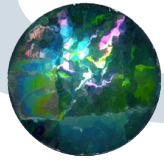
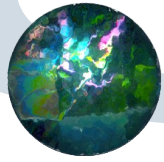


オブジェクト指向の考え方



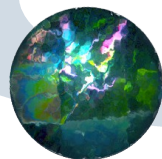
Programming framework : Procedure Oriented

- データ処理手順を中心に考える
 - データ処理のためのサブプログラムや関数を組み合わせる
- 流れ図による整理
- 手続き型プログラミング言語
 - C、Fortran
- 対象の操作・動作を手続きに翻訳しなければならない。
- アルゴリズム最適化などで高速化可能。



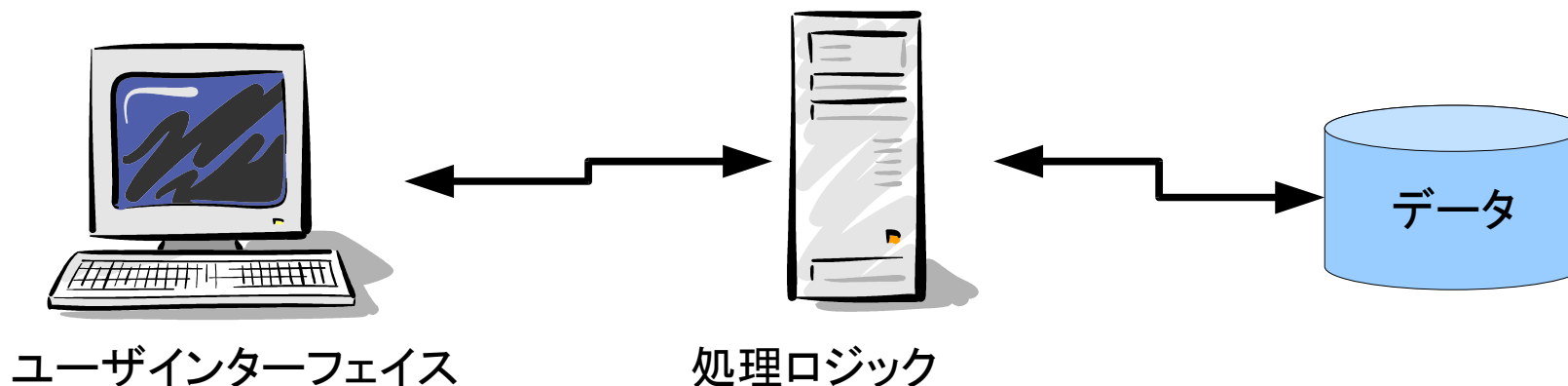
Programming framework : Data Oriented

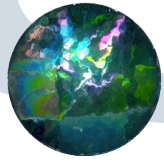
- データを中心に考える
 - データがどのように変化していくか
- データフローによる整理
- データベース処理
- 対象をデータとして整理しなければならない。
- 巨大なデータを扱える。



