

Exercise 4

- 氏名: 栗林健太郎
- 学生番号: 2030006
- 作成日: 2020年9月19日

Class

Lecture Noteのpp.5-15で解説されているクラスについて説明する。

1. Point.java

クラス`Point`は、2次元平面における点の位置を表すものである。

主な操作として、原点からの距離や他の`Point`オブジェクトからの距離を取得する`distance`メソッド、指定された位置へ移動する`move`メソッドなどを持つ。また、スタティック変数`n`としてインスタンス化された回数を保持しており、`howManyPoints`メソッドでその数を取得できる。

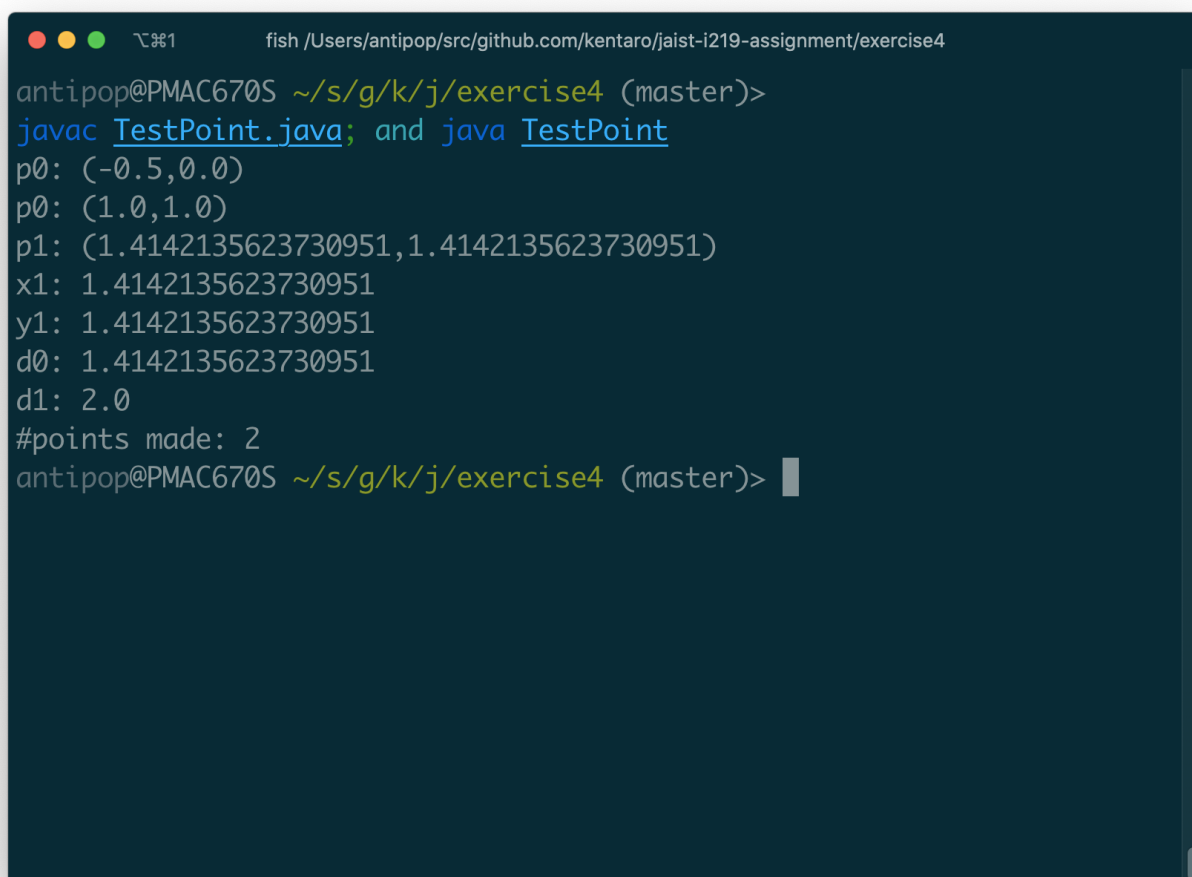
その他の特徴として、コンストラクタおよび`distance`メソッドがオーバーロードされている。

2. RandomWalking.java

クラス`RandomWalking`は、2つの`Point`クラスを生成し、一方をゴールを表す点、他方をゴール近傍を目指して移動する点としてランダムに動かし、`maxSteps`以内の試行でゴールに到達するか否かを判定する。

