



独習Javaゼミ

2.8 : StringBufferクラス

2.9 : オブジェクトの配列

2.10: コマンドライン引数

2.11: Systemクラス



StringBufferクラス (I)

- StringBufferクラスには以下のコンストラクタがある。

①StringBuffer()

→バッファ容量を16文字で初期化。

②StringBuffer(int *size*)

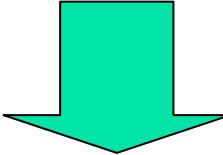
→指定の文字数で初期化。

③StringBuffer(String *s*)

→*s*の内容で初期化し、更に拡張用の16文字を余分に確保。

StringBufferクラス(Ⅱ)

- StringBufferオブジェクトが使う内部バッファは必要に応じて自動的に増やされる。

しかし 

- 余計な付加がかかるので多めのスペースを自動的に確保するようになっている。



StringBufferクラス (Ⅲ)

- 以下の例では、StringBufferオブジェクトを3つのコンストラクタによって生成し、各バッファの現在の容量とサイズを表示。

```
class StringBufferDemo{
    public static void main(String args[]){
        StringBuffer sb1 = new StringBuffer();
        StringBuffer sb2 = new StringBuffer(30);
        StringBuffer sb3 = new StringBuffer("abcde");

        System.out.println("sb1.capacity = " + sb1.capacity());
        System.out.println("sb1.length = " + sb1.length());

        System.out.println("sb2.capacity = " + sb2.capacity());
        System.out.println("sb2.length = " + sb2.length());

        System.out.println("sb3.capacity = " + sb3.capacity());
        System.out.println("sb3.length = " + sb3.length());
    }
}
```

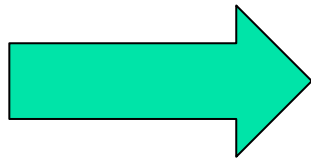
```
sb1.capacity = 16
sb1.length = 0
sb2.capacity = 30
sb1.length = 0
sb3.capacity = 21
sb3.length = 5
```



オブジェクトの配列 (I)

「手順」

- ①配列を宣言する.
- ②配列要素のスペースを確保する.
- ③要素を初期設定する.



作成手順は基本データ型と同様.



オブジェクトの配列 (Ⅱ)

- 「例」...五個の文字列から成る配列

```
class StringArray{  
    public static void main(String args[]){  
        StringArray[] = new String[5];  
        array[0] = "String 0";  
        array[1] = "String 1";  
        array[2] = "String 2";  
        array[3] = "String 3";  
        array[4] = "String 4";  
  
        System.out.println(array.length);  
        System.out.println(array[0]);  
        System.out.println(array[1]);  
        System.out.println(array[2]);  
        System.out.println(array[3]);  
        System.out.println(array[4]);  
    }  
}
```

5
<i>String0</i>
<i>String1</i>
<i>String2</i>
<i>String3</i>
<i>String4</i>



コマンドライン引数 (I)

- 「解説」

- Javaプログラムにはmain()という静的メソッドが存在する。これらは引数を1つ取る。 →Stringオブジェクト配列

- 「コマンドライン引数を取得する方法」

- コマンドライン引数の個数はargs.lengthという式で取得。

→int型

- 個々の引数はargs[0]、args[1]、args[2]等で操作。

コマンドライン引数 (Ⅱ)

- 「例」...コマンドライン引数を取得して表示。

```
class CommandLineArguments{
  public static void main(String args[]){
    System.out.println("args.length = " + args.length);
    System.out.println("args[0] = " + args[0]);
    System.out.println("args[1] = " + args[1]);
    System.out.println("args[2] = " + args[2]);
  }
}
```

- 「呼び出し方法 & 出力」

```
java CommandLineArguments 1 2 abcde
```

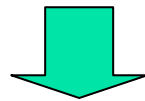
```
arg s.length = 3
arg s[0] = 1
arg s[1] = 2
args [2] = abcd
```



Systemクラス(I)

- 「解説」

- Systemクラスにはランタイム環境に関する属性が定義されている。



outという静的変数

- この変数にはPrintStreamオブジェクトへの参照が入っていて、そのオブジェクトprint()とprintln()メソッドで標準出力に文字列引数を表示する事になる。



Systemクラス(Ⅱ)

- 「静的メソッドの例」

- `void exit(int code)`

...現在のプロセスを終了する静的メソッド。

codeは終了コード.0(ゼロ)で正常終了を表す。

それ以外の値は問題を報告する為に利用可能。

- `long currentTimeMillis()`

...1970年1月1日午前0時(グリニッジ標準時)から経過

時間を返す。



Systemクラス(Ⅲ)

- 「例」... 五個の要素がarray1からarray2へコピーされ、その後array2の10個の要素が表示される。

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

    int array1[] = {0,1,2,3,4,5,6,7,8,9};
    int array2[] = {0,0,0,0,0,0,0,0,0,0};

    System.arraycopy(array1,0,array2,0,5);
    System.out.print("array2: ");
    System.out.print(array2[0] + " ");
    System.out.print(array2[1] + " ");
    System.out.print(array2[2] + " ");
    System.out.print(array2[3] + " ");
    System.out.print(array2[4] + " ");
    System.out.print(array2[5] + " ");
    System.out.print(array2[6] + " ");
    System.out.print(array2[7] + " ");
    System.out.print(array2[8] + " ");
    System.out.print(array2[9]);
}
}
```

array2: 0 1 2 3 4 0 0 0 0 0



宿題

- StringBufferオブジェクトを作成し、文字を付加して、文字列バッファの容量とサイズを表示しなさい。