

プログラム設計 前期期末試験

全てのプログラムファイルの先頭行に、コメントとして自分の番号と名前を書くこと。

準備 試験で利用可能なコードが事前に用意されている。端末内で、以下のコマンドを実行してコピーしておくこと。

```
$ cp /usr/local/common/kogai/201808pd.java . (←ここにピリオド)
```

1 次のような「半径と円周率」を表すスーパークラス Circle を考える。(コピーしたファイルから入手可能)

```
class Circle {
    protected double pi;    //円周率
    protected int r;        //半径
    public Circle(int r) {
        pi = 3.14159265359;
        this.r = r;
    }
    public void show() {
        System.out.printf("半径: %d\n", r);
    }
    public void calc() {
        System.out.printf("計算結果: _\n");
    }
}
```

この Circle を継承して、次のようなフィールドとメソッドが追加されたクラス Ellipse (楕円) と Arc (円弧) をそれぞれ作成しなさい。

「楕円を表す」クラス Ellipse (クラス Circle を継承する)

- フィールドとして、もう一つの半径を表す r2 (int 型) を持つ
- 以下のようなコンストラクタを持つ

```
public Ellipse(int r, int r2)
    //引数付き super() を使って、フィールド r に引数 r を代入する
    //フィールド r2 に引数 r2 を代入する
```

- 楕円の情報を出力するメソッド show をオーバーライドする

```
public void show()
    //スーパークラスのメソッド show を呼び出す
    //フィールド r2 を出力する (出力の様子は実行結果を参照)
```

- 楕円の情報から「楕円の面積」を計算するメソッド calc をオーバーライドする
なお、2 つの半径 $r, r2$ を持つ楕円の面積は「 $r \times r2 \times \pi$ 」である

```
public void calc()
    //フィールド r, r2, pi を使って楕円の面積を計算し、出力する (出力の様子は実行結果を参照)
```

「円弧を表す」クラス Arc (クラス Circle を継承する)

- フィールドとして、中心角を表す theta (int 型で単位は度) を持つ
- 以下のようなコンストラクタを持つ

```
public Arc(int r, int theta)
    //引数付き super() を使って、フィールド r に引数 r を代入する
    //フィールド theta に引数 theta を代入する
```

- 円弧の情報を出力するメソッド show をオーバーライドする

```
public void show()
    //スーパークラスのメソッド show を呼び出す
    //フィールド theta を出力する（出力の様子は実行結果を参照）
```

- 円弧の情報から「円弧の面積」を計算するメソッド calc をオーバーライドする
なお、2 つの半径 r と中心角 θ を持つ円弧の面積は「 $\frac{r^2 \times (\theta \times \pi \div 180)}{2}$ 」である

```
public void calc()
    //フィールド r, theta, pi を使って円弧の面積を計算し、出力する（出力の様子は実行結果を参照）
```

main の処理とその実行結果は以下のようになる。（コピーしたファイルから入手可能）

[main を持ったクラス]

```
class Pd1term1 {
    public static void main(String[] args) {
        Circle c1 = new Circle(3);
        Ellipse e1 = new Ellipse(4, 2);
        Arc a1 = new Arc(5, 120);
        System.out.println("-----show-----");
        c1.show();
        e1.show();
        a1.show();
        System.out.println("-----calc-----");
        c1.calc();
        e1.calc();
        a1.calc();
    }
}
```

[実行結果]

```
-----show-----
半径: 3                (← c1.show() の結果)
半径: 4                (← e1.show() の結果)
もう一つの半径: 2
半径: 5                (← a1.show() の結果)
中心角: 120
-----calc-----
計算結果: _           (← c1.calc() の結果)
楕円の面積: 25.132741 (← e1.calc() の結果)
円弧の面積: 26.179939 (← a1.calc() の結果)
```

- 2** 前問で示したクラス Circle の配列を作り、配列の要素となるインスタンスに対して、メソッド show と calc 呼び出しの繰り返し処理をする main を持ったクラス Pd1term2 を作成しなさい。ただし、配列には下表のような情報を持ったインスタンスが格納されるように作ること。