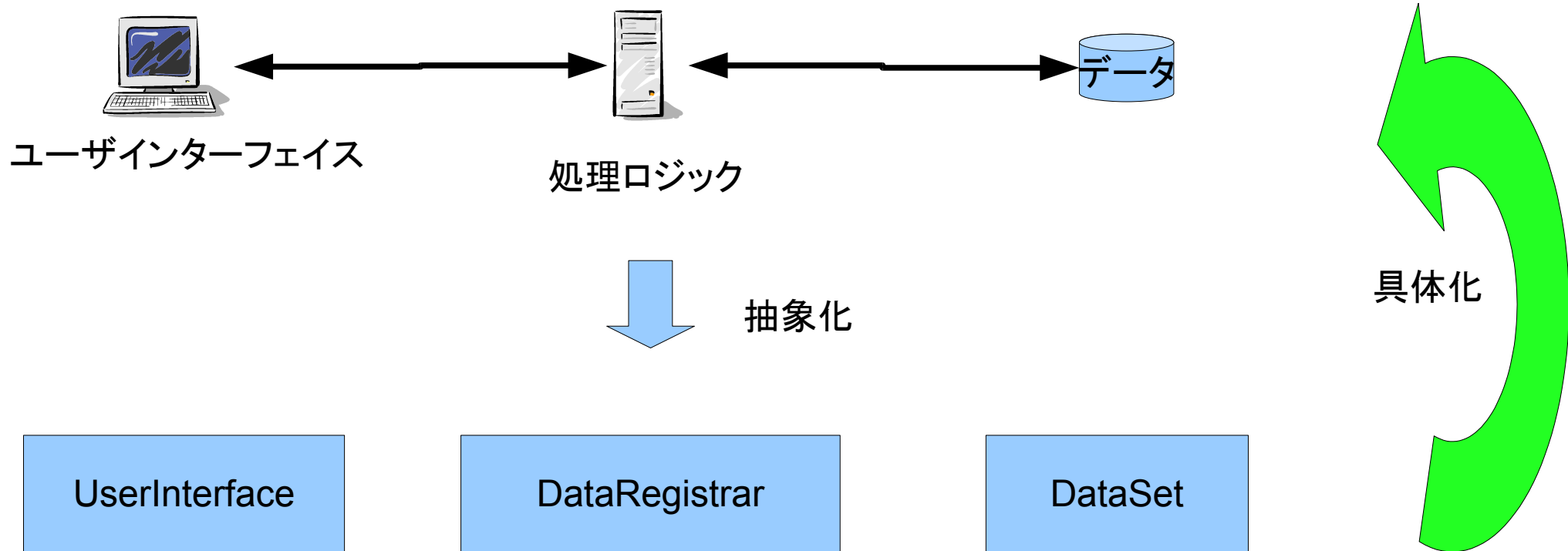
Programming framework : Object Oriented

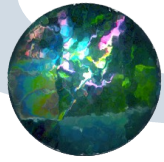
- オブジェクトとして対象を捉える
 - データとその動作・処理の組
- オブジェクトの相互作用としてシステムの動作をとらえる。
 - 日常の考えに近い





