プログラミングⅡ　2022　第４回 演習問題　「クラスとインスタンス（第８章)」

下記の各問に対して Java のプログラムを作成せよ。次回までに，ソースコードと出力結果を専用の提出用紙のフォーマットに入力して提出せよ。

1. 以下は，ロボットをシミュレーションするプログラムの例である。指示に従ってコードを追加してプログラムを完成させよ。完成して動作が確認できたところまでを提出せよ。

(1)　まず下記の内容を含むファイルを作成せよ。Java ソースコードのファイル名は何か。

```
public class RobotWorld {
    public static void main(String [] args) {
        Robot robo1;
    }
}
```

(2)　(1)のフォルダに，ロボットを表す「クラス」Robot の定義を記述したファイル Robot.java を追加し，空のクラス Robot の定義を記述せよ。この時点で結果は表示されないが，プログラムはコンパイルしてエラーなく実行できることを確認せよ。

(3)　ロボットは名前（文字列 name）と 2 次元座標（整数 x と y）を持つものとする。Robot.java のクラス定義に必要な「フィールド」を書き加えよ。

(4)　(1)で作成したファイルの main メソッドのなかで，(2)で定義したクラス Robot の「インスタンス」を 1 つ作成し，そのインスタンスを変数 robo1 に代入する処理を追加せよ。

(5)　(4)で作成したインスタンスが持つ名前のフィールドに「ロボコ」，2 次元座標のフィールドに $(1, 2)$ を代入する処理を追加せよ。main の中に記述せよ。

(6)　(5)の処理の後に，インスタンス robo1 のすべてのフィールド（name, x, y）の値を println を用いて表示する処理を追加せよ。プログラムを実行して表示結果を確認せよ。

(7)　(6)の処理の後に，変数 robo2 を定義して新しいロボットのインスタンスを代入し，その名前のフィールドに「ロボタ」， 2 次元座標のフィールドに $(3, 4)$ を代入する処理を追加せよ。

(8)　クラス Robot に，例えば「ロボコ：ワタシハ $(1, 2)$ ニイマス」のようにロボットの名前と座標値を表示するメソッド speak()を追加し，最後に robo1 と robo2 の speak()を実行させて結果を確認せよ。

2. 前問の Robot のクラス図と，「ロボコ」と「ロボタ」のインスタンス図（オブジェクト図）を示せ。

3. 以下に UML のクラス図を示した長方形を表すクラス Rectangle を定義せよ（public なクラスにすること）。Rectangle は，属性として縦（height）と横（width）の長さを持ち，操作として面積を求める double area()を持つ（area()には引数がないことに注意）。

Rectangle
height : double width : double
area() : double

次に，main メソッドだけを持つクラスを作成し，キーボードから値を読み込むことで，以下の UML のオブジェクト図（インスタンス図）で示す Rectangle のインスタンスを生成し，その面積を計算して表示するプログラムを作成せよ。

goldenRect : Rectangle
height = 1.0 width = 1.618

4. 以下に UML のクラス図で示した円を表すクラス Circle を定義せよ（public なクラスにすること）。Circle は属性として半径の長さ（radius）と定数の円周率（PI=3.14159）を持ち，操作として面積を求める double area()と円周の長さを返す double circum()を持つ。

Circle
radius : double PI : double = 3.14159
area() : double circum() : double

次に，main メソッドだけを持つクラスを作成し，そのなかで 2 つの要素を持つ Circle の配列を定義して，半径 100 と 200 の円のインスタンスを生成し，クラス Circle を用いてそれぞれの面積と円周の長さを計算して表示するプログラムを作成せよ。以下に，ヒントとして穴埋めにしたプログラムの一部を示す。

```
Circle[] circles = new

circles[0] = new
circles[0].radius =
System.out.println("半径 = " + circles[0].         );
System.out.println("面積 = " + circles[0].         );
System.out.println("円周 = " + circles[0].         );

circles[1] = new
circles[1].radius =
System.out.println("半径 = " + circles[1].         );
System.out.println("面積 = " + circles[1].         );
System.out.println("円周 = " + circles[1].         );
```

5. 下記の(1)〜(8)の指示に従って Java プログラムを作成し，完成したところまでのプログラムを示せ。

(1) ゲームキャラクタを表すクラス Ninja を定義せよ。Ninja は，フィールドとして縦横 5 マスのステージ上の座標 x, y（初期値は左上隅を示す 0, 0）と体力 hp（初期値 10）を持つ。また，メソッドとしては上下左右の方向へ 1 マス移動して体力を 1 だけ減らす up(), down(), left(), right()と，現在位置および体力を表示する show()を持つ。各移動メソッドでは，ステージ内からはみ出さない処理も行うこと。

(2) 次に，main メソッドだけを持つクラス Game を定義し，キャラクタの体力が 0 以上の間，キーボードからプレイヤーの方向指示を読み込んで，キャラクタを移動させるプログラムを作成せよ。実行して移動させるたびにキャラクタの体力が減ることを確認せよ。下記は一部分の例である。

```
public static void main(String[] args) {
    Ninja ninja = new Ninja();
    Scanner sc = new Scanner(System.in);

    while (ninja.hp > 0) {
        System.out.print("方向選択（1:左，2:上，3:下，4:右）→ ");
        int dir = sc.nextInt();
        switch (dir) {
        case 1:
            ninja.left();
            break;
        // 以下省略
        }
        ninja.show();
    }
}
```

(3) キャラクタが右下隅 (4, 4) に到達したら，「任務成功！」と残りの体力を表示してプログラムを終了する処理を追加せよ。

(4) 罠（わな）を表すクラス Trap を定義せよ。Trap は，フィールドとして座標 x, y を持ち，メソッドとしてキャラクタが同じ座標にいれば体力を 1 だけ減らす void check(Ninja ninja)を持つ。

(5) 罠を 5 個生成する。キャラクタを動かす前に Trap の配列を定義し，各要素に Trap のインスタンスを生成して代入し，それぞれの座標 x,y には乱数で発生させた値を代入せよ。0 以上 n 未満の整数の乱数を発生させるには，Ramdom クラスのメソッド nextInt(n)を用いればよい（教科書 p.91）。

```
Trap[] traps = new
Random rnd = new
for (int i = 0; i < traps.length; i++) {
    traps[i] =
    traps[i].x =
    traps[i].y =
}
```

(6) 毎回キャラクタを動かした後に，すべての罠の check メソッドを実行する処理を追加せよ。罠にかかったときにキャラクタの体力が減ることを確認せよ。