
Java

参照型の構造とメソッド作成

17 時間目

Javaの特徴である**オブジェクト指向**

を理解するには

Stringなどのデータ型を深く理解する必要がある

データ型

1 整数 : 1、2、3、0、-1、-2、-3 など

データ型名称	説明	大 ↑ ↓ 小
long	兆を超える数字。ビッグデータなどに使われる。	
int	21億くらいまでの数字。 (最も一般的に使われる。)	
short	3万くらいまでの数字。 (ほとんど使わない。)	
byte	8bitの符号付整数 (-128～127) を格納。	小

2 小数 : 1.1、0.5、3.14 など

データ型名称	説明	大 ↑ ↓ 小
double	通常はコレしか使わない。	
float	32bitの単精度浮動小数点数を格納。	

データ型

3 真偽値 : true、false

データ型名称	説明
boolean	通常はコレしか使わない。

4 文字 : a、あ、1 など

データ型名称	説明
char(キャラ)	1文字だけのもの（出力する際は、シングルクォテーションで囲む）

5 文字列 : abc、あいう、123など

データ型名称	説明
String	1文字以上の文字（＝文章）。（出力する際は、ダブルクォテーションで囲む） ※実は、文字列は、1文字ずつのchar型を別々に処理した後に、くっつけて文字列（文章）にしている。