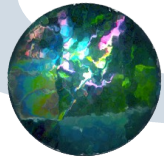
Objects, Classes and Instances



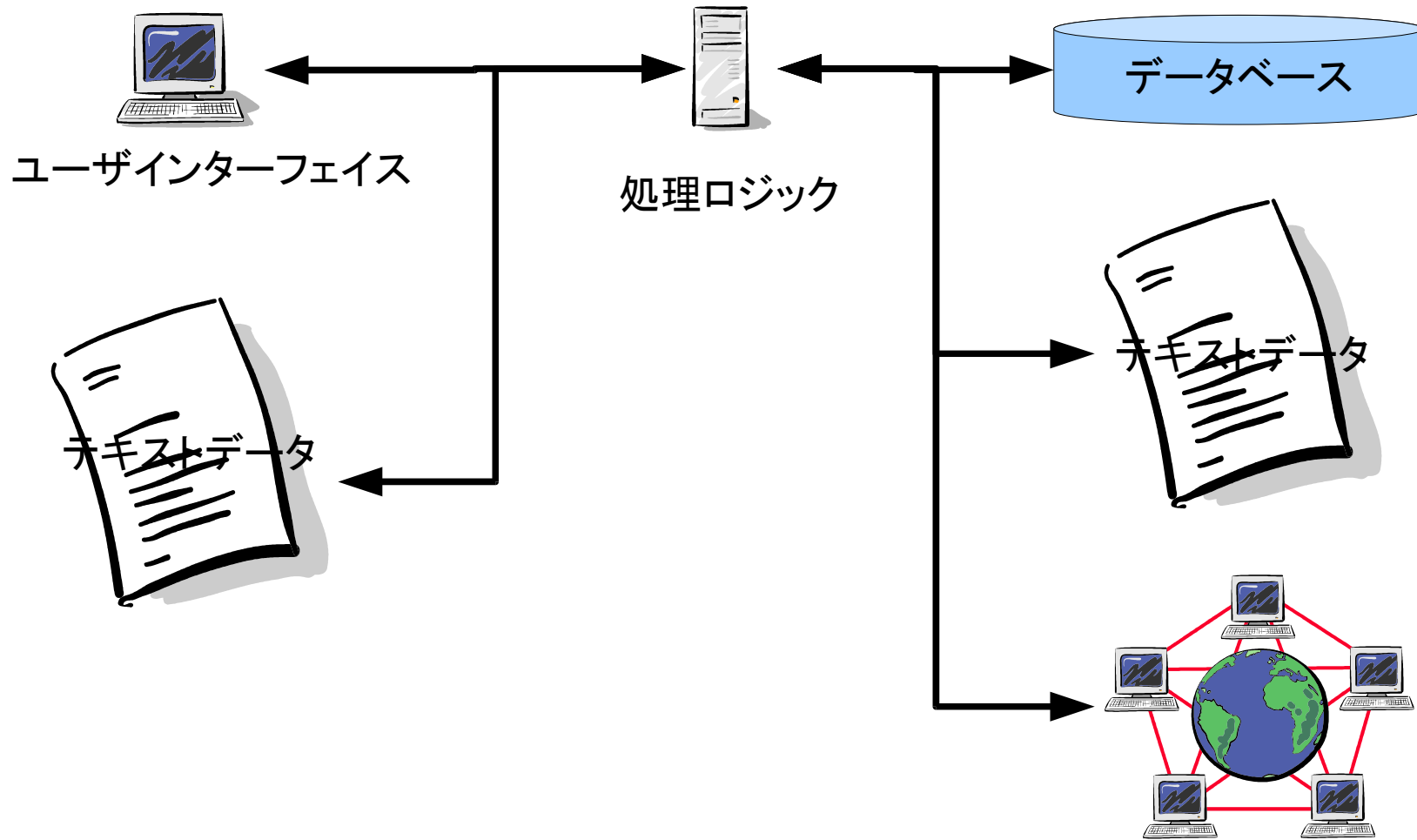


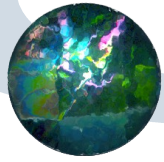
Objects, Classes and Instances

- 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



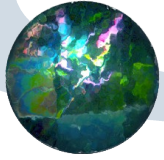
実装とインターフェースの分離





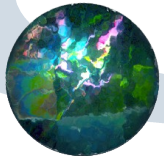
オブジェクト指向で得られるもの

- 様々な実装に対応したい
- できるだけプログラムをいじらない
- 他人の書いたプログラムを活用する
- 開発コスト・保守コストの削減



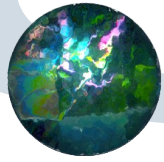
Field

- クラス内のデータ
- Fieldと呼ぶ
- FieldへのアクセスはMethodを通じて行う
 - アクセス制限
- クラスに属するfieldとインスタンスに属するfield



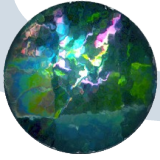
Method

- fieldへアクセスするmethod
 - クラスのデータへアクセス:Getter
 - クラスのデータを変更:Setter
- オブジェクトの操作を行うmethod

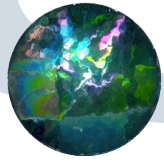


Javaでは全てがクラスに属する

- 主語を中心に問題を記述する。
- どうしても述語になってしまうものもある。
 - 数学関数など汎用的な処理類
- その述語を管理すべきクラスを考える
 - クラスに属するmethodとして定義