添字	半径	もう一つの半径	中心角
0	6	—	—
1	9	—	200
2	10	5	—
3	3	—	90

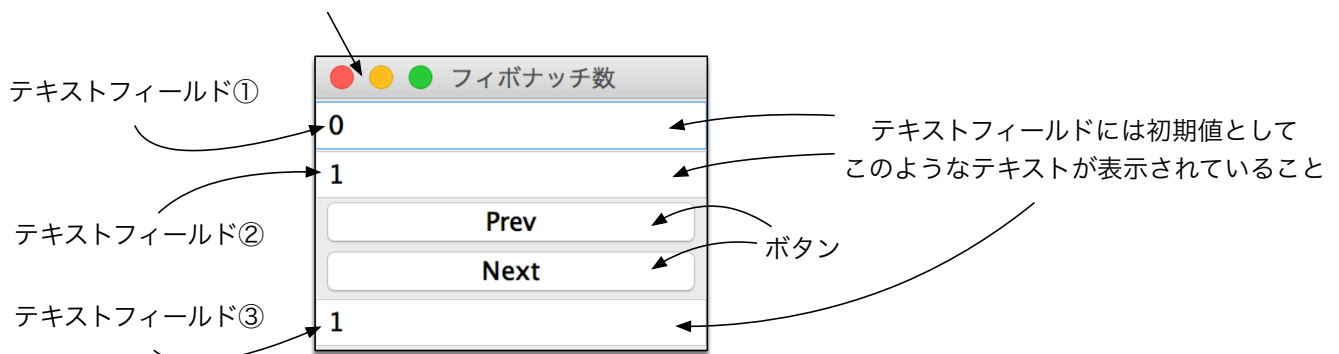
※表中の「—」欄は不要な列を表している

実行結果は以下のようになる。

```
[実行結果]
--- circles[0]---
半径: 6
計算結果: _
--- circles[1]---
半径: 9
中心角: 200
円弧の面積: 141.371669
--- circles[2]---
半径: 10
もう一つの半径: 5
楕円の面積: 157.079633
--- circles[3]---
半径: 3
中心角: 90
円弧の面積: 7.068583
```

3 「フィボナッチ数」を計算するためのアプリケーションを作ること考える。テキストフィールド、ボタンの部品が下図のように配置されたウィンドウが表示されるアプリケーションを作りなさい。

