

# 2018 年度情報実験 I

2018 年 4 月 19 日 (木)

松浦佐江子

□ テーマ：プログラムの保守性とプログラムのテスト

□ 担当：(4 回) 4 月 19 日、26 日、5 月 10 日、17 日

□ 目的：オブジェクト指向開発に基づき、仕様変更に伴うプログラムの変更方法とプログラムのテストについて学ぶ。

□ キーワード：プログラムのテスト、リファクタリング、仕様変更、CUI から GUI へ

□ 問題：

一定の幅と高さをもつボードがある。図 1 のようにボード上には左上隅を原点とする座標系 (x 座標、y 座標) が定義されているとする。この時、つぎの条件を満たすように二辺の長さ(幅と高さ)と左上の位置を表す座標をもつ長方形をボード上に配置するプログラムを作成しなさい。座標系の単位はピクセルとする。ボードの大きさは最終版では指定したアプレットの大きさに依存して決定するものとするので、CUI の場合には固定値 500×400 とする。

(1) 配置に関する以下の操作を行うことができる。各操作の名前には下線が引かれている。

- a) 幅、高さ、左上の位置 (x 座標、y 座標) を与えて長方形を作成する。create
- b) 長方形 R を指定して、R を現在位置から指定した x 方向の距離  $x_0$ , y 方向の距離  $y_0$  だけ移動する。move
- c) 長方形 R を指定して、指定した幅の倍率  $mx$  で幅を、高さの倍率  $my$  で高さをそれぞれ拡大または縮小する。倍率は有限小数で与えるものとする。幅・高さの値は Math クラスの round メソッドを使用して決定する。expand/shrink
- d) 長方形 R を指定して削除する。delete
- e) ボード上の長方形をすべて削除する。deleteAll
- f) 2 つの長方形 R1 と R2 を指定し、R1 と R2 の重なり部分を抽出し、新たな長方形 R3 として配置する。intersect

(2) その他に次の操作を行うことができる。

- a) ボード上の長方形を表示する。displayBoard  
ここで「ボード上の長方形を表示する」とはボード上にあるすべての長方形の属性 (幅、高さ、x 座標、y 座標) を表示する (標準出力に出力する) という事。
- b) 配置に関する操作を終了する。exit

(3) ユーザはキーボード (標準入力) から(1)の操作名と必要なデータを入力し、プログラムが結果のボード上のデータ (どの位置にどの長方形が配置されているか) を標準出力に出力する。このプログラムは以下のように使用することができる。

- a) 起動すると、操作一覧が表示される。
- b) ユーザは操作を選択し、要求されるデータを入力する。
- c) 操作の実行が終了すると、操作一覧に戻り、exit が実行されるまで操作を選択することができる。

(4) 長方形に関する条件：

- a) ボード上で同じ幅、高さ、位置をもつ長方形は同一の長方形とみなす。

- b) 今回は点および線分は長方形とはみなさない。
  - c) ボード上に配置できる長方形の数の上限は 10 とする。
- (5) 操作に関する条件：
- a) ある操作によって、ボードから長方形がはみ出す場合には、その入力の値を無効として、操作をやり直す。
  - b) 操作が無効である場合は、適切なメッセージを出力する。
- (6) クラス構成に関する条件：
- RectangleEditor：mainメソッドをもつクラス
  - Rectangle：長方形のクラス
  - Board：長方形を配置するボードのクラス
  - Command: ユーザからの入力を受け取り、長方形に対する操作を呼び出し、ボードの状態を更新するクラス