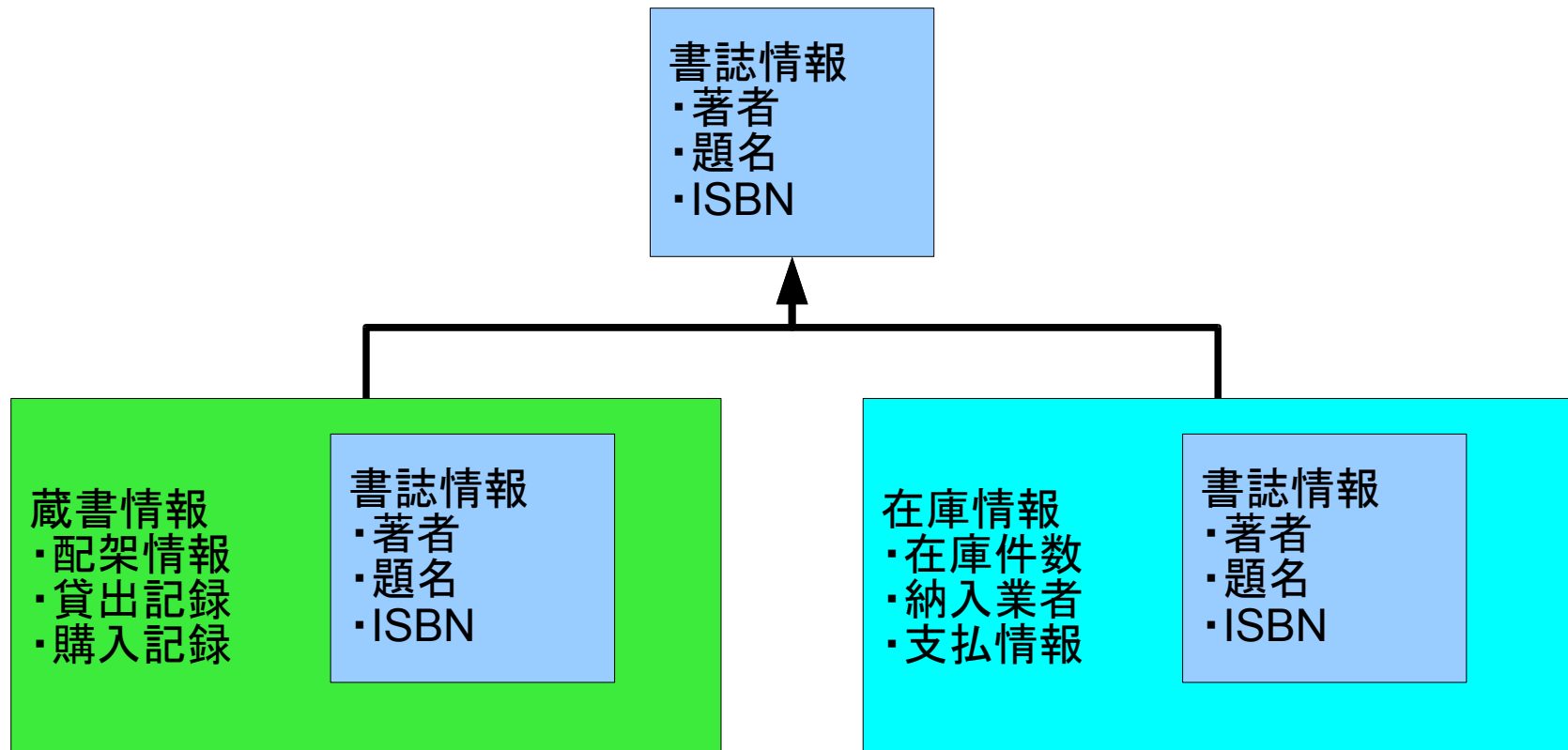
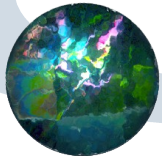


