

プログラミング基礎

第5.5回

補足 (static)

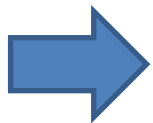
+

String型

補足 (static)

- mainメソッドのあるクラスに, メソッドを作る場合はstaticを付ける

```
public class 何かのプログラム
{
    static public void main(String[] args)
    {
        ...
    }
    static void 何かをするメソッド()
    {
        ...
    }
}
```

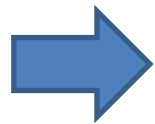


補足 (static)

- 自分で作った型 (クラス) にメソッドを作る場合はstaticを付けない

```
public class 何かのプログラム
{
    static public void main(String[] args)
    {
        ...
    }
}
```

```
class 何かのクラス
{
    void 何かをするメソッド()
    {
        ...
    }
}
```



* 本来付けることは可能だが今は上記の通りとしておく

補足 (static)

- もちろんstaticには重要な役割があるが、その意味については後日

String型

- int型 : 整数の値を記憶するための型
- double型 : 小数を含む値を記憶するための型
- **String型 : 文字列を記憶するための型
(クラス)**

String型

```
static void main(String[] args)
{
    String str = "test";
    System.out.println(str);
}
```

```
static void main(String[] args)
{
    System.out.println("test");
}
```

```
static String returnTestString()
{
    String str = "test";
    return str;
}

static void main(String[] args)
{
    String str = returnTestString();
    System.out.println(str);
}
```

いずれも結果は同じ

```
static String returnTestString()
{
    return "test";
}

static void main(String[] args)
{
    System.out.println(returnTestString());
}
```

String型

- 足し算(加算演算子)が使用可能

```
public class StringTest
{
    public static void main(String[] args)
    {
        String str1 = "I";
        String space = " ";
        String strAm = "am";
        String strlam;

        strlam = str1 + space + strAm + ".";


        System.out.println(strlam);
    }
}
```

実行結果:
>java StringTest
I am.

直接""で囲まれた文字列を記述した場合、String型の定数ができる

String型

- 文字列の足し算に含まれる他の型は、自動的にString型へ変換される

```
public class StringTest
{
    public static void main(String[] args)
    {
        int years = 10;
        String ago = " years ago";
        String str;
        str = years + ago;
        自動的に文字列"10"へ変換される
        System.out.println(str);
    }
}
```


実行結果：
>java StringTest
10 years ago

いままで、
int x = 3;
System.out.println(x + "個");
と書けたのはこれが理由

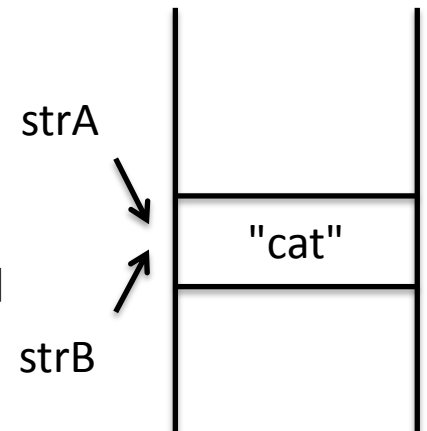
String型

- 文字列の内容の比較に
比較演算子は使えないので注意


```
String strA;  
String strB;  
...  
if (strA == strB)  
    System.out.println("二つの文字列は同じです");
```

 「strAとstrBは同じインスタンスか？」
チェックしているだけ

strA == strBがtrueになる状況



```
String strA;  
String strB;  
...  
if (strA.compareTo(strB) == 0)  
    System.out.println("二つの文字列は同じです");
```



strA.compareTo(strB) == 0
がtrueになる状況

