

問題 1(計 85 点・部分点有り)

製図を行うためのプログラムを考える。定義するクラス・インタフェイスは以下の通りである。クラス及びインタフェイス (a) ~ (d) の各定義を含むそれぞれのソースファイルの内容を解答欄 1-(a) ~ (d) に書け。なお、(e) の Test クラスは 2 枚目に記載・定義済みである。

- (a) 2次元図形一般を表す public な抽象クラス Fig2D。com.tuis.drawing.figure パッケージに属する。次のメンバを持つ。
- ・生成された Fig2D 型オブジェクトの総数を表す int 型の private な static フィールド num(初期値は 0)
 - ・2次元図形の位置 (2次元座標) を表す private な double 型フィールド x,y
 - ・num の public で static なゲッタ int getNum()
 - ・x の public なゲッタ double getX()
 - ・x の public なセッタ void setX(double x)
 - ・y の public なゲッタ double getY()
 - ・y の public なセッタ void setY(double y)
 - ・その 2次元図形の面積を計算して返す public な抽象メソッド double area()
 - ・フィールド x,y を初期化し、num の値を 1 増やす public なコンストラクタ Fig2D(double x, double y)
- (b) 「描画が可能である」という性質を表す public なインタフェイス Drawable。com.tuis.drawing.system パッケージに属する。次のメンバを持つ。
- ・「その図形を描画する」という処理を表す抽象メソッド void draw()
- (c) 「円」を表す public なクラス Circle。Fig2D のサブクラスで、Drawable を実装する。com.tuis.drawing.figure パッケージに属する。Fig2D から継承したフィールド x,y は、この円の中心座標として扱う。この Circle は次のメンバを持つ。
- ・円の半径を表す private な double 型フィールド r
 - ・Fig2D のコンストラクタを呼び出しフィールド x,y を初期化し、さらにフィールド r を初期化する public なコンストラクタ Circle(double x, double y, double r)
 - ・円の面積を返すようにオーバーライドした public な area メソッド。使用する円周率は 3.14 でよい。
(ヒント：円の面積は半径×半径×円周率)
 - ・円を表す文字 'o' を表示するようにオーバーライドした public な draw メソッド
- (d) 「長方形」を表す public なクラス Rectangle。Fig2D のサブクラスで、Drawable を実装する。com.tuis.drawing.figure パッケージに属する。Fig2D から継承したフィールド x,y は、この長方形の左上の頂点の座標として扱う。この Rectangle は次のメンバを持つ。
- ・縦辺の長さを表す private な double 型フィールド h と、横辺の長さを表す private な double 型フィールド w
 - ・Fig2D のコンストラクタを呼び出しフィールド x,y を初期化し、さらにフィールド h,w を初期化する public なコンストラクタ Rectangle(double x, double y, double h, double w)
 - ・長方形の面積を返すようにオーバーライドした public な area メソッド。
 - ・長方形を表す文字 'r' を表示するようにオーバーライドした public な draw メソッド
- (e) 動作テストを行う public なクラス Test(現状では 2 枚目 1-(e) 欄のように定義されている)

問題 2(計 5 点・部分点有り)

問題 1 で定義したクラスとインタフェイス (Fig2D,Drawable,Circle,Rectangle) の継承関係・実装関係を矢印を使って解答欄 2 に図示せよ。

問題 3(計 10 点・部分点有り)

まず、「is-a 関係」とは何かについて解答欄 3 に述べよ。そして、問題 1 で作成したプログラムで、クラス Test の letItCalcArea(Fig2D f) メソッドが、Circle 型オブジェクトと Rectangle 型オブジェクトを実引数として受け取れる理由を Circle, Rectangle, Fig2D の関係から説明せよ。また、クラス Test の letItDraw(Drawable d) メソッドが、Circle 型オブジェクトと Rectangle 型オブジェクトを実引数として受け取れる理由を Circle, Rectangle, Drawable の関係から説明せよ。

解答欄 1-(a) Fig2D.java の内容。30 点(部分点有り) スペースが足りない場合は、「裏へ続く」と書いて裏面に続きを書け。



解答欄 1-(b) Drawable.java の内容。15 点 (部分点有り)

解答欄 1-(d) Rectangle.java の内容。20 点 (部分点有り)

解答欄 1-(c) Circle.java の内容。20 点 (部分点有り)

1-(e) 参考: クラス Test の定義を含む Test.java の内容。

```
import com.tuis.drawing.figure.Fig2D;
import com.tuis.drawing.system.Drawable;
import com.tuis.drawing.figure.Circle;
import com.tuis.drawing.figure.Rectangle;

public class Test {
    static void letItCalcArea( Fig2D f ) {
        System.out.println( "面積は " + f.area( ) );
    }
    static void letItDraw( Drawable d ) { d.draw( ); }
    public static void main( String args[ ] ) {
        Circle c = new Circle( 10.0, 10.0, 3.0 );
        Rectangle r = new Rectangle( 20.0, 30.0, 4.5, 2.0 );
        letItCalcArea( c ); letItCalcArea( r );
        letItDraw( c ); letItDraw( r );
        System.out.println( "現在の 2 次元図形の総数は"
            + Fig2D.getNum( ) );
    }
}
```

解答欄 2 5 点 (部分点有り)

解答欄 3 10 点 (部分点有り)