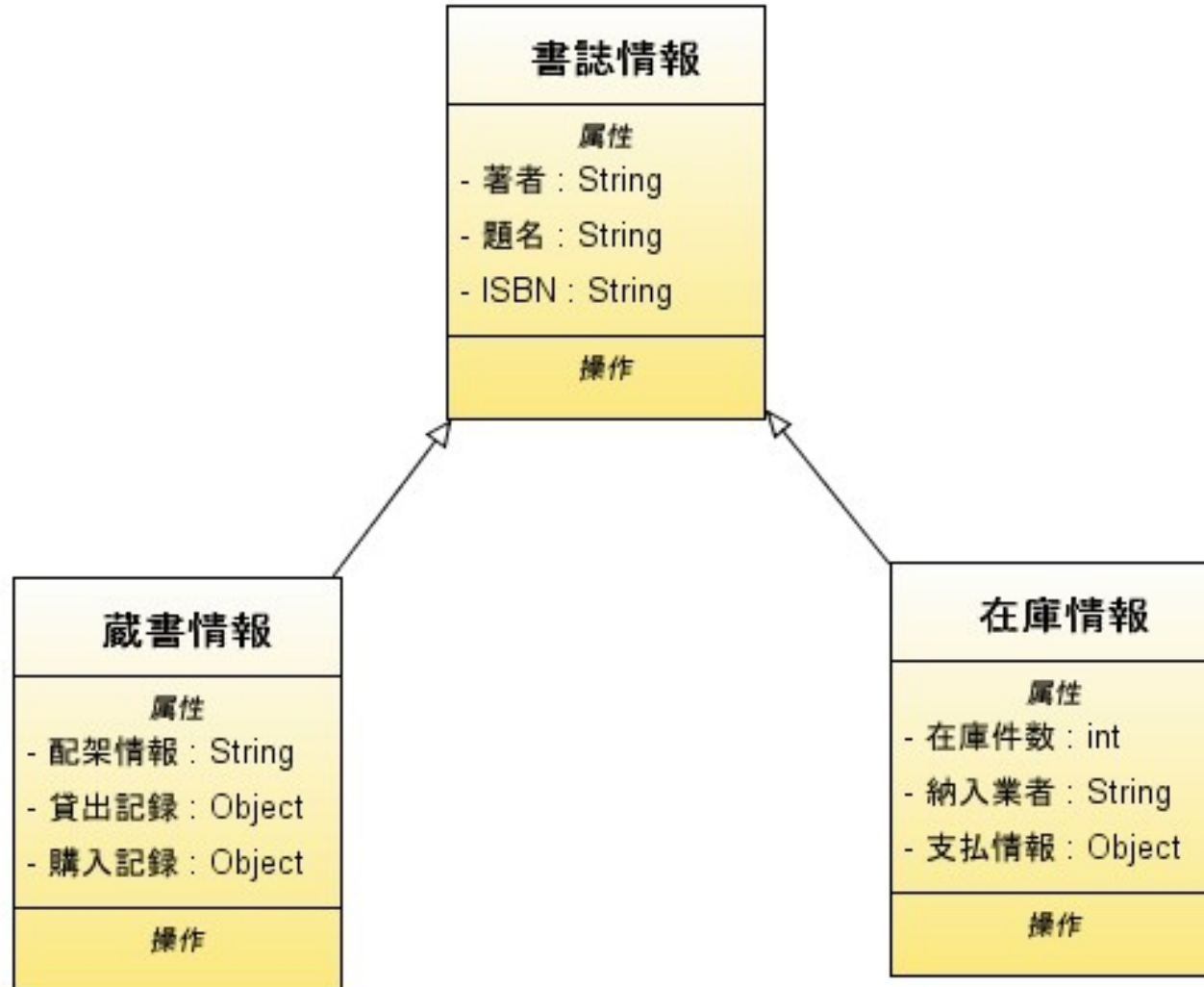
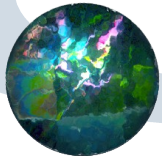
```
Public class Class {  
    private int field=0;  
    public getField() {return field;} ← Getter  
    public setField(int field) { ← Setter  
        this.field=field;  
    }  
}
```

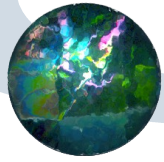


クラスの継承 (Inheritance)