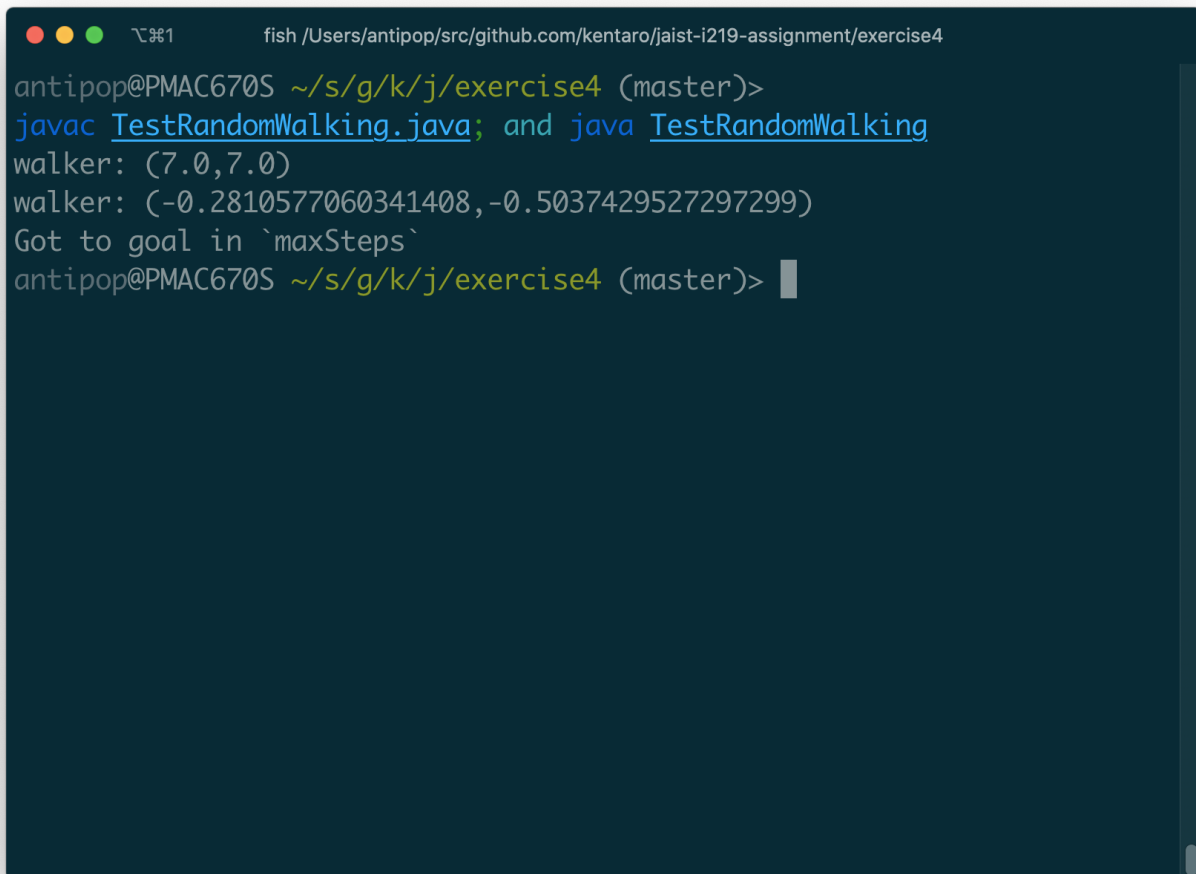
コードの実行

クラス`Point`については、以下の通り`TestPoint.java`をコンパイルした上で実行して動作を確認した。

A terminal window with a dark blue background and light green text. The window title bar shows three colored circles (red, yellow, green) and the text 'fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/exercise4'. The terminal content shows a user prompt 'antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>' followed by the command 'javac TestPoint.java; and java TestPoint'. The output consists of several lines of coordinates and values: 'p0: (-0.5,0.0)', 'p0: (1.0,1.0)', 'p1: (1.4142135623730951,1.4142135623730951)', 'x1: 1.4142135623730951', 'y1: 1.4142135623730951', 'd0: 1.4142135623730951', 'd1: 2.0', and '#points made: 2'. The terminal ends with the same user prompt and a cursor.

```
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
javac TestPoint.java; and java TestPoint
p0: (-0.5,0.0)
p0: (1.0,1.0)
p1: (1.4142135623730951,1.4142135623730951)
x1: 1.4142135623730951
y1: 1.4142135623730951
d0: 1.4142135623730951
d1: 2.0
#points made: 2
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
```