フレーム（このフレームはGridLayoutで配置する）



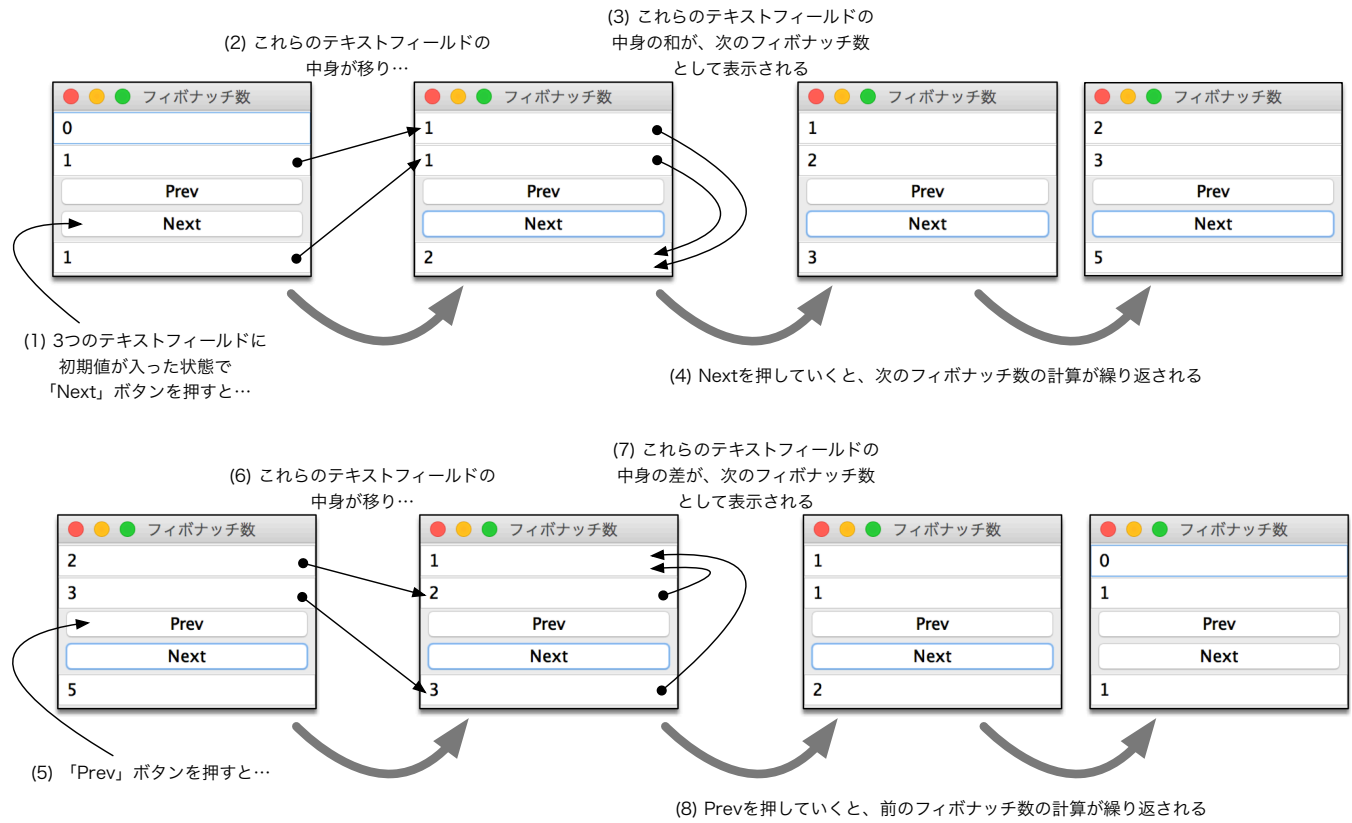
※ フィボナッチ数列は、前の2つの項の和が次の項になる、つまり…
 テキストフィールド①の整数 + テキストフィールド②の整数 = テキストフィールド③の整数

4 前問で作成したアプリケーションに、次に示すような「次のフィボナッチ数/前のフィボナッチ数を計算する」動作を追加しなさい。

- 3つのテキストフィールドにフィボナッチ数の初期値（前問で示した初期値）を入力して、ボタンを押すと以下のような動作をする
- Next ボタンを押すと、次のフィボナッチ数を求める（動作の詳細は下図参照）
 - Prev ボタンを押すと、前のフィボナッチ数を求める（動作の詳細は下図参照）

※ String 型と int 型の相互変換には、第 13 週の内容を参照

この動作の様子は下図のようになる。



問題はここまで (各 25 点)

定期試験の実施について

試験中に使用できるもの

- 筆記用具 (メモ用紙は必要な人に配布)
- 演習室のコンピューター一台 (一つの机に一人の配置で、座る場所はどこでもよい)

試験中に参照できるもの

- 自分のホームディレクトリ (ホームフォルダ) 以下に保存されているファイル (定期試験では紙媒体のものは参照不可)
- * **上記以外の情報を参照することは不正行為とする** (例: USB で接続された機器に保存されているファイルの参照など)
- * 試験中は、演習室外へのネットワークアクセスは遮断される

答案の提出

- 提出する全てのプログラムファイルの先頭行に、**自分の学科の出席番号と氏名**をコメントとして書く
- 保存したファイルは次のように「report」コマンドで提出する (ちゃんと提出できた場合は、「Succeed.」と画面に表示される)
\$ ~kogai/report pd1term 「プログラムファイル」
- 複数のファイルを提出する場合は、report コマンドを分けて提出する
例えば、test1.java と test2.java のファイルを提出したい場合は、次のように 2 回に分けて提出する
\$ ~kogai/report pd1term test1.java
\$ ~kogai/report pd1term test2.java
- 同じ問題に対して、複数の提出ファイルが存在した場合は、更新日時が新しい方を提出ファイルとする

