Interfaceの利用

オブジェクト指向プログラミング特論

2020年度

只木進一：理工学研究科

クラスと階層構造

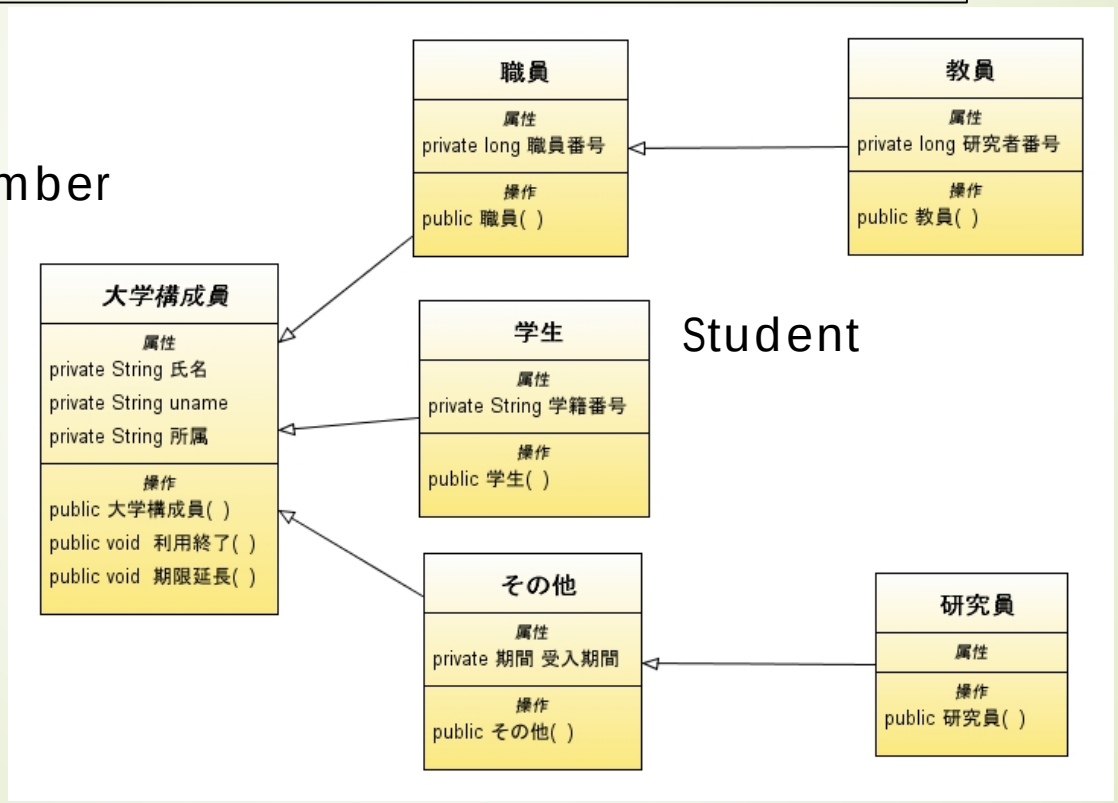
- Class
  - 対象の類型を表す
- 日常における類型化
  - 階層構造
  - 上位：一般化
  - 下位：具体化
- 適切な階層化が概念整理に有効

Class, Super Class, Subclass

- クラスには階層構造がある
  - 例：生き物の階層構造
  - 例：組織の階層構造
- 上位のクラス：super class
  - 抽象化、汎化 (Generalization)
- 下位のクラス：subclass
  - 具体化 (Specialization)

クラス階層の例

```
Member a = new Staff(nameA,unameA,divisionA);
Member b = new Student(nambeB,unameB,division);
```



継承 (Inheritance)

- Subclassの定義
- 全てのフィールドとメソッドが継承される
- 具体化 (Specialization)
  - フィールド・メソッドの追加
  - 実装の追加・変更

汎化 (Generalization)

- 共通なものとしての取扱い
  - Superclass
- 共通的フィールド・メソッドの抽出
  - 実装なしの場合も
  - 名前だけの定義

Method Overload

- メソッドの二つの要素
  - contact またはsignature
    - メソッド名
    - メソッドの引数並び
- 実装
  - メソッドを多重に定義できる

Polymorphism

- 継承したクラスでメソッドを上書き

```
Member a = new Staff(nameA,unameA,divisionA);
Member b = new Student(nambeB,unameB,division);
```

  - クラスを利用する側から見ると、親クラスのインスタンスとしても利用できる。
  - a とbのインスタンスでは、同じメソッドでも動作が異なる可能性

Javaでの継承の制約

- 多重継承の困難さ
  - 複数の親クラスのどの性質を引き継ぐか
- Javaでは
  - 一つのクラスしかextendsで継承できない
  - 特殊なクラスinterface
  - 複数のinterfaceを継承できる
- Abstract classとInterfaceの使い分け

interface

- フィールドの制限
  - static finalのみ持つことができる
  - これらキーワードは省略化
- メソッドの制限
  - abstractメソッド：実装なし
  - defaultメソッド：実装あり

- 継承クラスで、interfaceをimplementsする
  - メソッドを必ず実装する
  - defaultメソッドをそのまま使う場合にはsuperで明示する。
- 操作する側の視点での抽象化
  - 特定のメソッドを有しているため、同じ方法で操作できる
  - 実装は異なる可能性に注意

Comparableインターフェース

- APIドキュメントを見よう
  - https://docs.oracle.com/javase/jp/11/docs/api/java.base/java/lang/Comparable.html
- java.lang.Comparable
  - 定義されているメソッド
  - 既知の実装クラス

今日のtasks

- example1パッケージで作業
- StudentクラスにComparableインターフェイスを追加
  - compareTo()メソッド実装
- MergeSortクラスをComparableの拡張クラスへ対応

本日のタスク

StudentクラスにComparableインターフェイスを追加

- example0/Student.javaをコピー
  - package名に注意
- クラス定義

public class Student implements Comparable<Student>

- メソッド実装

public int compareTo(Student o)

本日のタスク

MergeSortをComparableであるクラスに対応させる

- example1/MergeSort.javaをコピー
- クラステンプレートの利用
  - 特定のクラスを指定しない
  - どのクラスの拡張であるかを指定し、使えるメソッドを特定

public class **MergeSort<T extends Comparable<T>>**

本日のタスク

MergeSortクラスをクラステンプレートに対応

- Studentクラスを全てTに変更
- 要素の大小関係を調べている部分の特定
  - compareTo()メソッドに置き換え
- 正しく動作することを確認

宿題

- Comparableの派生クラスを対象としたBubbleソートを作成
  - template配布済み
- MergeSortクラスと共通の部分
  - コンストラクタでデータを与える
  - sort()メソッドで実行
  - less()メソッドは共通
- swap()メソッドを作る