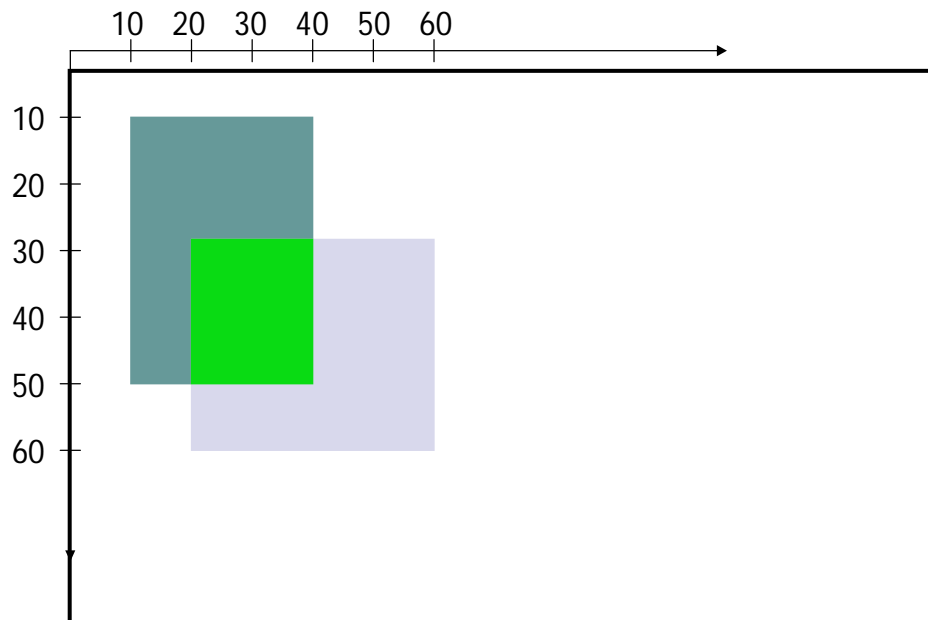


図1 ボードと長方形

※「ボード上のデータ（どの位置にどの長方形が配置されているか）」については、長方形は上記の属性がすべて等しい値の場合は同じとみなすことから、ここではこれら4つの属性でボード上の長方形を識別できるものとする。

□ 予定

第1回（4月19日）

- 実験の目的の説明
- プログラムの仕様の説明
- プログラムのテストに関する講義
- 各操作のテストケースを考える。
- プログラムを実装する。

次回までの課題：

- (1)(2)(3) (6)を満たすプログラムの作成（ただし、(1) の操作のうち a) create d)delete のみを実装する。）プログラムは全体を zip 形式にまとめて提出する。提出ファイルの構成は各自で定義する。
- 操作 create, delete のテストケースの定義（これは(4)(5)の要求も加味すること）  
形式は doc, docx, pdf, xls,xlsx,txt

## 第2回（4月26日）

- 2人1組になり、テストケースを交換し、自分のプログラムをテストする。
  - テスト結果を報告する・テストケースについて再考する。
- 機能拡張のためのリファクタリングについて
  - 自分のプログラムの評価
  - 入出力インタフェースの改善
- 拡張要求：
  - 色付長方形
- テストケースの追加

次回までの課題：

- 未実装操作（(1) b) c) e) f)、ただし、f)intersect はオプションとする）+ (4)と(5)の要求を満たすように変更・拡張したプログラムの作成
- テストケースの洗練と追加操作のテストケースの追加

## 第3回（5月10日）

- 2人1組になり、プログラムを交換し、自分のテストケースで相手のプログラムをテストする。
  - テスト結果を報告する・テストケースについて再考する。
- GUI の講義 (1)
- 仕様変更要求：出力の仕様

次回までの課題

- 出力を GUI 化したプログラムの作成

## 第4回（5月17日）

- 2人1組になり、プログラムを交換し、自分のテストケースで相手のプログラムをテストする。
  - テスト結果を報告する・テストケースについて再考
- GUI の講義 (2)
- 仕様変更要求：入力仕様

最終課題：

- 入出力を GUI 化したプログラムの作成
- テスト結果報告
- 考察

## テストケースの定義について

テストケースの定義については以下の内容がわかるように記述する。第1回で説明するテストの目的は何であったかを良く考えて記述すること。

- テスト対象：操作名
- テスト種別：正常ケース | 例外ケース
- テスト開始時の状態（例えば、「ボード上には長方形が1つもない」「ボード上には幅10高さ20左上隅の座標(20,30)の長方形が存在する」等々）
- テスト手順と必要な入力データの値（具体的な長方形の幅、高さ、左上隅の座標の値、等々）
- 期待されるテスト結果（例えば、「指定した長方形が削除されている」「入力値に対応する長方形は作成されていない」等々）

#### □評価

相互にテストすることが目的であるので、プログラムの提出は期限までに毎回行うものとする。  
これを行わないとテスト結果が報告できないので注意すること。

評価は以下の項目に対して行う。

- 第2回と第4回終了1週間後の12時までに提出したプログラム。ただし、完成している段階に応じて評価を行う。オプションを実装した場合は加点する。
- 授業中に行うテストの結果報告（授業時間内に報告）
- 第4回課題の考察(1週間後の12時までに提出)

なお、第4回課題の「考察」では

- 1 回目の課題から2 回目の課題への仕様変更に伴うプログラムの変更点
- 2 回目の課題から3 回目の課題への仕様変更に伴うプログラムの変更点
- 3 回目の課題から4 回目の課題への仕様変更に伴うプログラムの変更点

の観点から、仕様変更に伴うプログラムの変更について考察してもらいます。どこをどの程度変更する必要があったのか、\*\*のように作成したので、\*\*のクラスはそのまま使用することができたといったこと等をメモしておいてください。

資料の配布ならびにレポートの提出は大学のポータルサイト **Scomb** を使用して、電子的に行います。

#### ※開発のヒント

ソフトウェア設計論―役に立つUMLモデリングへ向けて― 松浦佐江子 著、コロナ社 3.3節