

第二回 独習Java ゼミ

第二章 クラスとメソッド

2. 1 メソッドの構造

2. 2 静的メソッドと静的変数の概要

2. 3 インスタンスメソッドとインスタンス変数の概要

2. 4 Integerクラス

2.1 メソッドの構造

メソッドとは

- ・プログラムステータメントの集合体
- ・Javaの基本的な実行単位
- ・クラスの一部
- ・メソッドの外部にプログラムコードを置いたり、
クラスの外部にメソッドを置くことはできない

メソッドの形式

```
retType mthName(paramList) {  
    // メソッドの本体  
}
```

retType : 戻り値の型
mthName : メソッドの名前
paramList : メソッドのパラメータリスト

パラメータリストの一般形式

```
pType0 p0 , pType1 p1 , pType0 p0 , ... , pTypeN pN
```

pType0 ~ *pTypeN* : 各パラメータの型
p0 ~ *pN* : 各パラメータ

例外とは

メソッドの実行中にランタイムエラー(実行時エラー)が発生したときに生成されるオブジェクト

メソッドの宣言の中で、例外の型を指定することができる

メソッドの例外指定

```
RetType mthName(paramList) throws exceptionList {  
    //メソッドの本体  
}
```

exceptionList: 例外の型を示すリスト

メソッドの呼び出し

```
mthName(args);
```

args: 引数のリスト
(引数が複数ある場合はカンマ(,)でくぐる)

メソッドの例

1. int型とStringのパラメータを取り、boolean型の値を返すメソッド

```
boolean process(int i, String s) {  
    // メソッドの本体  
}
```

2. 文字列パラメータを1つ取り、int型を返すメソッド
場合によってExceptionXクラスの例外

```
Int parse(String s) throws ExceptionX {  
    // メソッドの本体  
}
```

2. 2 静的メソッドと静的変数の概要

静的メソッド

- ・クラスに結びつけられているので、呼び出すためにインスタンス(オブジェクト)を作る必要がない
- ・Mathクラスによい例がある

Mathクラス

EとPIという2つの定数を提供

Eは自然対数の底e ($e=2.718281828459045$)

PIは円周率 π ($\pi=3.141592653589793$)

静的メソッドの呼び出し

```
clsName.mthName(args);
```

clsName : クラスの名前

Mathクラスで定義されている静的メソッド

メソッド	説明
<code>static double abs(double d)</code>	dの絶対値を返す
<code>static double cos(double radians)</code>	radiansの余弦(コサイン)を返す
<code>static double sin(double radians)</code>	radiansの正弦(サイン)を返す
<code>static double exp(double d)</code>	dの累乗を返す
<code>static double random()</code>	0.0 から1.0 未満の間の乱数を返す
<code>static int max(int i, int j)</code>	i と j の大きいほうの値を返す
<code>static int min(int i, int j)</code>	i と j の小さいほうの値を返す
・	・
・	・

静的変数

- ・静的メソッド同様、クラスに結びつけられているので、呼び出すためにインスタンスを作る必要がない
- ・クラスのすべてのオブジェクトに共有される

静的変数の参照

className.varName

varName : 変数の名前

静的メソッド、静的変数の使い方

プログラム

```
class MathDemo {  
    public static void main(String args[]) {  
        System.out.println("Max of -8 and -4 is " +  
            Math.max(-8, -4));  
        System.out.println("PI = " + Math.PI);  
    }  
}
```

出力結果

```
Max of -8 and -4 is -4  
PI = 3.141592653589793
```

2.3 インスタンスメソッドとインスタンス変数の概要

インスタンスメソッド

- ・ オブジェクトに結びつけられているので、呼び出すためにはインスタンスを生成する必要がある
- ・ Stringクラスによい例がある

インスタンスメソッドの呼び出し

```
objRef.mthName(args);
```

objRef : オブジェクト参照変数

Stringクラスで定義されている主なインスタンスメソッド

メソッド	説明
<code>char charAt(int i)</code>	iで指定された位置の文字を返す
<code>boolean equals(Object s)</code>	現在のオブジェクトとsが同じ文字順序なら真、そうでなければ偽を返す
<code>String substring(int start)</code>	現在のオブジェクトの部分文字列であるStringオブジェクトを返す。現在のオブジェクトのstartの位置から最後まで全文字が部分文字列となる
<code>String substring(int start, int end)</code>	現在のオブジェクトの部分文字列であるStringオブジェクトを返す。現在のオブジェクトのstartの位置からendの直前の位置までの全文字が部分文字列となる
▪	▪
▪	▪
▪	▪