クラス`RandomWalking`については、以下の通り`TestRandomWalking.java`をコンパイルした上で実行して動作を確認した。

A terminal window with a dark blue background and light green text. The window title bar shows three colored circles (red, yellow, green) and the text 'fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/exercise4'. The terminal content shows a user 'antipop@PMAC670S' in a directory '~/s/g/k/j/exercise4' on the 'master' branch. They run 'javac TestRandomWalking.java; and java TestRandomWalking', which outputs 'walker: (7.0,7.0)' and 'walker: (-0.2810577060341408,-0.5037429527297299)', followed by 'Got to goal in `maxSteps`'. The prompt returns to 'antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>'.

```
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
javac TestRandomWalking.java; and java TestRandomWalking
walker: (7.0,7.0)
walker: (-0.2810577060341408,-0.5037429527297299)
Got to goal in `maxSteps`
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
```

Inheritance

Lecture Noteのpp.16-20で解説されている継承について説明する。

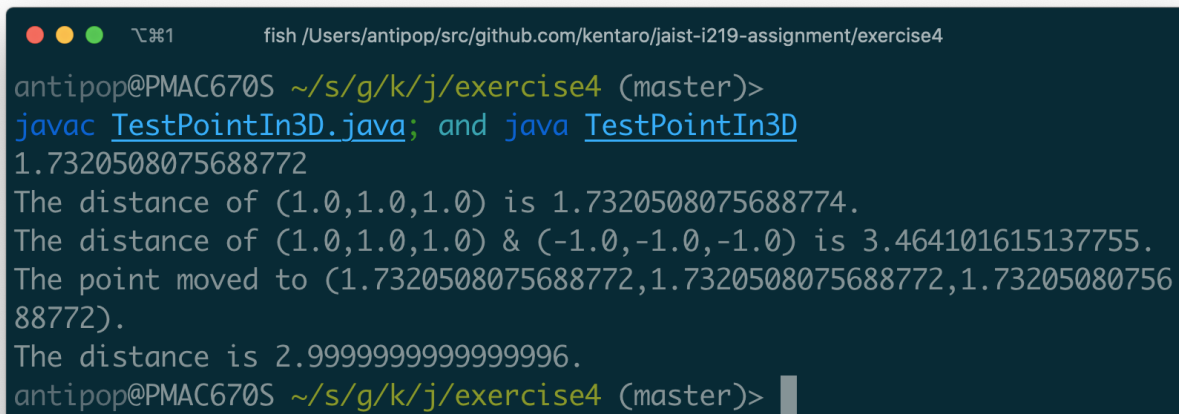
3. PointIn3D.java

クラス**PointIn3D**は、**Point**を継承し、3次元空間における点の位置を表すべく拡張するものである。

3次元空間上の点を表現するために**z**を導入した他、**distance**メソッドをオーバーライドした上で、2次元平面の距離を3次元空間上の距離とすべく計算内容を拡張している。その際、親クラスの処理を用いつつ拡張をするために**super**の呼び出しを用いている点が注目すべきポイントである。

コードの実行

クラス**PointIn3D**については、以下の通りTestPointIn3D.javaをコンパイルした上で実行して動作を確認した。

A terminal window with a dark blue background. The title bar shows three colored circles (red, yellow, green) and the text "fish /Users/antipop/src/github.com/kentaro/jaist-i219-assignment/exercise4". The prompt is "antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>". The user enters "javac TestPointIn3D.java; and java TestPointIn3D". The output is: "1.7320508075688772", "The distance of (1.0,1.0,1.0) is 1.7320508075688774.", "The distance of (1.0,1.0,1.0) & (-1.0,-1.0,-1.0) is 3.464101615137755.", "The point moved to (1.7320508075688772,1.7320508075688772,1.7320508075688772).", "The distance is 2.9999999999999996.", and the prompt "antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>".

```
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
javac TestPointIn3D.java; and java TestPointIn3D
1.7320508075688772
The distance of (1.0,1.0,1.0) is 1.7320508075688774.
The distance of (1.0,1.0,1.0) & (-1.0,-1.0,-1.0) is 3.464101615137755.
The point moved to (1.7320508075688772,1.7320508075688772,1.7320508075688772).
The distance is 2.9999999999999996.
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
```

Interface, Exception, Exception Handling, Type Cast

Lecture Noteのpp.21-28で解説されているインタフェース、例外、例外ハンドリング、型変換について説明する（実行するコードが資料上の章立てと一致しないので、まとめて記述する）。