- 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

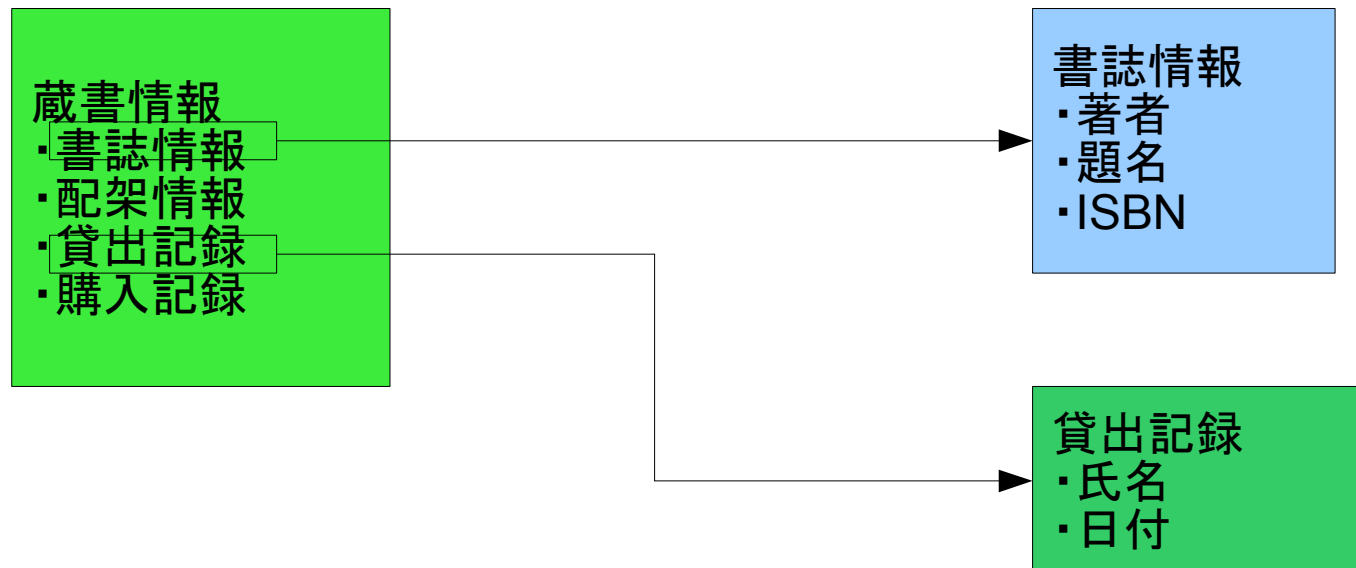


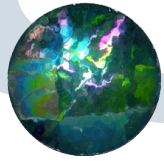




集約 (Aggregation) と 合成集約 (Composition)