データ型

プリミティブ型（＝基本データ型＝実体）

参照型（＝オブジェクト型）

1 整数

データ型名称
long
int
short
byte

2 小数

データ型名称
double
float

3 真偽値

データ型名称
boolean

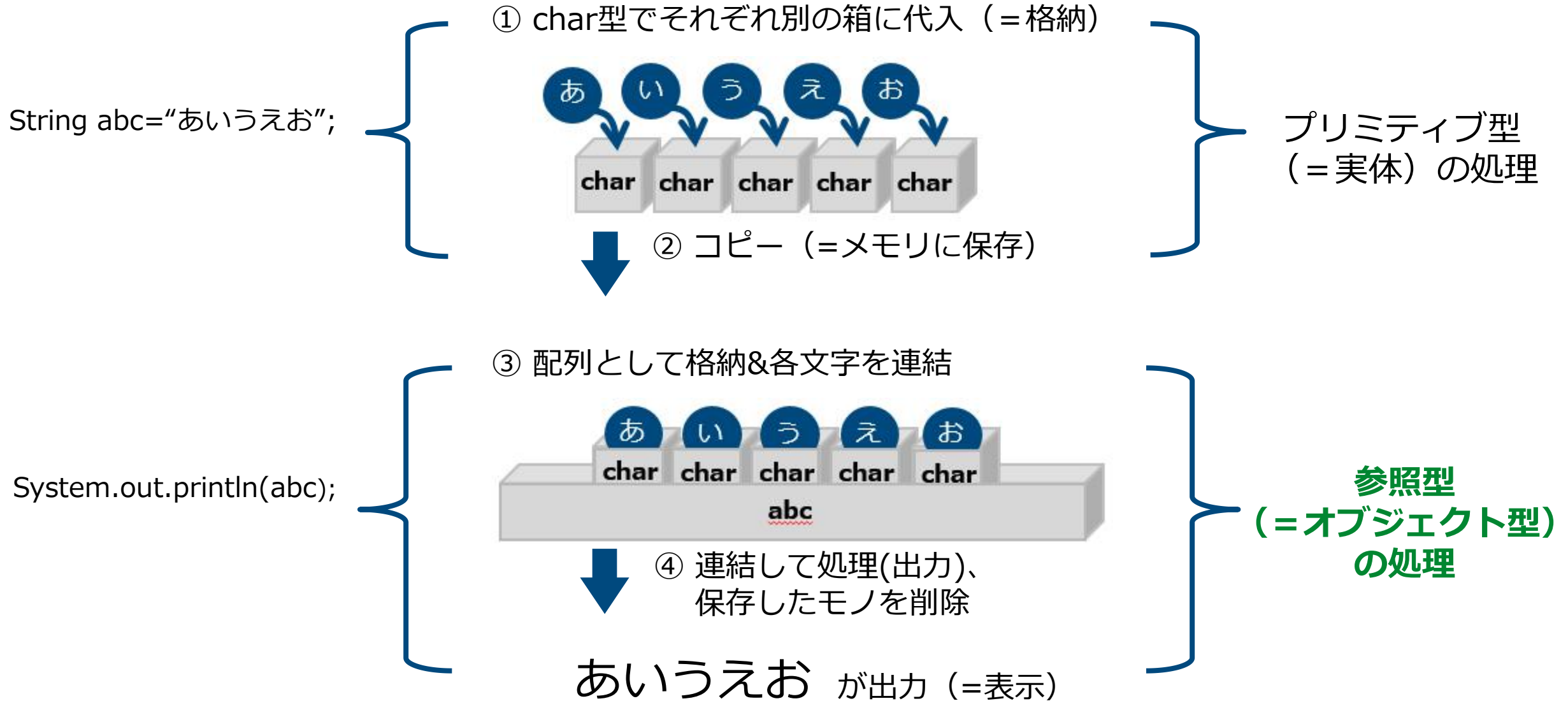
4 文字：

データ型名称
char(キャラ)

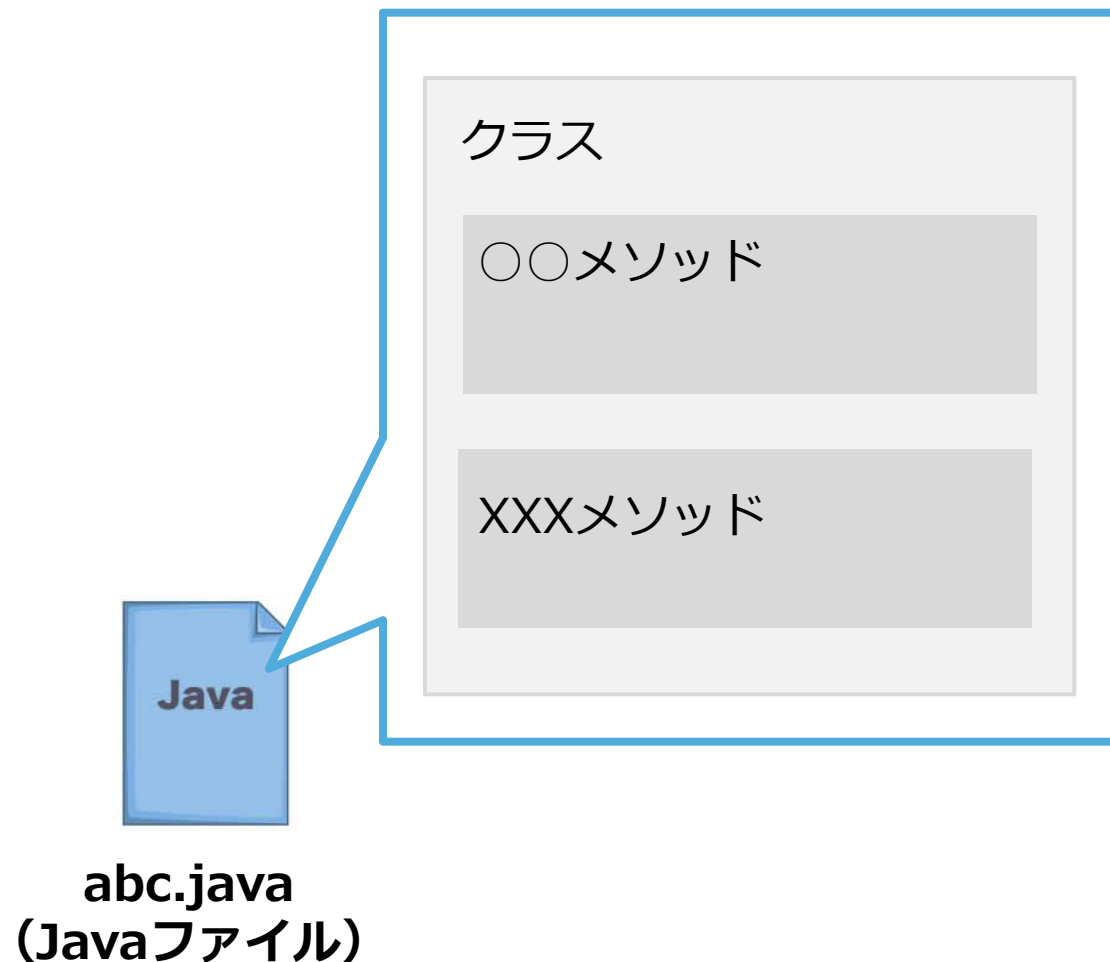
5 文字列

データ型名称
String

参照型(String)の中身の動き



メソッドとは



クラスの中には、
メソッドと呼ばれる処理を複数書く
ことができる

メソッドとは、
**JavaScriptやPHPで勉強した関数と
似た書き方をする**

メソッドの書き方

public static データ型 メソッド名 () {

return 処理結果;