前期期末試験 模範解答 (平均 81.5 点)

答案用紙に記載されている点数は以下のようにになっています。

前期中間成績 (X) 前期期末試験 小テスト平均 前期期末成績 (Y) 前期成績 (XとYの平均)

採点について コンパイル時にエラーとなる箇所は -4 点, 実行可能だが処理内容が問題の意図と違う箇所は -2 点を基本とする。

配点: 1 ~ 4 各 25 点

1, 2 のプログラム

```
class Circle {
    protected double pi;    //円周率
    protected int r;        //半径
    public Circle(int r) {
        pi = 3.14159265359;
        this.r = r;
    }
    public void show() {
        System.out.printf("半径: %d\n", r);
    }
    public void calc() {
        System.out.printf("計算結果: _\n");
    }
}

class Ellipse extends Circle {
    private int r2; //もう一つの半径
    public Ellipse(int r, int r2) {
        super(r);
        this.r2 = r2;
    }
    public void show() {
        super.show();
        System.out.printf("もう一つの半径: %d\n", r2);
    }
    public void calc() {
        System.out.printf("楕円の面積: %f\n", r*r2*pi);
    }
}

class Arc extends Circle {
    private int theta; //中心角
    public Arc(int r, int theta) {
        super(r);
        this.theta = theta;
    }
    public void show() {
        super.show();
        System.out.printf("中心角: %d\n", theta);
    }
    public void calc() {
        System.out.printf("円弧の面積: %f\n", r*r*(theta*pi/180)/2);
    }
}

class Pd1term1 {
    public static void main(String[] args) {
        Circle c1 = new Circle(3);
        Ellipse e1 = new Ellipse(4, 2);
        Arc a1 = new Arc(5, 120);
        System.out.println("-----show-----");
        c1.show();
        e1.show();
        a1.show();
        System.out.println("-----calc-----");
        c1.calc();
        e1.calc();
        a1.calc();
    }
}
```

```

class Pdterm2 {
    public static void main(String[] args) {
        Circle[] circles = new Circle[4];
        circles[0] = new Circle(6);
        circles[1] = new Arc(9, 200);
        circles[2] = new Ellipse(10, 5);
        circles[3] = new Arc(3, 90);
        for(int i=0; i<circles.length; i++) {
            System.out.printf("--- circles[%d]---\n", i);
            circles[i].show();
            circles[i].calc();
        }
    }
}

```

3, 4 のプログラム

```

import java.awt.*;
import java.awt.event.*;
class FiboApp implements ActionListener {
    Frame f1;
    Label op;
    TextField num1, num2, num3;
    Button prev, next;
    FiboApp() {
        f1 = new Frame("フィボナッチ数");
        f1.setLayout(new GridLayout(5, 1));
        //部品を作る
        num1 = new TextField("0");
        num2 = new TextField("1");
        num3 = new TextField("1");
        prev = new Button("Prev");
        next = new Button("Next");
        //フレームに部品を追加する
        f1.add(num1);
        f1.add(num2);
        f1.add(prev);
        f1.add(next);
        f1.add(num3);
        //アクションリスナを登録する
        prev.addActionListener(this);
        next.addActionListener(this);
        f1.setSize(200, 150);
        f1.setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==next) {
            //入力された数値を取得する
            String s2 = num2.getText();
            String s3 = num3.getText();
            //取得した数値 (文字列) を整数に変換する
            int n2 = Integer.parseInt(s2);
            int n3 = Integer.parseInt(s3);
            //結果を文字列に変換し x に代入する
            String x = String.valueOf(n2+n3);
            num1.setText(s2);
            num2.setText(s3);
            num3.setText(x);
        }
        if(e.getSource()==prev) {
            String s1 = num1.getText();
            String s2 = num2.getText();
            int n1 = Integer.parseInt(s1);
            int n2 = Integer.parseInt(s2);
            String x = String.valueOf(n2-n1);
            num1.setText(x);
            num2.setText(s1);
            num3.setText(s2);
        }
    }
    public static void main(String[] args) {
        new FiboApp();
    }
}

```