WRAPPERS

Классы оболочки (Wrapper classes)

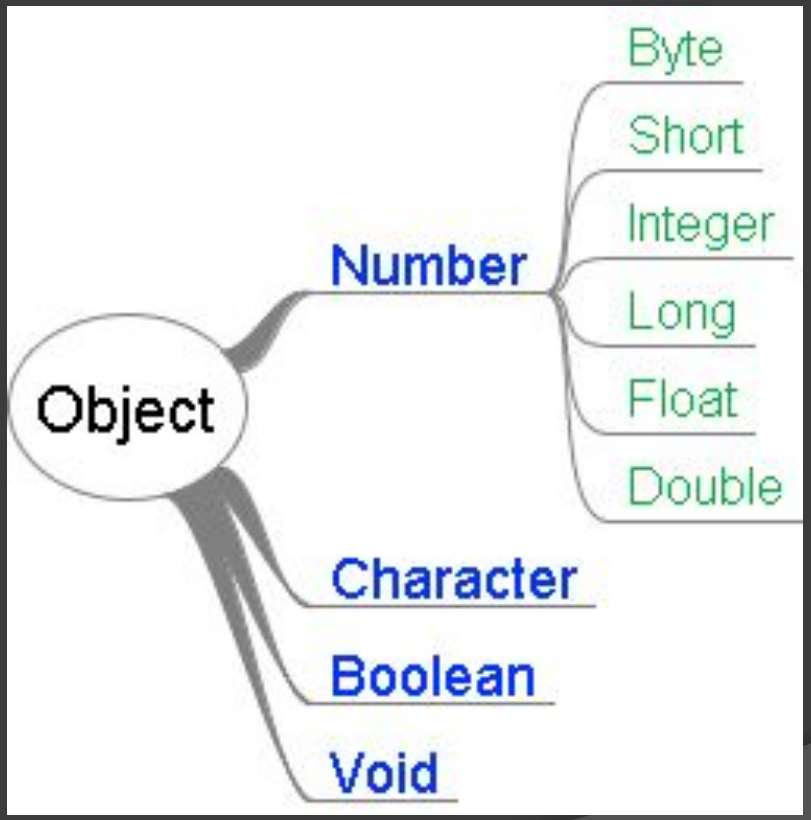
Каждый примитивный тип имеет соответствующий ему класс оболочку.

byte ==> Byte
short ==> Short
int ==> Integer
long ==> Long
char ==> Character
float ==> Float
double ==> Double
boolean ==> Boolean
void ==> Void

Модификатору void, который предназначен для того, чтобы показать, что метод не возвращает значения, поставлен в соответствие класс Void.

Иерархия классов оболочек

Классы Void, Boolean и Character наследуются напрямую от Object. Остальные - расширяют java.lang.Number.



Autoboxing

Если в операции должен участвовать объект, а участвует примитив, то этот примитив автоматически оборачивается в объектную оболочку.

Integer x = 7; Integer x = Integer.valueOf(7)

Boolean b = true Boolean b = Boolean.valueOf(true);

Unboxing

Если в операции должен участвовать примитив, то можно подставить туда объектную оболочку. Значение будет автоматически из нее развернуто.

```
Integer x = new Integer(7);
int y = x + 2;

int y = x.intValue() + 2;
```

```
Boolean bool = new Boolean(true);
boolean f = !bool ^ false;

boolean f = !bool.booleanValue() ^ false;
```

Замечание. В версиях Java до 5.0 autoboxing и unboxing для классов оболочек отсутствовали.

Начиная с 5.0 класс Boolean реализует интерфейс Comparable (true > false).

```
Boolean f1 = true;
Boolean f2 = false;
int x = f1.compareTo(f2)); // x = 1
```

Инициализация объектов классов оболочек

1) Для любого примитивного типа prime соответствующий класс оболочка PrimeWrapper содержит конструктор вида

public PrimeWrapper(prime primeValue)

2) Классы оболочки для всех примитивных типов за исключением *char* и *void* содержат конструктор вида

public PrimeWrapper(String primeValueAsString)
throws NumberFormatException

3) Создание объектов классов оболочек происходит при autobox преобразованиях (но не всегда).

Преобразование строки в число

Классы оболочки для числовых примитивных типов (byte, short, int, long, float, double) содержат статический метод parseXxx, который конвертирует строку в число соответствующего примитивного типа.

public static byte parseByte(String s)
...
public static double parseDouble(String s)

Если строка не содержит запись числа, то генерируется исключение java.lang.NumberFormatException.

Замечание. В *целочисленных* классах оболочках (Byte, Short, Int, Long)

метод parseXxx перегружен для случая, когда в строке целое число записано в определенной системе счисления, при этом основание системы указывается вторым параметром.

public static long parseLong(String s, int radix)

Замечание. Для записи *цифр* в системах счисления, основание которых больше 10, используют десятичные цифры и 26 букв латинского алфавита игнорируя регистр.

Максимальное значение основания системы счисления записано в статическом поле MAX_RADIX класса оболочки Character.

(для JDK 1.4/5/6 Character.MAX_RADIX = 36)

Метод valueOf классов оболочек

Числовые классы оболочки
(Byte, Short, Integer, Long, Float, Double)
содержат статический метод valueOf, который принимает строку в качестве параметра, конвертирует ее в соответствующее примитивное значение и возвращает оболочку.

Метод выбрасывает исключение java.lang.NumberFormatException если преобразование из строки в соответствующий тип невозможно.

```
public static Float valueOf(String s)
    throws NumberFormatException
```

Класс оболочка Boolean также содержит метод valueOf, который возвращает значение true внутри объекта Boolean если в строке записано значение true, игнорируя регистр.

В противном случае метод возвращает false внутри объекта Boolean.

public static Boolean valueOf(String s)

Классы Void и Character метод valueOf не содержат.