

プログラム設計 前期期末試験

準備 プログラムを作る前に、以下の操作をしてファイルの準備をしておくこと。

1. 授業の配付資料を全てダウンロードする場合は、以下を実行する（既に実行済みの場合は不要）
`$ mygitclone-pd`
2. GitHub から自分のリポジトリを clone しておく（既に実行済みの場合は不要）
`$ mygitclone4d` 自分の GitHub ユーザ名
3. リポジトリのフォルダに移動して、設定用のスクリプトを実行する
`$ cd ~/prog4d-` ユーザ名
`$./myconf`
4. 今回の定期試験用のフォルダをコピーする
`$ cp -r /usr/local/common/kogai/pd/test1term .` （←ここにピリオド）
`$ cd test1term` （コピーしたフォルダに移動する）
`$ ls` （フォルダ内のファイルを確認すると、以下のファイルがコピーされている）
`12_event.java` `34_calc.java`
5. テキストエディタでプログラムを開き、**まず先頭行にコメントとして自分の番号と名前を書いて、解答を始める**
`$ gedit` 各問のファイル名 `&`

1 次のような「カレンダーに記入するスケジュール（イベント）」を扱うスーパークラス `EventBase` を考える。

```
class EventBase {
    protected String title;
    protected int year;
    protected int month;
    protected int day;
    public EventBase(String t, int y, int m, int d) {
        title = t;
        year = y; month = m; day = d;
    }
    public void show() {
        System.out.printf("(%d 年 %d 月 %d 日) %s\n", year, month, day, title);
    }
}
```

この `EventBase` 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 `EventPlace` と `EventTime` をファイル「12_event.java」に作成しなさい。

「場所の情報を持つ」クラス `EventPlace`（クラス `EventBase` を継承する）

- フィールドとして、イベントの場所を表す `String` 型の `place` を持つ。
- 以下のようなコンストラクタを持つ。

```
public EventPlace(String t, int y, int m, int d, String p)
    //super() を使って、フィールド title に t を、year に y を、month に m を、day に d を代入する
    //フィールド place に p を代入する
```

- 次のようなメソッド `show` をオーバーライドする。

```
public void show()
    //super を使って、スーパークラスのメソッド show を呼び出す
    //フィールド place を出力する（出力の様子は、以下に示す実行結果を参照）
```

「時間の情報を持つ」クラス EventTime (クラス EventBase を継承する)

- フィールドとして、イベントの開始時間を表す int 型の start と、終了時間を表す int 型の end を持つ。
(これらには、0 時～23 時を表す整数が入ることを想定している)
- 以下のようなコンストラクタを持つ。

```
public EventTime(String t, int y, int m, int d, int s, int e)
    //super() を使って、フィールド title に t を、year に y を、month に m を、day に d を代入する
    //フィールド start に s を、end に e を代入する
```

- 次のようなメソッド show をオーバーライドする。

```
public void show()
    //フィールド title, year, month, day, start, end を出力する
    //ただし、end の値によって、時間情報の出力は以下のように変わる
    //    ・end の値が-1 の場合、時間は開始時間 (start) のみを出力する
    //    ・それ以外の場合、時間は開始時間 (start) ～終了時間 (end) を出力する
    //(出力の様子は、以下に示す実行結果を参照、なおスーパークラスの show は使っていない)
```

main の処理とその実行結果は以下のようになる。

[main を持ったクラス]

```
class Pd1term1 {
    public static void main(String[] args) {
        EventBase e1 = new EventBase("海の日", 2019, 7, 15);
        EventPlace e2 = new EventPlace("インターンシップガイダンス", 2019, 8, 2, "大教室");
        EventTime e3 = new EventTime("ショートホームルーム", 2019, 9, 19, 8, -1);
        EventTime e4 = new EventTime("秋季全校集会", 2019, 9, 19, 9, 10);

        System.out.println("--- e1 ---"); e1.show();
        System.out.println("--- e2 ---"); e2.show();
        System.out.println("--- e3 ---"); e3.show();
        System.out.println("--- e4 ---"); e4.show();
    }
}
```

[実行結果]

```
--- e1 ---
(2019 年 7 月 15 日) 海の日
--- e2 ---
(2019 年 8 月 2 日) インターンシップガイダンス
    場所: 大教室
--- e3 ---
(2019 年 9 月 19 日 8 時) ショートホームルーム
--- e4 ---
(2019 年 9 月 19 日 9 時～10 時) 秋季全校集会
```

2 前問で示したクラス EventBase 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 の呼び出しを繰り返し処理をする main をもったクラス Pd1term2 をファイル「12_event.java」に作成しなさい。

