オブジェクト指向プログラミングⅠ演習問題 5

2017 年 7 月 3 日

学習ポイント
● アクセス制御 ● 例外処理

1. 6 月 12 日問題 3 で用いた Point クラスに下記のようにアクセス制御をつけた。コンパイル時にエラーが生じるかを調べよ。対処方法を考え、プログラムを修正し、Scomb のアンケートで報告しなさい。

```
class Point{
        private double x;
        private double y;
        public Point (){
                this.x = 0.0;
                this.y = 0.0;
        }
        public Point(double x, double y){
                this.x = x;
                this.y = y;
        }
        public Point linearTransfer(){
                double x0 = 6*this.x + 4* this.y;
                double y0 = (-2) * this.x + 1* this.y;
                return new Point(x0,y0);
        }
        public String toString(){
                return ("(" + this.x + ", " + this.y + ")");
        }
}
```

2. スタックの基本操作は push と pop の 2 つである。push はデータをスタックに追加する操作であり、pop は最後に push されたデータを取り出す操作である。スタックを固定長配列と、下記に定義する Cell クラスを使って 2 通りに定義しなさい。どちらの問題も、スタックのクラス名は Stack とする。また、使用するクラス Cell も含めて、すべてのクラスのフィールドならびにメソッドのアクセス制御を適切に定義しなさい。さらに2つの定義に対し、push および pop に関していろいろなテストをプログラム内で行いどのような場合に何という例外が発生するか（エラーメッセージにある例外クラス名とそのエラーが発生する状況の説明）を画面に表示しなさい。なお，2 つの実装方法はパッケージを分けて定義すること。

```
class Cell{
	private Object element;
	private Cell next;
	Cell(Object element){
		this.element = element;
	}
	public Object getElement(){
		return element;
	}
	public Cell getNext(){
		return next;
	}
	public void setElement(Object element){
		this.element = element;
	}
	public void setNext(Cell next){
		this.next = next;
	}
	public String toString(){
		String s = (String)this.element;
		if (this.next == null){
					return s;
			}
			else {
			return (s + " <-- " + (this.next).toString());
			}
	}
}
```