情報実験I
プログラムの仕様とテスト

第2回
4月26日

本日の予定

- テストの実施
- プログラムの構造と責務の確認
- 仕様変更要求の説明
- 開発のヒントの説明

テストの実施

- 2人で1組になる。
- 他方のテストケースを用いて、自分のプログラムをテストする。

- 報告
  - 自分が定義したテストケースの数
  - 上記のテストケースの内、今回のプログラムの範囲に適用できるテストケースの数
  - テストケース提供者の学籍番号
  - テストケース提供者の氏名
  - 提供者のテストケースの内、今回のプログラムの範囲に適用できるテストケースの数
  - 上記の内、自分のプログラムが合格した数
  - 提供者のすべてのテストケース数
  - 上記の内、自分のプログラムが合格した数

プログラムの構造と責務の確認

- 指定の4つのクラスはあるか？
- 指定したクラス以外のクラスは何か？
- Commandクラスのコンストラクタは何をしているか？
- Boadクラスのコンストラクタは何をしているか？
- Rectangleクラスのコンストラクタは何をしているか？
- System.out はいくつのクラスに存在するか？
- System.inはいくつのクラスに存在するか？

リファクタリング

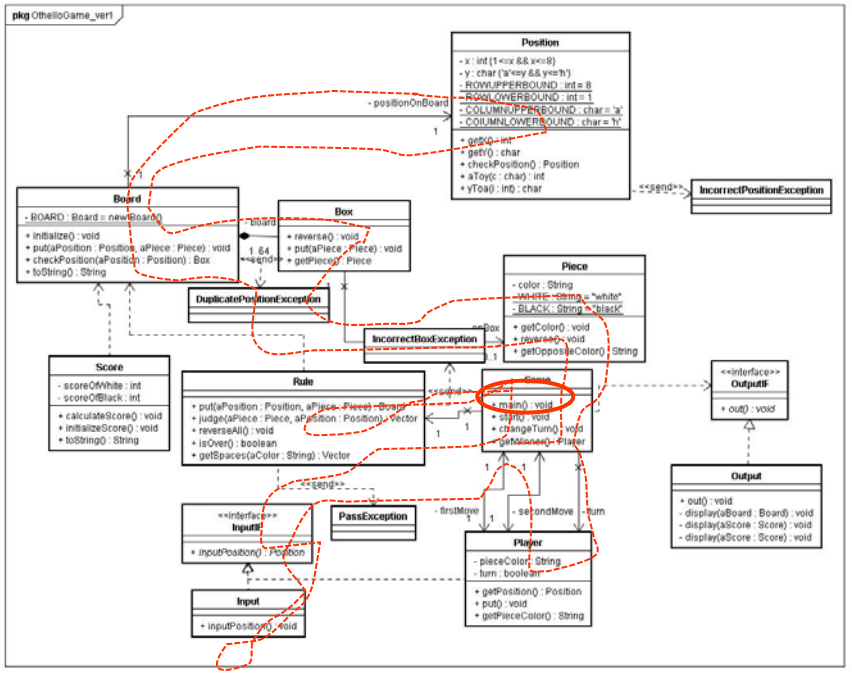
- 外部から見た時の振る舞いを保ちつつ、理解や修正が簡単になるようにソフトウェアの内部構造を変化させること
- Martin Fowler : Refactoring 1999
  - 72のリファクタリング
- ソフトウェアの体質改善

- コードの不吉な匂い
  - 重複したコード・長すぎるメソッド・巨大なクラス・多すぎる引数・基本データ型への執着等々

プログラムを読む

- 構文要素の意味を解釈する
  - 変数　型
    - クラス・インスタンスフィールド
    - メソッドローカル
  - 演算子
  - メソッド
  - 制御
  - 集合
  - データの構造化

プログラムを読む？



- 動的な側面
  - プログラムの開始点 main メソッド
    - mainメソッド内に書かれている手続きを順次たどる
- 静的な側面
  - プログラムは自分が定義した1つ以上のクラスで構成されている
  - ライブラリ（API）を使っている
    - クラスはどのような役割のフィールドとメソッドをもち、何をするのかというクラスの責務を読む

不吉な匂いとは

- コードは読んで理解できなければならない
- コードはコンピュータへの指示書
  - 手続きがmainメソッドを起点として、順番に書かれている
  - 反復、選択、メソッドの呼出し、例外処理の構造がある
  - 手続きを再利用できるように名前を付けたメソッドがある
  - データと手続きを一体化したある責務をもつクラスがある

- 理解しにくいとは？

不吉な匂い

- プログラム要素の意味が把握しにくい
  - マジックナンバー
    - 使われている数値の意味が把握しにくい。同じ数値はあちらこちらに使われる。
  - 制御フラグ
    - 意味のわからない制御用のフラグ変数が使用されている。
  - 不適切な命名
    - クラス・フィールド・メソッドの名前がわかりにくい。

不吉な匂い

- プログラム要素の何かが極端に大きいあるいは多い
  - 多数の同じ処理系列・1つのメソッド・1つのクラス・多数のメソッドの引数
    - 重複したコード
      - 同じコードがあちらこちらにある
    - 長すぎるメソッド
      - メソッド行数が長すぎる
    - 巨大なクラス
      - クラスのフィールドやメソッドの数が多すぎる
    - 多すぎる引数
      - メソッドに渡す引数が多すぎる

不吉な匂い

- プログラム要素の関連が複雑
  - 本来はクラス内でクラスに関することは閉じている
  - 変更とクラスは1対1にしたい
    - 変更の発散
      - 仕様変更が起きた時に、修正箇所があちらこちらに散らばっている
    - 変更の分散
      - 変更を行うたびにいろいろなクラスが変更されていく
    - 属性・操作の横恋慕
      - いつも他のクラスの中身をいじっている

不吉な匂い

- データと振舞いのまとまりがつくれていない
  - プログラムの対象のデータの扱い方がわかりにくい
    - データの群れ
      - まとめて扱うべき複数のデータが、1つのクラスにまとまっていない
    - 基本データ型への執着
      - クラスを作らずにintのような基本型ばかりを使っている
  - 振舞いがデータ毎に分離していない
    - スイッチ文
      - コードのあちらこちら(1つのクラスの複数のメソッド)に同じ場合わけがある
  - 振舞いが付随していないデータがある
    - データクラス
      - クラスがgetterメソッドとsetterメソッドしかもっていない

不吉な匂い

- 冗長・無駄・意味のない使い方でわかりにくいプログラムになっている
  - 怠け者クラス
    - クラスがたいした仕事をしていない
  - 一時的属性
    - 一時的にしか使用しないフィールドがある
  - 疑わしき一般化
    - いつかこのような拡張をするであろうとして一般化しすぎる
  - 仲介人
    - 委譲ばかりしていて、自分では仕事をしていないクラスがある
  - 不適切な関係
    - 必要がないのに双方向の関連をもっていたり、is-a関係がないのに継承を行っている