ただし、配列には以下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、適切なクラスを各要素に用いて作ること。表中の「—」は情報を持たないことを表している。

添字	title	year	month	day	place	start	end
0	前期期末試験開始	2019	7	26	—	—	—
1	期末試験 2 日目	2019	7	29	—	9	—1
2	電気回路 II	2019	8	2	—	9	10
3	プログラム設計	2019	8	2	情報工学演習室	—	—
4	前期期末試験終了	2019	8	2	—	—	—

実行結果は以下のようになる。

[実行結果]

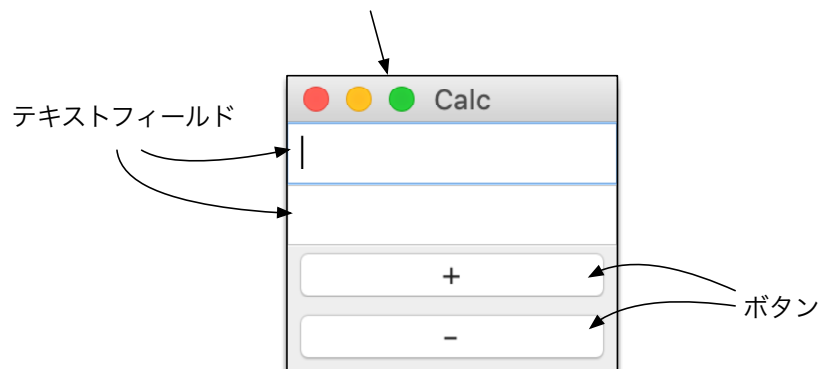
```

--- events[0].show() ---
(2019 年 7 月 26 日) 前期期末試験開始
--- events[1].show() ---
(2019 年 7 月 29 日 9 時) 期末試験 2 日目
--- events[2].show() ---
(2019 年 8 月 2 日 9 時～10 時) 電気回路 II
--- events[3].show() ---
(2019 年 8 月 2 日) プログラム設計
      場所：情報工学演習室
--- events[4].show() ---
(2019 年 8 月 2 日) 前期期末試験終了

```

3 「入力された 2 つの整数の加算/減算をする」アプリケーションを作ることを考える。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プログラムをファイル「34_calc.java」に作成しなさい。

フレーム（このフレームはGridLayoutで配置する）

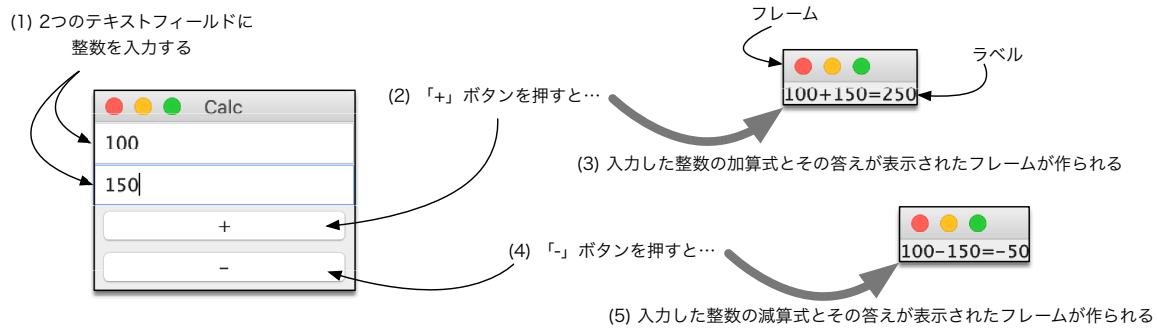


4 前問で作成したアプリケーションに、次に示すような「加算または減算の計算をして、計算式が表示されたフレームが作られる」動作をファイル「34_calc.java」に作成しなさい。前問で作ったクラスに追記して作成して構わない。

2 つのテキストフィールドに整数を入力して、ボタンを押すと以下のような動作をする

- 「+」 ボタンを押すと、2 つの整数の加算結果を表示したフレームを作る（動作の詳細は下図参照）
- 「-」 ボタンを押すと、2 つの整数の減算結果を表示したフレームを作る（動作の詳細は下図参照）

※ String 型と int 型の相互変換には第 14 週の内容を参照



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、Web ブラウザやネットワークを介した情報の参照、自分の PC を使用する、など)
- * 試験中 (開始 5 分後～開始 60 分後) は、演習室外へのネットワークアクセスは遮断される
- * GitHub への提出のためのコマンドに限ってネットワーク利用が可能 (それ以外は不正行為とする)

