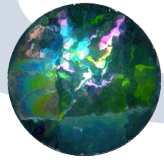


Java入門：その2



Javaのプログラムの基本構造

package パッケージ名;

import インポートするパッケージ名またはクラス名;

public class クラス名 extends 親クラス名 implements インターフェイス名{
 クラス内のフィールド(定数、変数);

```
public クラス名(引数並び){//コンストラクタ  
}
```

```
public 型 メソッド(引数並び){
```

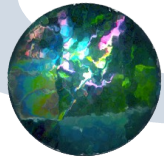
```
    return 戻り値;
```

```
}
```

```
private 型 メソッド(引数並び){
```

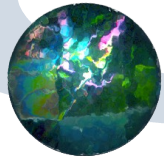
```
    return 戻り値;
```

```
}
```



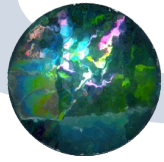
C++と異なる点

- 一つのファイルに一つのクラス
 - ファイル名はクラス名
- デストラクタが無い
 - 使われていないインスタンスは自動的に消去
- 二種類の継承
 - クラスとインターフェイス



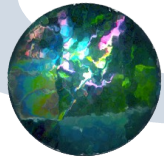
プログラム開発の要点

- アルゴリズムをデータの詳細と切り離す
 - ソートのアルゴリズムは、ソートされるデータの詳細とは関係ない
 - スレッドプログラムは、各スレッド内で何をしているかとは関係ない
- 問題をオブジェクトの運動として捉える
 - 小さなオブジェクトへ分割



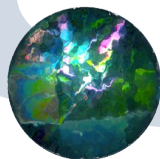
StudentSampleを改善する

- StudentSampleをもとに、ソートプログラムを
- 新しいパッケージStudentSample2を作る
- 元のファイルをコピー
- クラスStudentにComparableインターフェイスを
 - 中の構造によらずに比較する
- クラスStudentSampleにsortメソッドを
 - データの種類によらずにソートできるように
- Student.java
- StudentRecord.java



Comparable インターフェイス

- 必ずメソッド `int compareTo()` を実装しなければならない。
 - 引数と比較して、0 または ± 1 を返す
- ソートプログラムは、Comparable インターフェイスを持ったクラスを `compareTo()` を使って並べ替える。
 - クラスの中の構造は知らなくて良い



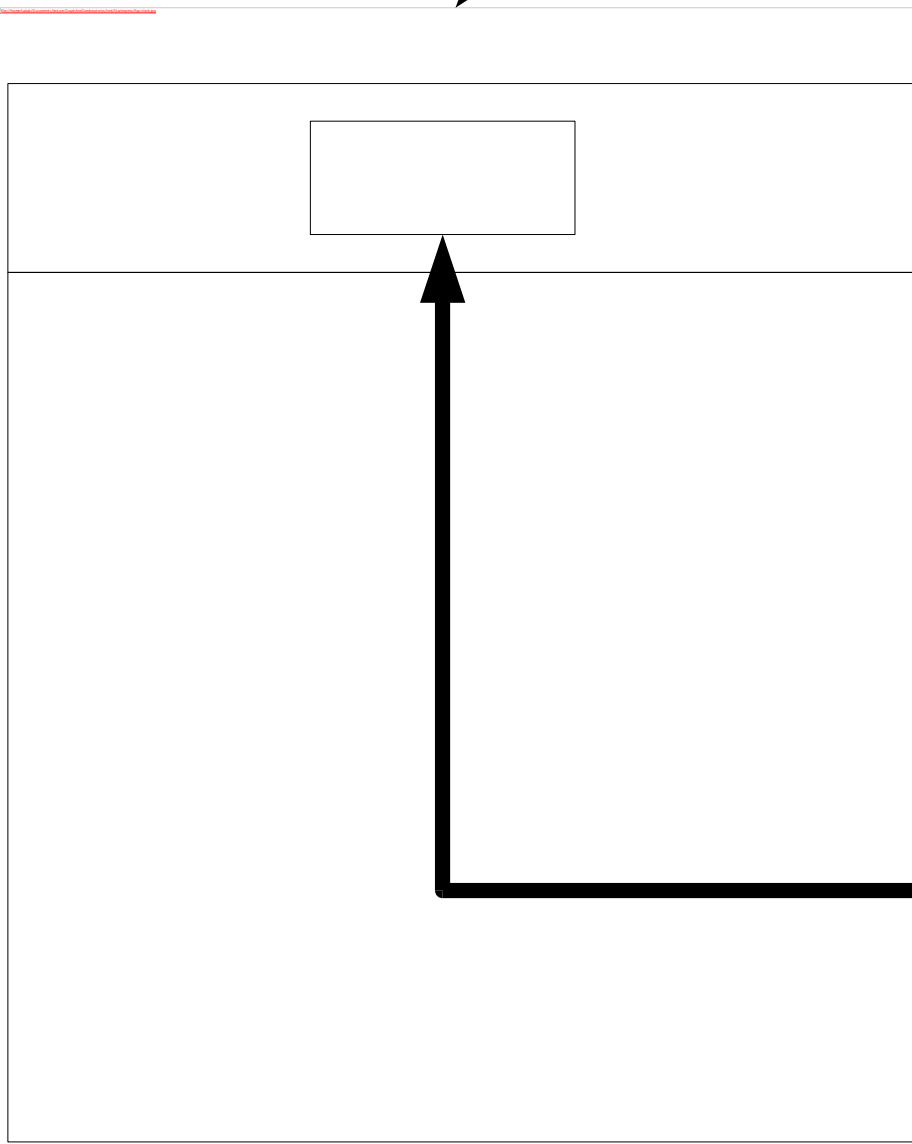
GUIのある例

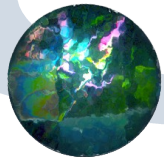
JFrame : 一番外側

JPanel : ボタンを置く場所

JPanel : 時刻の文字列を書く場所

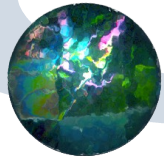
JButton : ボタン





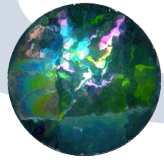
GUIアプリケーション構築手順

- 中心となるパーツのクラスを定義する
 - 今回は時刻を表示するJPanel
 - 詳細は後で作る
- 外側のJFrameを作る
 - ボタンを置くJPanelを置く
 - ボタンを作る
 - 中心となるパーツを置く



時刻を表示するパネル

- Runnableインターフェイス
 - スレッドからrun()メソッドを通じて呼び出し
- paint()メソッドで時刻の文字列を描画
- **ClockPanel.java**



メインのJFrameを作る

- レイアウトマネージャの設定
 - 通常はBorderLayoutを使用
- ボタン用のJPanelを置く
 - レイアウトマネージャー：FlowLayout
- メインの要素を配置
- ボタンを配置
- ボタンの動作を記述する