

プログラム設計 前期中間試験

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」
\$ cd prog4d-(ユーザ名)
\$./myconf
3. 今回の定期試験用のフォルダをコピーする
\$ cd ~/prog4d-(ユーザ名)
\$ cp -r /usr/local/common/kogai/pd/test1mid . (←ここにピリオド)
\$ cd test1mid (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)
1_array.java 2_reminder.java 3_overload.java 4_add.java 5_hasa.java
4. 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 int 型の配列 array1 に格納された整数を、int 型の配列 array2 に逆順で格納し、出力するプログラムをファイル「1_array.java」に作成しなさい。プログラムは、以下に示すコードに不足部分を追加すると完成させることができる。

```
class Pd1mid1 {
    public static void main(String[] args) {
        int[] array1 = {80, 60, 22, 50, 75};
        int[] array2 = new int[array1.length];

        //-----ここに逆順で array2 に格納する処理を入れる-----

        for(int i=0; i<array2.length; i++) {
            System.out.printf("array2[%d]: %d ", i, array2[i]);
        }
    }
}
```

この実行結果は以下のようになる。

[実行結果]
array2[0]: 75 array2[1]: 50 array2[2]: 22 array2[3]: 60 array2[4]: 80

2 次のような「リマインダ（何かを思い出させたり、気付かせたりするメモや助言など。）」を表すクラスをファイル「2_reminder.java」に定義しなさい。

- クラス名：Reminder
- フィールド：
 - String 型の name（リマインダの内容）
 - int 型の priority（リマインダの優先順位で「5（高）～1（低）、0（なし）」とする）
- メソッド：

```
public Reminder()
    //クラス Reminder のコンストラクタ
    //フィールド name に"My task"、priority に 0 を代入する
```

```
public void setNP(String n, int p)
    //フィールド name に仮引数 n、priority に仮引数 p を代入する
```

```
public void show()
    //フィールド name と priority の値を出力する（出力の様子は以下に示す実行結果を参照）
```

main の処理とその実行結果は以下のようになる。

```
[main を持ったクラス]
class Pd1mid2 {
    public static void main(String[] args) {
        Reminder r1;
        r1 = new Reminder();
        r1.show();
        r1.setNP("Clean up my table", 3);
        r1.show();
    }
}
```

[実行結果]

```
(0) My task
(3) Clean up my table
```

3 前問で作ったクラス Reminderに、次のようなメソッド（とコンストラクタ）をオーバーロードしなさい。ファイル「3 overload.java」に、**前問で作ったクラス Reminder をコピーして**から解答をすること。

```
public Reminder(String n)
    //フィールド name に仮引数 n、priority に 3 を代入する
```

```
public void setNP(int p)
    //フィールド priority に仮引数 p を代入する（フィールド name は変更されない）
```

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 Pd1mid3 {
    public static void main(String[] args) {
        Reminder r2;
        r2 = new Reminder("Set up my coffee brewer");
        r2.show();
        r2.setNP(1);
        r2.show();
    }
}
```

[実行結果]

```
(3) Set up my coffee brewer
(1) Set up my coffee brewer
```

4 前問で作ったクラス Reminderに、「仮引数で渡されたクラス Reminder のインスタンスの情報を、自身のフィールドに結合する」メソッド add を追加しなさい。ファイル「4.add.java」に、**前問で作ったクラス Reminder をコピーして**から解答をすること。

```
public void add(Reminder r)
    //フィールド name 同士は "->" を間に挟んで文字列結合する
    //フィールド priority は以下のように決まる
    // ・自身の priority が 0 の場合、仮引数 r の priority となる
    // ・仮引数 r の priority が 0 の場合、自身の priority となる
    // ・それ以外の場合、自身の priority と仮引数 r の priority の平均値となる（ただし int 型のままの計算で構わない）
```

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 Pd1mid4 {
    public static void main(String[] args) {
        Reminder r3, r4, r5, r6;
        r3 = new Reminder(); r4 = new Reminder(); r5 = new Reminder(); r6 = new Reminder();
        r3.setNP("Buy milk", 0);
        r4.setNP("Buy coffee beans", 1);
        r5.setNP("Grind beans", 0);
        r6.setNP("Make a latte" , 5);
        r3.add(r4); r3.show();    //自身の priority が 0 の場合の add
        r3.add(r5); r3.show();    //仮引数 r の priority が 0 の場合の add
        r3.add(r6); r3.show();    //上記以外の場合の add
    }
}

[実行結果]
(1) Buy milk -> Buy coffee beans
(1) Buy milk -> Buy coffee beans -> Grind beans
(3) Buy milk -> Buy coffee beans -> Grind beans -> Make a latte
```

5 前問で作ったクラス Reminderに、「場所」を表すクラス Place 型のフィールド place を追加して、以下のメソッドを place に対応するように変更し、新規にメソッド setPlace を追加しなさい。（変更内容については、以下のコメントを参照）ファイル「5.hasa.java」に、**前問で作ったクラス Reminder をコピーして**から解答をすること。

なお、クラス Place は同ファイルに定義済みである。

```
public Reminder()
public Reminder(String n)
    //フィールド place に対してクラス Place のインスタンスを作成する（引数なしコンストラクタでよい）
```

```
public void show()
    //フィールド name, priority の出力の後に、「@」を出力して、place のメソッド show で場所の情報を出力する
```

```
public void setPlace(Place p)
    //フィールド place が仮引数 p で渡されたインスタンスを参照する
```