答案の提出

1. **提出する全ての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**
2. 端末内で、以下のコマンドで課題を提出

```
$ git add -A
```

```
$ git commit -m "前期期末提出"
```

```
$ git push origin master
```

(push が成功すると「done」が含まれたメッセージが表示される)

3. 提出が完了しているかを確認したい人は声をかけて下さい。(その場で教員側の画面で確認します)

前期期末試験 模範解答 (平均 90.7 点)

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

■問 1, 問 2

```

class EventBase {
    protected String title; //イベントの内容
    protected int year;      //イベントの年月日
    protected int month;
    protected int day;
    public EventBase(String t, int y, int m, int d) {
        title = t;
        year = y; month = m; day = d;
    }
    public void show() {
        System.out.printf("(%d 年 %d 月 %d 日) %s\n",
                           year, month, day, title);
    }
}

class EventPlace extends EventBase {
    private String place;
    public EventPlace(String t, int y, int m, int d,
                      String p) {
        super(t, y, m, d);
        place = p;
    }
    public void show() {
        super.show();
        System.out.println("    場所: " + place);
    }
}

class EventTime extends EventBase {
    private int start;
    private int end;    //-1 はなしを意味する
    public EventTime(String t, int y, int m, int d,
                     int s, int e) {
        super(t, y, m, d);
        start = s;
        end = e;
    }
    public void show() {
        if(end== -1) {
            System.out.printf("(%d 年 %d 月 %d 日 %d
時) %s\n",
                             year, month, day, start, title);
        } else {
            System.out.printf("(%d 年 %d 月 %d 日 %d 時
~%d 時) %s\n",
                             year, month, day, start, end, title);
        }
    }
}

class Pd1term1 {
    public static void main(String[] args) {
        EventBase e1 = new EventBase("海の日", 2019, 7, 15);
        EventPlace e2 = new EventPlace("インターンシ
ップガイダンス", 2019, 8, 2, "大教室");
        EventTime e3 = new EventTime("シヨートホーム
ルーム", 2019, 9, 19, 8, -1);
        EventTime e4 = new EventTime("秋 季 全 校 集
会", 2019, 9, 19, 9, 10);

        System.out.println("--- e1 ---"); e1.show();
        System.out.println("--- e2 ---"); e2.show();
        System.out.println("--- e3 ---"); e3.show();
        System.out.println("--- e4 ---"); e4.show();
    }
}

class Pd1term2 {
    public static void main(String[] args) {
        EventBase[] events = new EventBase[5];
        events[0] = new EventBase("前期期末試験開始",
                                   2019, 7, 26);
        events[1] = new EventTime("期末試験 2 日目",
                                   2019, 7, 29, 9, -1);
        events[2] = new EventTime("電気回路 II",
                                   2019, 8, 2, 9, 10);
        events[3] = new EventPlace("プログラム設計",
                                   2019, 8, 2, "情報工学演習室");
        events[4] = new EventBase("前期期末試験終了",
                                   2019, 8, 2);
        for(int i=0; i<events.length; i++) {
            System.out.printf(
                "--- events[%d].show() ---\n", i);
            events[i].show();
        }
    }
}

```

■問 3, 4

```

import java.awt.*;
import java.awt.event.*;

class Calc implements ActionListener {
    private Frame f1;
    private Button pl, mi;
    private TextField num1, num2;
    public Calc() {
        f1 = new Frame("Calc");
        num1 = new TextField();
        num2 = new TextField();
        pl = new Button("+");
        mi = new Button("-");
        f1.setLayout(new GridLayout(4, 1));
        f1.add(num1);
        f1.add(num2);
        f1.add(pl);
        f1.add(mi);
        pl.addActionListener(this);
        mi.addActionListener(this);
        f1.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
        f1.pack();
        f1.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        Frame ansf = new Frame();
        Label ans1 = new Label();
        String s1, s2, sa = "";
        int n1, n2;
        s1 = num1.getText();
        s2 = num2.getText();
        n1 = Integer.parseInt(s1);
        n2 = Integer.parseInt(s2);
        if(e.getSource()==pl) {
            sa = String.valueOf(n1) + '+'
                + String.valueOf(n2) + '=';
            sa = sa + String.valueOf(n1+n2);
        }
        if(e.getSource()==mi) {
            sa = String.valueOf(n1) + '-'
                + String.valueOf(n2) + '=';
            sa = sa + String.valueOf(n1-n2);
        }
        ans1.setText(sa);
        ansf.add(ans1);
        ansf.pack();
        ansf.setVisible(true);
    }

    public static void main(String[] args) {
        Calc obj = new Calc();
    }
}

```