^a 값 리턴
static double ceil(double a)	올림. 실수 a보다 크거나 같은 수 중에서 가장 작은 정수를 실수 타입으로 리턴
static double floor(double a)	내림. 실수 a보다 작거나 같은 수 중에서 가장 큰 정수를 실수 타입으로 리턴
static double max(double a, double b)	두 수 a, b 중에서 큰 수 리턴
static double min(double a, double b)	두 수 a, b 중에서 작은 수 리턴
static double random()	0.0보다 크거나 같고 1.0보다 작은 임의의 실수 리턴
static long round(double a)	반올림. 실수 a를 소수 첫째 자리에서 반올림한 정수를 long 타입으로 반환
static double sqrt(double a)	실수 a의 제곱근 리턴

Math 클래스

```
public class MathExample {
    public static void main(String[] args) {
        int v1 = Math.abs(-5);
        double v2 = Math.abs(-3.14);
        System.out.println("v1=" + v1);
        System.out.println("v2=" + v2);

        double v3 = Math.ceil(5.3);
        double v4 = Math.ceil(-5.3);
        System.out.println("v3=" + v3);
        System.out.println("v4=" + v4);

        double v5 = Math.floor(5.3);
        double v6 = Math.floor(-5.3);
        System.out.println("v5=" + v5);
        System.out.println("v6=" + v6);

        int v7 = Math.max(5, 9);
        double v8 = Math.max(5.3, 2.5);
        System.out.println("v7=" + v7);
        System.out.println("v8=" + v8);

        int v9 = Math.min(5, 9);
        double v10 = Math.min(5.3, 2.5);
        System.out.println("v9=" + v9);
        System.out.println("v10=" + v10);

        double v11 = Math.random();
        System.out.println("v11=" + v11);

        double v12 = Math.rint(5.3);
        double v13 = Math.rint(5.7);
        System.out.println("v12=" + v12);
        System.out.println("v13=" + v13);

        long v14 = Math.round(5.3);
        long v15 = Math.round(5.7);
        System.out.println("v14=" + v14);
        System.out.println("v15=" + v15);

        double value = 12.3456;
        double temp1 = value * 100;
        long temp2 = Math.round(temp1);
        double v16 = temp2 / 100.0;
        System.out.println("v16=" + v16);
    }
}
```

Math 클래스를 활용한 난수 발생

- 난수 발생
  - ✓static double random()
    - 0.0 이상 1.0 미만의 임의의 double 값을 반환
    - 0에서 100사이의 난수 10개 발생시키는 샘플 코드

```
for(int x=0; x<10; x++) {
   int n = (int)(Math.random()*100 + 1); // n은 [1~100] 사이의 랜덤 정수
   System.out.println(n);
}
```

- Math.random()*100은 0.0~99.99.. 사이의 실수 리턴
- Math.random()*100+1은 1.0~100.99.. 사이의 실수 값
- (int)(Math.random()*100 + 1)는 소수점이하를 제거하여 1~100 사이의 정수 값

Random 클래스를 이용한 난수 발생

- Random 클래스
  - ✓java.util.Random 클래스
  - ✓boolean, int, long, float, double 난수 입수 가능
  - ✓난수를 만드는 알고리즘에 사용되는 종자값(seed) 설정 가능
    - 종자값이 같으면 같은 난수
  - ✓Random 클래스로 부터 Random객체 생성하는 방법

생성자	설명
Random()	호출시 마다 다른 종자값(현재시간 이용)이 자동 설정된다.
Random(long seed)	매개값으로 주어진 종자값이 설정된다.

  - ✓Random 클래스가 제공하는 메소드

리턴값	메소드(매개변수)	설명
boolean	nextBoolean()	boolean 타입의 난수를 리턴
double	nextDouble()	double 타입의 난수를 리턴(0.0<= ~ < 1.0)
int	nextInt()	int 타입의 난수를 리턴(-2^{32}<= ~ <=2^{32}-1);
int	nextInt(int n)	int 타입의 난수를 리턴(0<= ~ <n)

Random 클래스 메소드 활용

```
class RandomNumberGenerator
{
   public static void main(String[] args)
   {
      Random rand=new Random();

      for(int i=0; i<100; i++)
         System.out.println(rand.nextInt(1000));
   }
}
```

```
class PseudoRandom
{
   public static void main(String[] args)
   {
      Random rand=new Random(12);

      for(int i=0; i<100; i++)
         System.out.println(rand.nextInt(1000));
   }
}
```

```
class SeedChangeRandom
{
   public static void main(String[] args)
   {
      Random rand=new Random(12);
      rand.setSeed(System.currentTimeMillis());

      for(int i=0; i<100; i++)
         System.out.println(rand.nextInt(1000));
   }
}
```