}

決まり文句として、()を記述

任意（好きな）名前を記述

intやStringなどのデータ型を記述

処理結果を記述

決まり文句として、returnを記述

メソッドを記述

前のレッスンで作成した「Main.java」の中に 次のコードを記述しましょう。

The screenshot shows an IDE window with the file `Main.java` open. The code in the editor is:

```
1 package jp.co.internous.action;
2
3 public class Main {
4
5     public static void main(String[] args){
6         System.out.println("Hello World");
7     }
8
9
10 }
11
```

Two callout boxes provide instructions:

- ② `System.out.println("Hello World");` の下に
`System.out.println(gokei());`
と記述する。
- ① このスペースに、
`public static int gokei() {`
 `return 1+1;`
`}`
と記述する。

The console at the bottom shows the output: `<終了> Main (1) [Java アプリケーション] C:\pleiades\java\bin\javaw.exe (2018/01/0 Hello World`

メソッドを記述

Main.java

```
1 package jp.co.internous.action;
2
3 public class Main {
4
5     public static void main(String[] args){
6         System.out.println("Hello World");
7         System.out.println(gokei());
8     }
9     public static int gokei() {
10         return 1+1;
11     }
12
13 }
14
```

実行

① 『Main.java』を右クリック。

② 『実行』を選択

③ 『Javaアプリケーション』をクリック。

コンソール

```
<終了> Main (1) [Java アプリケーション] C:\pleiades\java\bin\java.exe
Hello World
```

```
package jp.co.internous.action;

public class Main {

    public static void main(String[] args){
        System.out.println("Hello World");
        System.out.println(gokei());
    }

    public static int gokei() {
        return 1+1;
    }
}
```

メソッドの実行結果

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H)

パッケージ・エクスプローラー

- sample
 - JRE システム・ライブラリー [JavaSE-1.8]
 - src
 - jp.co.internous.action
 - Main.java

アウトライン

- jp.co.internous.action
 - Main
 - main(String[]) : void
 - gokei() : int

Main.java

```
1 package jp.co.internous.action;
2
3 public class Main {
4
5     public static void main(String[] args){
6         System.out.println("Hello World");
7         System.out.println(gokei());
8     }
9     public static int gokei() {
10         return 1+1;
11     }
12 }
13
14
```

作成済みのメソッドはここに表示される

この『gokeiメソッド』を呼び出してる

gokeiメソッドの結果が表示された

2

別の方法でもメソッドを出力可能

The screenshot shows an IDE with the following components:

- メニューバー:** ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H)
- ツールバー:** Various icons for file operations, editing, and running.
- パッケージ・エクスプローラー:** Shows the project structure: sample > JRE システム・ライブラリー [JavaSE-1.8] > src > jp.co.internous.action > Main.java.
- アウトライン:** Shows the class structure: jp.co.internous.action > Main > main(String[]) : void, gokei() : int.
- Main.java:** Contains the following code:

```
1 package jp.co.internous.action;
2
3 public class Main {
4
5     public static void main(String[] args){
6         System.out.println("Hello World");
7         int total=gokei();
8         System.out.println(total);
9     }
10    public static int gokei() {
11        return 1+1;
12    }
13 }
```
- コンソール:** Shows the execution output: <終了> Main (1) [Java アプリケーション] C:\pleiades\java\8\bin\javaw.exe (2018/0: Hello World 2

A blue callout box points to the `int total=gokei();` line in the code, containing the following text:

下記のように一度、totalという変数に代入してもOK

```
int total =gokei();
System.out.println(total);
```

※変数とメソッドのデータ型は一致させること

いろいろなメソッド

いくつかメソッドの事例を見てみましょう。

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H)

パッケージ・エクスプローラー

- sample
 - JRE システム・ライブラリー [JavaSE-1.8]
 - src
 - jp.co.internous.action
 - Main.java

アウトライン

- jp.co.internous.action
 - Main
 - main(String[]) : void
 - gokei2(int, int) : int

Main.java

```
1 package jp.co.internous.action;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("Hello World");
7         System.out.println(gokei2(2,3));
8     }
9
10    public static int gokei2(int number1, int number2) {
11        return number1+number2;
12    }
13
14 }
```

コンソール

<終了> Main (1) [Java アプリケーション] C:\pleiades\java\8\bin\javaw.exe (2018/0...

