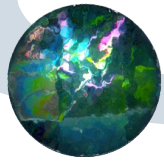
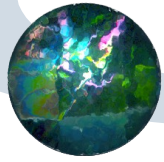


# まとめ



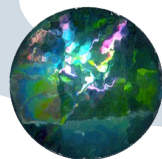
# プログラミング上達のために

- モジュール化
  - 全体の構成を整理する
  - データ・モデル、全体の流れ、UIを分離する
- 適切なライブラリの活用
  - 必ず理解すること
- 良い例題の学習
  - 流れと、細かな実装を読む
- 良い教科書を持つ



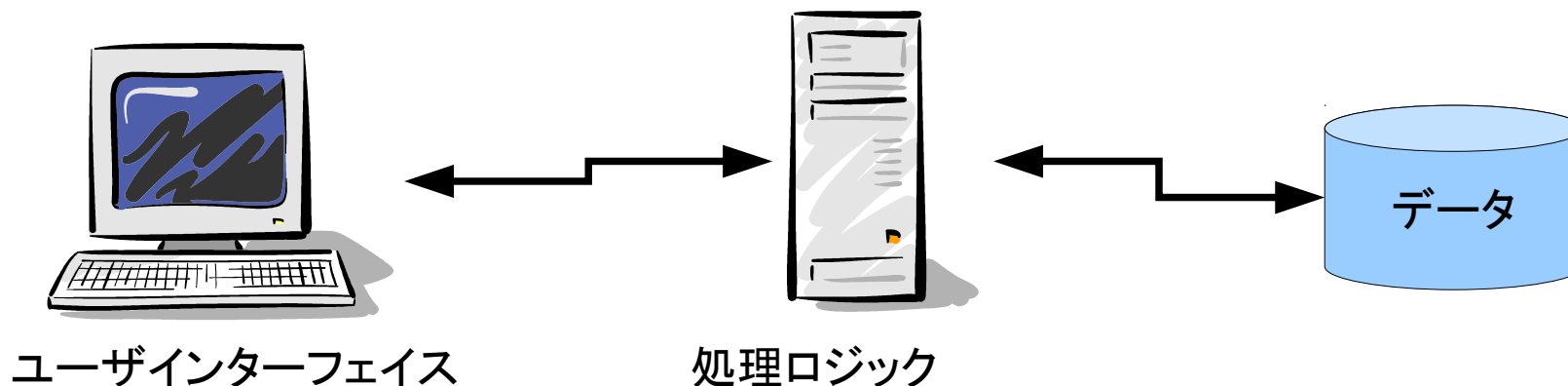
# オブジェクト指向プログラミング

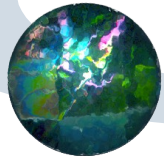
- プログラミング対象を、実際の操作対象のように捉える
  - シミュレーション：物体
  - データベース：レコード
- ユースケース的解析
  - 処理を短い文に分解する
  - 名詞に注目し、クラスを定義する



# Programming framework : Object Oriented

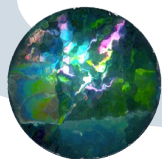
- オブジェクトとして対象を捉える
  - データとその動作・処理の組
- オブジェクトの相互作用としてシステムの動作をとらえる。
  - 日常の考えに近い