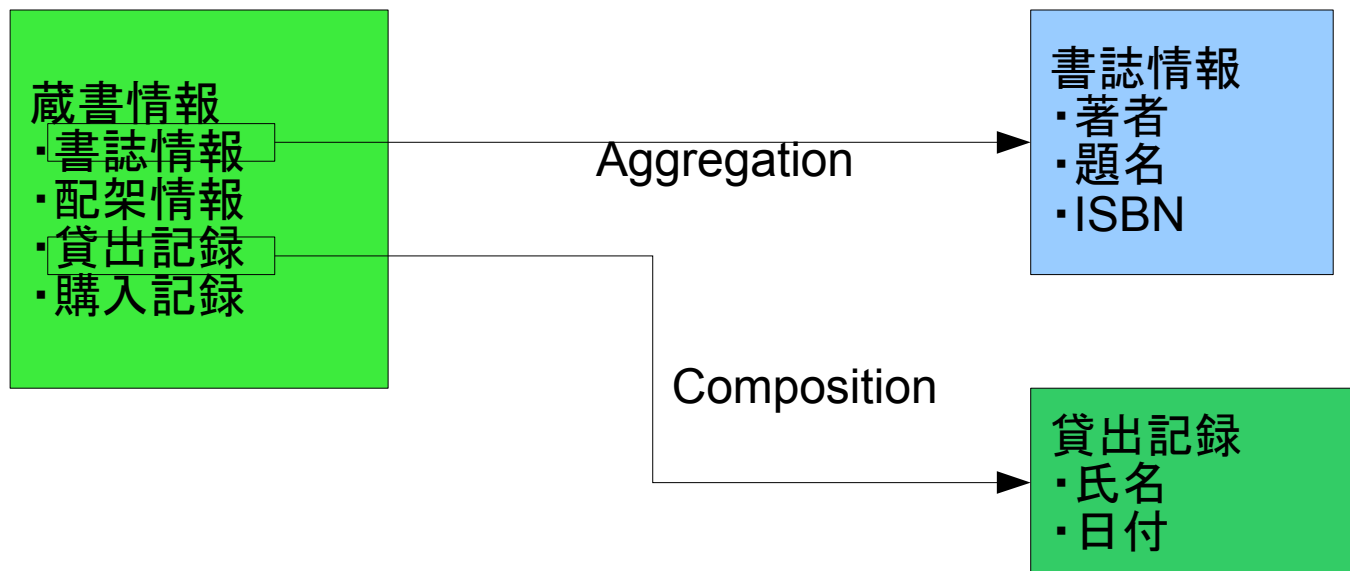
- オブジェクトの集まりをオブジェクトにする。
- 例：図書館の蔵書情報

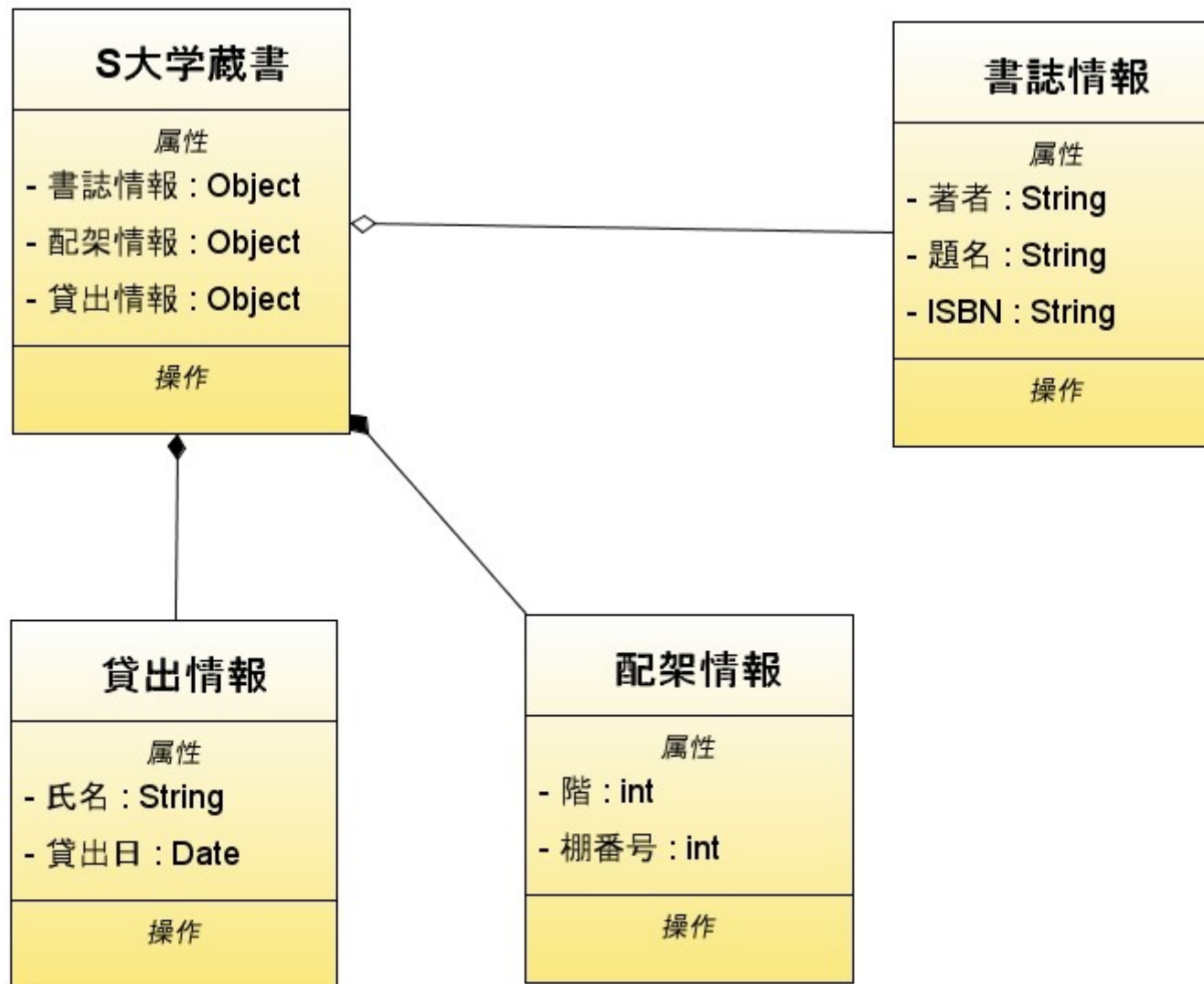
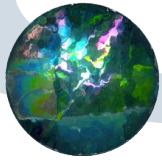


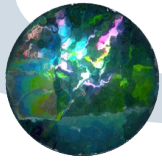


集約 (Aggregation) と 合成集約 (Composition)

- 集約：つながりの弱い場合
 - 書誌情報が外部DB：蔵書情報と独立に存在
- 合成集約：つながりが強い場合
 - 貸出情報は蔵書情報が無いと無意味

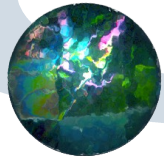






抽象クラス

- Methodが実装されていないクラス
- データの持ち方がほぼ決まっている
 - 一部methodが実装されていない
- 操作のされ方がほぼ決まっている
 - インターフェイスの名前が決まっている
 - 実装がされていない



多態性 (Polymorphism)

- 同じメッセージに対して異なる処理を行う

