

第18章 staticメンバ

目次

- インスタンス変数と インスタンスメソッド
- static変数とstaticメソッド
- staticメソッドの注意点

インスタンス変数とインスタンスメソッド

インスタンス変数: オブジェクトのフィールド。

インスタンスメソッド: オブジェクトのメソッド。

ローカル変数:

メソッド内で宣言した変数やメソッドの仮引数。

宣言したメソッド内でのみアクセス可能。

オブジェクトを初期化したい場合は、コンストラクタを利用します。

インスタンス変数とインスタンスメソッド

```
public class Student {  
    private int no;  
    private String name;  
    private String gender;  
    private int age;  
  
    public Student(int no, String name, String gender, int age) {  
        this.no = no;  
        this.name = name;  
        this.gender = gender;  
        this.age = age;  
        System.out.println("学生を作成しました。");  
    }  
}
```

インスタンス変数

ローカル変数

インスタンス変数とインスタンスメソッド

```
public void show() {  
    System.out.println("学生番号は" + no + "番です。");  
    System.out.println("名前は" + name + "です。");  
    System.out.println("性別は" + gender + "です。");  
    System.out.println("年齢は" + age + "歳です。");  
}  
}
```

インスタンス
メソッド

インスタンス変数とインスタンスメソッド

オブジェクトを生成することで、
インスタンス変数とインスタンスメソッドを利用できる。

```
public class Sample {  
    public static void main(String[] args) {  
        Student student1 = new Student(1, "まなみ", "女", 19);  
        student1.show();  
        Student student2 = new Student(2, "けんた", "男", 22);  
        student2.show();  
    }  
}
```

インスタンス変数とインスタンスメソッド

オブジェクトを生成することで、それぞれのオブジェクトのフィールドの値の代入や取得ができる。

学生番号、名前、性別、年齢の情報(インスタンス変数):

→オブジェクトごとに存在する。

オブジェクトごとに各フィールド が関連付けられている状態。

show()メソッド(インスタンスメソッド):

オブジェクトに関連付けられている。

インスタンス変数とインスタンスメソッドは
オブジェクトに関連付けられて存在している。

インスタンス変数とインスタンスメソッド



オブジェクトに関連づけられているフィールドをインスタンス変数と呼びます。
オブジェクトに関連づけられているメソッドをインスタンスメソッドと呼びます。

static変数とstaticメソッド

オブジェクトに関連づかず、クラス全体に関連付けられたフィールド、メソッドをstaticメンバという。

フィールド: static変数(クラス変数)

メソッド: staticメソッド(クラスメソッド)

```
static 型名 変数名;
```

```
static 戻り値の型 メソッド名(引数リスト){  
    処理;  
}
```

static変数とstaticメソッド

staticメンバの呼び出しは、以下のように記述する。

クラス名.staticメンバ名(メソッド名or変数名)

【Sample1801 staticメンバ】を作成しましょう



Sample1801のポイント

Student1801クラスでは、
static変数sumを宣言して、0で初期化している。
このstatic変数は、学生の人数をあらわす
フィールドとして使う。

```
public static int sum = 0;
```

Sample1801のポイント

オブジェクトが1つ生成されるたびに、
コンストラクタ内でstatic変数sumが1加算される。
この処理により学生の人数が増える。

```
sum++;
```

Sample1801のポイント

static変数の値はstaticメソッドから呼び出せる。

サンプルコードでは、showSum()メソッド内でstatic変数sumを呼び出している。

```
public static void showSum() {  
    System.out.println("学生は合計" + sum + "人です。");  
}
```