4. PointInterface.java

p.21からは、先に導入した地点の表現を極座標系と直交座標系でもって表現する例が、これから説明するインタフェース、抽象クラスなどを用いて説明される。

このインタフェース`PointInterface`は、それらの階層関係のルートにあたるものであり、`getX()`、`getY()`の実装を要求する。

5. AbstractPoint.java

抽象クラス`AbstractPoint`は、`PointInterface`を実装する。また、3つの抽象メソッドを定義するとともに、`distance`メソッドをオーバーロードしている。

6. AbstractPointPolarCoordinates.java

抽象クラス`AbstractPointPolarCoordinates`は、`AbstractPoint`を継承し、`r`および`theta`で表される極座標系における点を表すクラスの基本的な操作を定義している。

また、`convertToRectangularCoordinates`として、自身の点を直交座標系へ変換する処理を定義しているのも注目すべきポイントである。

7. PointPolarCoordinatesRadian.java

クラス`PointPolarCoordinatesRadian`は、極座標系における点を表す前述の抽象クラス`AbstractPointPolarCoordinates`を継承し、`theta`がラジアンで表される場合に対応するものである。

このクラスのインスタンス生成時には、引数`r`および`theta`を検証し、不正な値が渡された場合には例外`PointException`が投げられる。

また、`r`および`theta`で表される点を、直交座標系におけるX軸上、Y軸上の座標へ変換するため、`getX()`および`getY()`が定義されている。親クラスで定義されている`convertToRectangularCoordinates`は、これらを用いて極座標系を直交座標系へと変換する処理を実現しているのも注目すべきポイントである。

8. PointException.java

例外`PointException`は、上述のクラス`PointPolarCoordinatesRadian`および、本レポートでは説明しないクラス`PointPolarCoordinatesDegree`のインスタンス生成時にわたされた引数が不正なときに投げられる例外を表すものである。

9. DistanceBetweenTwoPoints

クラス`DistanceBetweenTwoPoints`は、エントリーポイントmain関数を含む実行プログラムである。

`r`および`theta`をコマンドライン入力として2回受け取り、その二組みの値からそれぞれに対応する`PointPolarCoordinatesRadian`オブジェクトを生成する。このとき、`r`および`theta`として不正な値がわたされた場合には例外`PointException`が、数字として不正な値が渡された場合には例外`NumberFormatException`が発生するが、try-catch文によりプログラムが不正終了しないよう例外ハンドリングを行っている。

その後、`AbstractPoint[]`型の変数として`q`が宣言され、その`q`に格納する値を取得するために、`AbstractPoint[]`型の変数`p`の要素に対して`convertToRectangularCoordinates`メソッドを実行しようとしている。しかし、`p[j]`は`AbstractPoint`型であるため、そのメソッドを呼べるようにするために子クラスの`AbstractPointPolarCoordinates`に型変換をしているのも注目すべきポイントである。

コードの実行

上述したクラス等については、以下の通りTestPoint2.javaをコンパイルした上で実行して動作を確認した。

```
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
javac TestPoint2.java; and java TestPoint2
p1(x,y): (1.0000000000000002,1.0)
p2(x,y): (2.0,1.0)
p1(r,theta): <r: 1.4142135623730951,theta(in radian): 0.785398163397448
3>
p2(r,theta): <r: 2.23606797749979,theta(in radian): 0.4636476090008061>
p1->p2: 0.9999999999999998
p2->p1: 0.9999999999999998
PointException: r should not be negative!
PointException: theta should satisfy inequality: 0.0 <= theta <= 2.0
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
```

クラスDistanceBetweenTwoPointsについては、以下の通りDistanceBetweenTwoPoints.javaをコンパイルした上で実行して動作を確認した。

```
antipop@PMAC670S ~/s/g/k/j/exercise4 (master) [SIGINT]>
javac DistanceBetweenTwoPoints.java; and java DistanceBetweenTwoPoints
Input two points in polar coordinate system
r0: 7
theta0: 0.57
r1: 10
theta1: 1.2
<r: 7.0,theta(in radian): 0.57>-><r: 10.0,theta(in radian): 1.2>: 5.989
670177588973
<r: 7.0,theta(in radian): 0.57>: (5.893306826135881,3.7774243411377846)
<r: 10.0,theta(in radian): 1.2>: (3.623577544766736,9.320390859672262)
antipop@PMAC670S ~/s/g/k/j/exercise4 (master)>
```