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 Pd1mid5 {
    public static void main(String[] args) {
        Reminder r7, r8;
        r7 = new Reminder(); r8 = new Reminder("Charge my wallet");
        Place p1;
        p1 = new Place();
        p1.setNC("Coffee Shop", "Tsukuba");
        r7.show(); r8.show();          //初期状態の出力
        r7.setPlace(p1); r8.setPlace(p1);
        r7.setNP("Buy coffee cups", 5);
        r7.show(); r8.show();          //情報更新後の出力
    }
}

```

[実行結果]

```

(0) My task
  @ somewhere(Mito)
(3) Charge my wallet
  @ somewhere(Mito)
(5) Buy coffee cups
  @ Coffee Shop(Tsukuba)
(3) Charge my wallet
  @ Coffee Shop(Tsukuba)

```

問題はここまで (各 20 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 "前期中間提出"
$ 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）

前期中間試験 模範解答 (平均 93.3 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 ~ 5 各 20 点

■問 1

```
class Pd1mid1 {
    public static void main(String[] args) {
        int[] array1 = {80, 60, 22, 50, 75};
        int[] array2 = new int[array1.length];

        for(int i=0; i<array1.length; i++) {
            array2[i] = array1[array1.length-i-1];
        }

        for(int i=0; i<array2.length; i++) {
            System.out.printf("array2[%d]: %d ",
                               i, array2[i]);
        }
    }
}
```

■問 2

```
class Reminder {
    private String name;
    private int priority;
    public Reminder() {
        name = "My task";
        priority = 0;
    }
    public void setNP(String n, int p) {
        name = n;
        priority = p;
    }
    public void show() {
        System.out.printf("(%d) %s\n",
                           priority, name);
    }
}

class Pd1mid2 {
    public static void main(String[] args) {
        Reminder r1;
        r1 = new Reminder();
        r1.show();
        r1.setNP("Clean up my table", 3);
        r1.show();
    }
}
```

■問 3

```
class Reminder {
    private String name;
    private int priority;
    public Reminder() {
        name = "My task";
        priority = 0;
    }
}
```

```
public Reminder(String n) {
    name = n;
    priority = 3;
}
public void setNP(String n, int p) {
    name = n;
    priority = p;
}
public void setNP(int p) {
    priority = p;
}
public void show() {
    System.out.printf("(%d) %s\n",
                       priority, name);
}
}
```

```
class Pd1mid3 {
    public static void main(String[] args) {
        Reminder r2;
        r2 = new Reminder("Set up my coffee brewer");
        r2.show();
        r2.setNP(1);
        r2.show();
    }
}
```

■問 4

```
class Reminder {
    private String name;
    private int priority;
    public Reminder() {
        name = "My task";
        priority = 0;
    }
    public Reminder(String n) {
        name = n;
        priority = 3;
    }
    public void setNP(String n, int p) {
        name = n;
        priority = p;
    }
    public void setNP(int p) {
        priority = p;
    }
    public void show() {
        System.out.printf("(%d) %s\n",
                           priority, name);
    }
    public void add(Reminder r) {
        this.name = this.name + " -> " + r.name;
        if(this.priority == 0) {
            this.priority = r.priority;
        } else if(r.priority == 0) {

```

```

        } else {
            this.priority =
                (this.priority + r.priority)/2;
        }
    }
}

class Pd1mid4 {
    public static void main(String[] args) {
        Reminder r3, r4, r5, r6;
        r3 = new Reminder(); r4 = new Reminder();
        r5 = new Reminder(); r6 = new Reminder();
        r3.setNP("Buy milk", 0);
        r4.setNP("Buy coffee beans", 1);
        r5.setNP("Grind beans", 0);
        r6.setNP("Make a latte", 5);
        r3.add(r4); r3.show();
        r3.add(r5); r3.show();
        r3.add(r6); r3.show();
    }
}

```

■問5

```

class Reminder {
    private String name;
    private int priority;
    private Place place;
    public Reminder() {
        name = "My task";
        priority = 0;
        place = new Place();
    }
    public Reminder(String n) {
        name = n;
        priority = 3;
        place = new Place();
    }
    public void setNP(String n, int p) {
        name = n;
        priority = p;
    }
    public void setNP(int p) {
        priority = p;
    }
    public void show() {
        System.out.printf("(%d) %s\n",
                           priority, name);
    }
}

```

```

        System.out.print(" @ ");
        place.show();
    }
    public void setPlace(Place p) {
        place = p;
    }
    public void add(Reminder r) {
        this.name = this.name + " -> " + r.name;
        if(this.priority == 0) {
            this.priority = r.priority;
        } else if(r.priority == 0) {
        } else {
            this.priority =
                (this.priority + r.priority)/2;
        }
    }
}

class Place {
    private String name;
    private String city;
    public Place() {
        name = "somewhere";
        city = "Mito";
    }
    public void setNC(String n, String c) {
        name = n;
        city = c;
    }
    public void show() {
        System.out.printf("%s(%s)\n", name, city);
    }
}

class Pd1mid5 {
    public static void main(String[] args) {
        Reminder r7, r8;
        r7 = new Reminder();
        r8 = new Reminder("Charge my wallet");
        Place p1;
        p1 = new Place();
        p1.setNC("Coffee Shop", "Tsukuba");
        r7.show(); r8.show();
        r7.setPlace(p1); r8.setPlace(p1);
        r7.setNP("Buy coffee cups", 5);
        r7.show(); r8.show();
    }
}

```