2.3 インスタンスメソッドとインスタンス変数の概要

インスタンス変数

- ・ インスタンスメソッド同様、オブジェクトに結びつけられているので、呼び出すためにはインスタンスを生成する必要がある

インスタンス変数の参照

objRef.varName

varName : インスタンス変数の名前

インスタンスメソッドの使い方

プログラム

```
class StringDemo {  
    public static void main(String args[]) {  
        String s = "One Two Three Four Five Six Seven";  
        String substring = s.substring(0, 10);  
        System.out.println(substring);  
    }  
}
```

出力結果

One Two Th

2. 4 Integerクラス

integerクラス

- ・Javaのクラスライブラリの中でも特によく使われるクラスの1つ
- ・基本データ型intの値をカプセル化するもの(intのラップクラス)
- ・カプセル化されている値は変更できない
- ・MAX_VALUEとMIN_VALUEという2つの静的変数が定義されている

MAX_VALUE : 32ビットの基本データ型intが
保持できる最大値 2147483647

MIN_VALUE : 32ビットの基本データ型intが
保持できる最小値 -2147483648

- ・静的メソッドとインスタンスメソッドの両方を提供するクラスのよい例

Integerクラスで定義されている主な静的メソッド

メソッド	説明
<code>static String toBinaryString(int i)</code>	i の2進表現をStringオブジェクトで返す
<code>static String toHexString(int i)</code> す	i の16進表現をStringオブジェクトで返す
<code>static String toOctalString(int i)</code>	i の8進表現をStringオブジェクトで返す
<code>static Integer valueOf(String s)</code> throws <code>NumberFormatException</code>	s を基数10の数値として変換してIntegerオブジェクトで返す
<code>static Integer valueOf(String s, int r)</code> throws <code>NumberFormatException</code>	s を基数 r の数値として変換してIntegerオブジェクトで返す
⋮	⋮
⋮	⋮

Integerクラスで定義されている主なインスタンスメソッド

メソッド	説明
<code>double doubleValue()</code>	現在のオブジェクトの値をdouble型で返す
<code>boolean equals(Object obj)</code>	obj と現在のオブジェクトの値が同じなら真を返す
<code>float floatValue()</code>	現在のオブジェクトの値をfloat型で返す
<code>int intValue()</code>	現在のオブジェクトの値をint型で返す
<code>long longValue()</code>	現在のオブジェクトの値をlong型で返す
<code>String toString()</code>	現在のオブジェクトの値の文字列表現を返す
⋮	⋮
⋮	⋮

integerクラスの静的メソッドとインスタンスメソッドの使い方

プログラム

```
class StringToInt {  
    public static void main(String args[]) {  
        String s = "125";  
        Integer obj = Integer.valueOf(s);  
        int i = obj.intValue();  
        i += 10;  
        System.out.println(i);  
    }  
}
```

出力結果

135

2行目と3行目をまとめることもできる

```
Integer obj = Integer.valueOf(s);  
int i = obj.intValue();
```



```
int i = Integer.valueOf(s).intValue();
```

練習問題

1. 静的メソッドの練習

Mathクラスを用いて、 $\sin(\pi)$ と $\cos(\pi)$ を求め、表示させるプログラムを作りなさい。

2. 静的メソッドとインスタンスメソッドの練習

Integerクラスを用いて、2進数 $(1001001)_2$, $(110110)_2$ の2つの値を10進数に変換して、和を求めるプログラムを作成しなさい。
また、求めた値を2進数に変換して表示させなさい。