Hello World
5

② System.out.println(gokei2(2,3));
で出力。

① int(整数) のデータ型でgokei2メソッドを作成。
引数に『int number1』と『int number2』を設定。
処理結果に『number1+number2』を設定。

③ 実行結果『5』が表示された。

The screenshot shows an IDE with the following components:

- Package Explorer:** Shows the project structure with 'sample' containing 'JRE システム・ライブラリー [JavaSE-1.8]', 'src' containing 'jp.co.internous.action' containing 'Main.java'.
- Outline:** Shows the 'Main' class with methods 'main(String[]) : void' and 'gokei2(int, int) : int'.
- Main.java:** Contains the following code:

```
1 package jp.co.internous.action;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("Hello World");
7         System.out.println(gokei2(5,7));
8     }
9
10    public static int gokei2(int number1, int number2) {
11        return number1+number2;
12    }
13
14 }
```
- Console:** Shows the execution output: '<終了> Main (1) [Java アプリケーション] C:\pleiades\java\bin\javaw.exe (2018/0/12) Hello World 12'.

② System.out.println(gokei2(5,7));
で出力。

① int(整数) のデータ型でgokei2メソッドを作成。
引数に『int number1』と『int number2』を設定。
処理結果に『number1+number2』を設定。

③ 実行結果『12』が表示された。

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H)

パッケージ・エクスプローラー

sample

JRE システム・ライブラリー [JavaSE-1.8]

src

jp.co.internous.action

Main.java

アウトライン

jp.co.internous.action

Main

main(String[]) : void

circle(int) : double

Main.java

```
1 package jp.co.internous.action;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("Hello World");
7         System.out.println(circle(5));
8     }
9
10    public static double circle(int hankei) {
11        return hankei*hankei*3.14;
12    }
13 }
```

② System.out.println(circle(5));
で出力。

① double(小数) のデータ型でcircleメソッドを作成。
引数に『int hankei』を設定。
処理結果に『hankei*hankei*3.14』を設定。

コンソール 呼び出し階層

<終了> Main (1) [Java アプリケーション] C:\nlelades¥java¥8¥bin¥javaw.exe (2018/0

Hello World
78.5

③ 実行結果『78.5』が表示された。

ファイル(F) 編集(E) ソース(S) リファクタリング(T) ナビゲート(N) 検索(A) プロジェクト(P) 実行(R) ウィンドウ(W) ヘルプ(H)

パッケージ・エクスプローラー

sample

JRE システム・ライブラリー [JavaSE]

src

jp.co.internous.action

```
② System.out.println(hikizan(10-10));  
System.out.println(kakezan(10-10));  
System.out.println(warizan(10-10));
```

で出力。

- ① intのデータ型でhikizan、kakezam、warizanメソッドを作成。
各メソッドの引数に『int number1, int number2』を設定。
hikizanの処理内容に『number1-number2』を設定。
kakezanの処理内容に『number1*number2』を設定。
warizanの処理内容に『number1/number2』を設定。

※引数名は、別メソッドであれば同じでも良い

Main.java

```
package jp.co.internous.action;
```

```
public class Main {
```

```
    public static void main(String[] args){  
        System.out.println("Hello World");  
        System.out.println(hikizan(10,10));  
        System.out.println(kakezan(10,10));  
        System.out.println(warizan(10,10));  
    }
```

```
    public static int hikizan(int number1,int number2){  
        return number1-number2;  
    }
```

```
    public static int kakezan(int number1,int number2){  
        return number1*number2;  
    }
```

```
    public static int warizan(int number1,int number2){  
        return number1/number2;  
    }
```

- ③ hikizanの実行結果『0』
kakezanの実行結果『100』
warizanの実行結果『1』が表示された。

<終了> Main (1) [Java アプリケーション] C:\pleiades\java\8\bin\javaw.exe (2018/0

Hello World

0
100
1