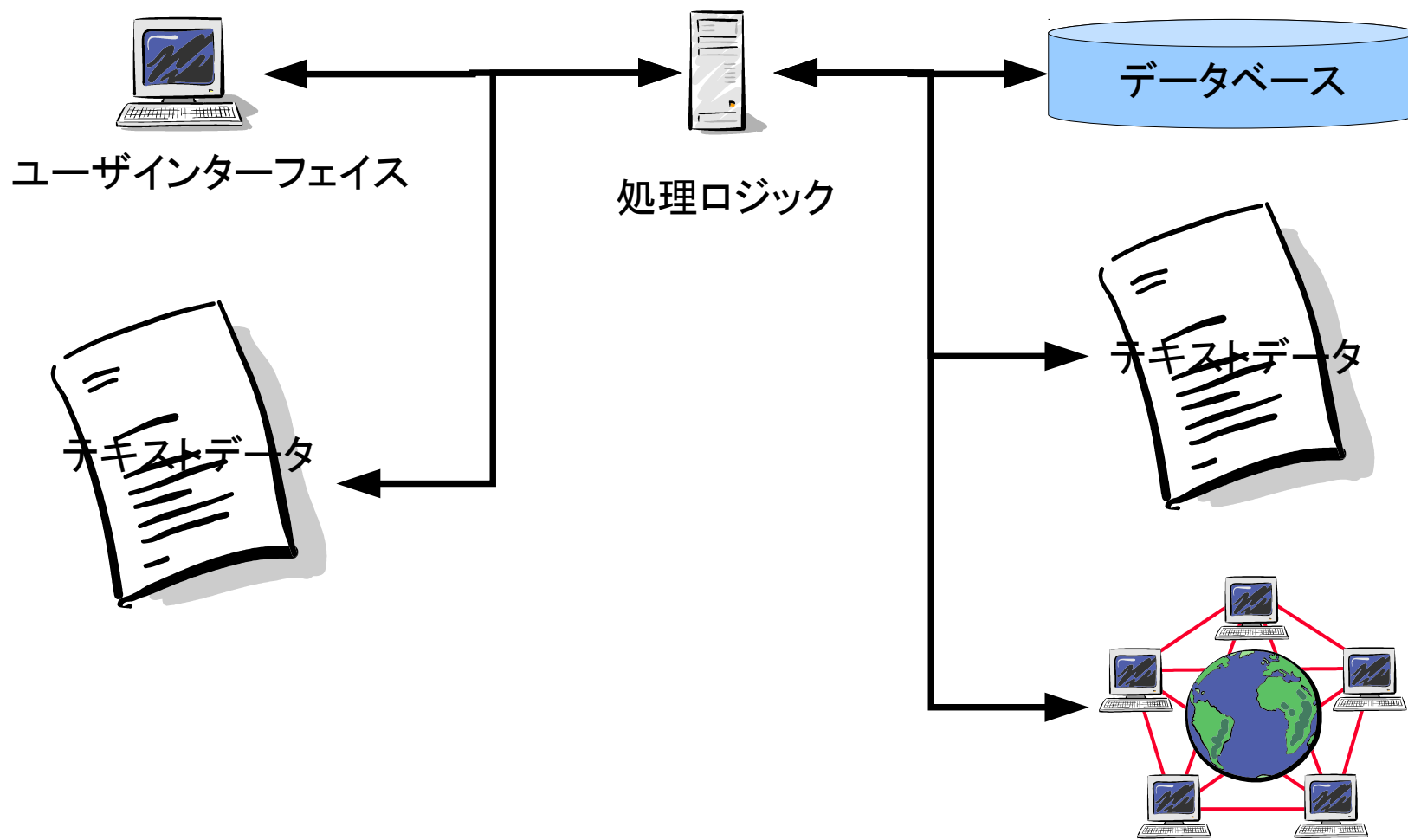


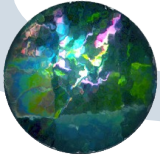
# Objects, Classes and Instances

- オブジェクト
  - 取り扱い対象
- クラス
  - オブジェクトを抽象化したもの
  - データの型に相当
- インスタンス
  - クラスを実体化したもの
  - 具体的な値が入っているデータ



# 実装とインターフェースの分離

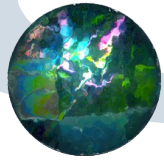




```
Public class Class {  
    private int field=0;  
    public getField(){return field;}  
    public setField(int field){  
        this.field=field;  
    }  
}
```

取得メソッド  
getter

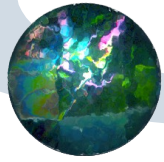
設定メソッド  
setter



# クラスの継承 (Inheritance)

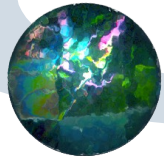
- 既存のクラスを引き継いで新しいクラスを定義
  - より具体化する
  - 対象に近づける
- 元のクラスのFieldとMethodを引き継ぐ
- 新しいFieldやMethodを追加する
- Methodを実装する：抽象クラスの継承





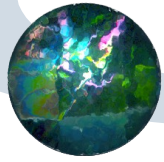
# Abstract(抽象)クラス

- 継承する元を定義する
- 共通的動作・fieldを定義する
- 一つ以上のmethodが実装されていない
  - abstract メソッド
  - 継承クラスで必ず実装



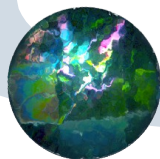
# interface

- 特殊なクラス
- 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`メソッドのみ：実装なし
- 継承クラスで、`interface`をimplementsする
  - メソッドを必ず実装する



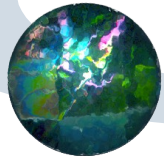
# クラス継承の例：通常のクラス

- Swingを使ったGUI構築
  - 一番外側のWindow: javax.swing.JFrameの拡張
  - 図などを表示するパネル: javax.swing.JPanelの拡張



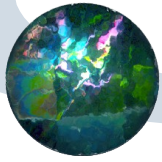
# クラスの継承の例：抽象クラス

- 表の作成
  - `javax.swing.table.AbstractTableModel`の拡張
- スレッドの利用
  - `java.lang.Runnable`のインプリメント
- 比較可能なオブジェクト
  - `java.lang.Comparable`のインプリメント
- クラステンプレートの活用



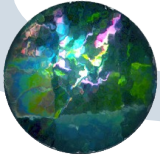
## 役立つ事例

- ファイル入出力
- 標準入出力のラップ
- ファイル選択ダイアログ



## 参考書

- Patrick Niemeyer, Jonathan Knudsen, *Learning Java 3<sup>rd</sup> ed.* (Oreilly, 2005).
- M. Loy, R. Eckstein, D. Wood, J. Elliott and B. Cole, *Java Swing 2<sup>nd</sup> ed.* (Oreilly, 2002).
- G. Reese, *JDBC and Java 2<sup>nd</sup> ed.*, (Oreilly, 2000).
- B. McLaughlin, *Java & XML 2<sup>nd</sup> ed.* (Oreilly, 2001).
- I. F. Darwin, *Java Cookbook* (Oreilly, 2001).



- D. Poo, D. Kiong and S. Ashok, *Object-Oriented Programming and java* (Springer, 2008).
- E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns* (Addison-Wesley, 1995).