

Javaプログラミング実習

26. ジェネリクス

株式会社ジーードライブ

今回学ぶこと

- 汎用性をもつクラスの作成
 - ジェネリクスについて

汎用性をもつクラス

- 以下のような「会員の情報を保持するためのクラス」を例に考えます

```
public class Member {  
    private String name;          // 氏名  
    private int age;              // 年齢  
    private ???型 extraInfo1;    // 追加情報1  
    private ???型 extraInfo2;    // 追加情報2  
    ...  
}
```

型については、利用する際に決定したい

- 追加情報については、汎用的に(柔軟に)情報を格納できるようにしたいと考えています
 - 追加情報については、会員の生年月日、職業、住所など、利用する際に決定できるようにしたい

ジェネリクス

- これを可能にするのがジェネリクスという仕組み
 - ジェネリクスを使用すると、利用時に型を決定することができる
- ジェネリクスでは、`< >` ダイヤモンド演算子を使用し、型の識別子を定義する
 - 識別子は T や U といった 1 字の大文字が使用されることが一般的

例：ジェネリクスを使用したクラスの定義

```
public class Member<T, U> {  
    private String name;           // 氏名  
    private int age;               // 年齢  
    private T extraInfo1;          // 追加情報 1  
    private U extraInfo2;          // 追加情報 2  
    ...  
}
```

ジェネリクス

- 利用時に型を決定することができる
 - プリミティブ型は使用不可なので、intやdoubleなどを使用したい場合は、ラッパークラスの型を指定する

例：ジェネリクスを使用したクラスの利用例

```
// 付加情報1：勤務先(文字列)  
// 付加情報2：給与(整数)  
Member<String, Integer> member =  
    new Member<String, Integer>();
```

前頁の **T** が String になる

前頁の **U** が Integer になる

省略可能

練習

- 練習26-1