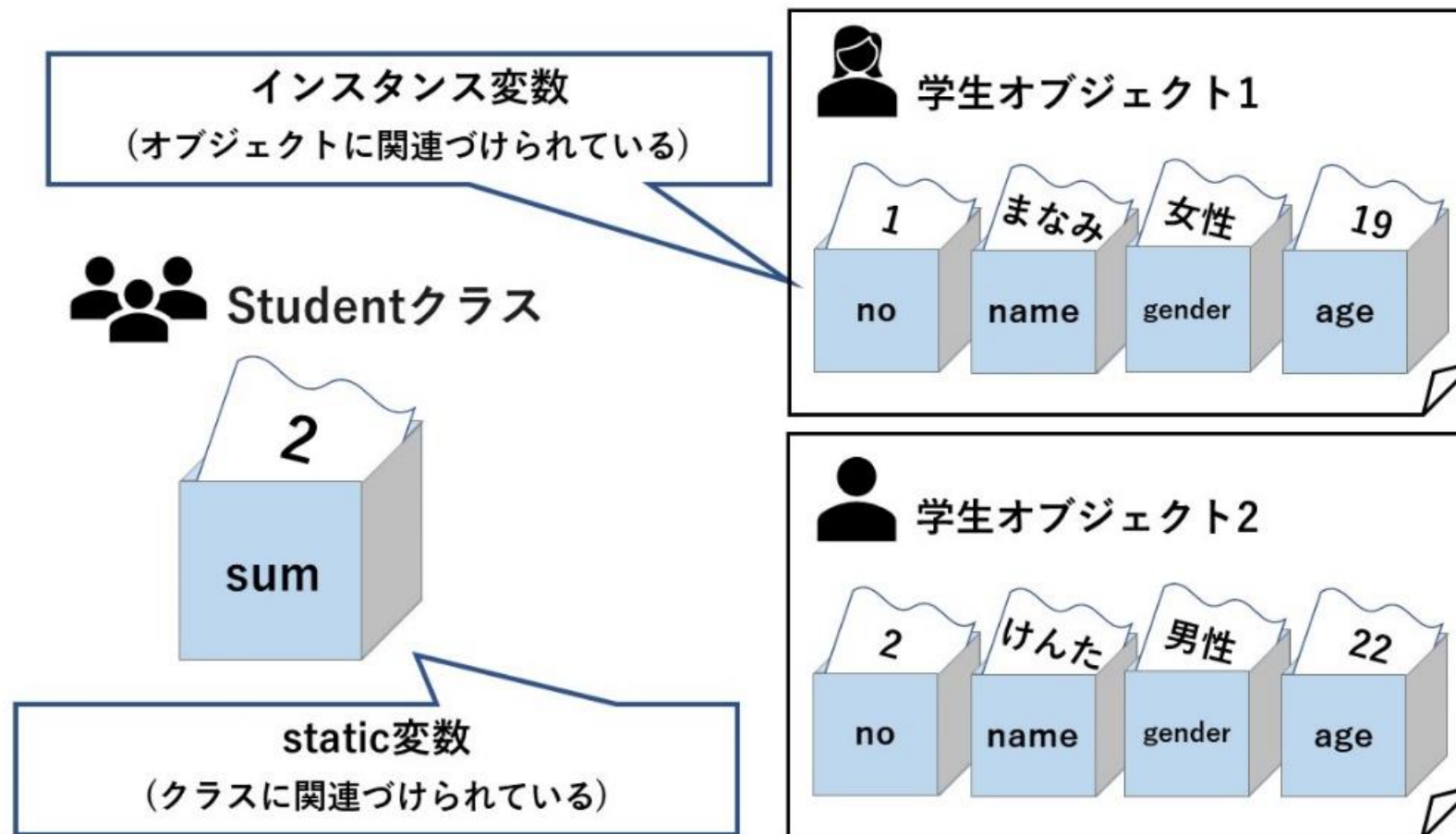
Sample1801のポイント

Sample1801クラスのmain()メソッドでは
showSum()メソッドを3回呼び出している。

staticメソッドは、オブジェクトを生成しなくても、
クラスから直接呼び出せる。

```
Student1801.showSum();
```

static変数とインスタンス変数の違い



static変数とインスタンス変数の違い

インスタンス変数、インスタンスメソッドの利用:

各オブジェクトを独立した部品として機能させられる

staticメンバの利用:

- オブジェクト全体に関わる情報を管理したい場合
- オブジェクト間でデータや機能を共有したい場合
- インスタンス変数やインスタンスメソッドを必要としない処理を実行したい場合 (parseInt()メソッドなど)

クラスに関連付けられているフィールドをstatic変数と呼びます。
クラスに関連付けられているメソッドをstaticメソッドと呼びます。

staticメソッドの注意点

「メンバがそのオブジェクトに属する」を表すthisは、staticメンバには付けられない。

```
private static int sum = 0;  
public static void showSum() {  
    System.out.println("学生は合計" + this.sum + "人です。  
    ");  
}
```

コンパイルエラーとなる

staticメソッドの注意点

staticメソッド内では、インスタンス変数とインスタンスメソッドにアクセスできない。
(staticメソッドは特定のオブジェクトに関連付けられたものではないため)

```
private String name;
public static void showSum(){
    System.out.println("生徒名" + name + "です。");
}
```

インスタンス変数

コンパイルエラーとなる

章のまとめ

- オブジェクトに関連付けられているフィールドをインスタンス変数と呼びます。
- オブジェクトに関連付けられているメソッドをインスタンスメソッドと呼びます。
- クラスに関連付けられているフィールドをstatic変数と呼びます。
- クラスに関連付けられるメソッドをstaticメソッドと呼びます。
- 「this.」はインスタンス変数とインスタンスメソッドにのみ付けられます。
- staticメソッドからはインスタンス変数、インスタンスメソッドにアクセスできません。