オブジェクト指向プログラミング 令和２年度 後期中間試験

(2020.12.08 重村 哲至) IE5 ＿＿＿番 氏名＿＿＿＿＿＿ 模範解答

1 用語

語群から適切な言葉を記号で選びなさい．(2 点 ×10 問=20 点)

オブジェクト指向プログラミング (Object-oriented programming:OOP) は，相互に **(1)** を送り合う **(2)** の集まりとしてプログラムを構成するプログラミング技法である．

Java 言語では，オブジェクトの型を **(3)** で定義し，**(3)** から **(4)** （オブジェクト）を作成する．**(3)** はデータと **(5)** を持つ．これらのうち外部から必要とされるものだけを公開し，それ以外は内部に隠蔽する．隠蔽されたデータは **(6)** のみを通してアクセスできる．隠蔽されたデータの仕様と，オブジェクトの外部から見たデータの仕様が分離され，データ抽象がなされる．このように，オブジェクトの外部からは決められた窓口のみを通してアクセスし，内部の構造を隠蔽することを **(7)** と呼ぶ．

既存のクラスを引き継いで新しい派生クラスを作ることを **(8)** と呼ぶ．ここで既存のクラスのことを **(9)** クラス，派生クラスのことを **(10)** クラスと呼ぶ．

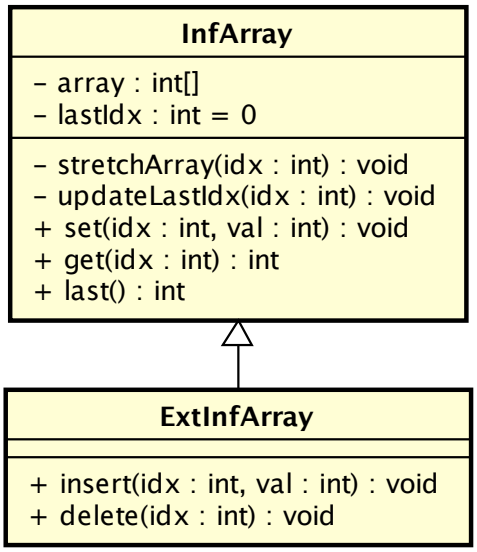
語群：

（あ）インスタンス，（い）オブジェクト，（う）カプセル化，（え）クラス，（お）サブ，（か）スーパー，
（き）セッター・ゲッター，（く）メッセージ，（け）メソッド，（こ）継承

(1)	（く）	(2)	（い）	(3)	（え）	(4)	（あ）	(5)	（け）	(6)	（き）
(7)	（う）	(8)	（こ）	(9)	（か）	(10)	（お）				

2 無限長配列クラス

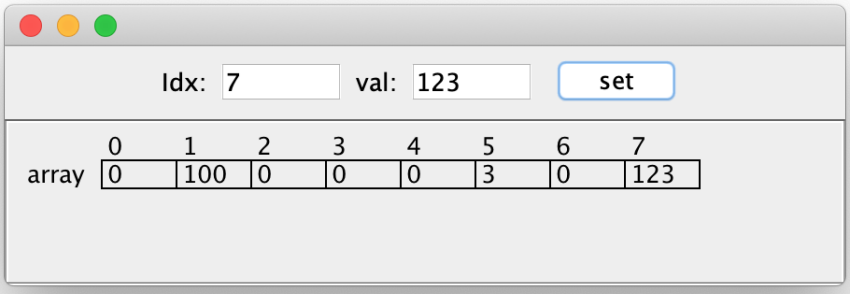
1. 付録 1 の InfArray クラスと ExtInfArray クラスをクラス図で示しなさい．（2 つのクラスとクラス間の関連を描くこと．クラスのデータ，メソッドは全て書くこと．）(20 点)

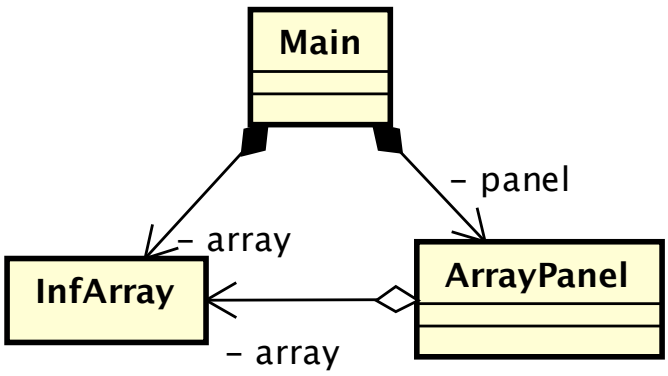


オブジェクト指向プログラミング 令和２年度 後期中間試験

(2020.12.08 重村 哲至) IE5 _____番 氏名_____ 模範解答

2. 付録 1 の Main クラス，ArrayPanel クラス，InfArray クラスの関連をクラス図で示しなさい．なお，クラスは「クラス名」のみで表現し，関連の種類やロール名はできるだけ正確に描くこと．なお，このプログラムの実行例は，次のスクリーンショットの通りです．(20 点)

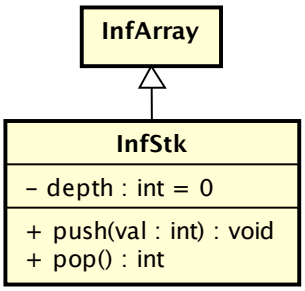




オブジェクト指向プログラミング 令和２年度 後期中間試験

(2020.12.08 重村 哲至) IE5 ＿＿＿番 氏名＿＿＿＿＿＿ 模範解答

3. InfArray クラスを継承し，スタックを表現する InfStk クラスを Java で記述しなさい．InfStk クラスは現在のスタックの深さを記録する depth 変数と，スタックに int 型の値を書き込む push() メソッド，スタックから値を取り出す pop() メソッドを持ちます．スタックが空の時，pop() はゼロを返すものとします．(20 点)

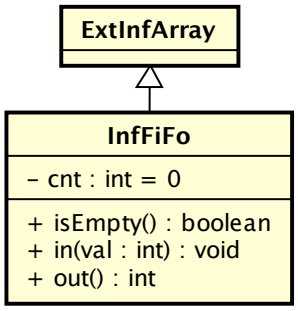



```
public class InfStk extends InfArray {
  private int depth = 0;
  public void push(int val) {
    set(depth, val);
    depth++;
  }
  public int pop() {
    if (depth<1) return 0;
    depth--;
    int val = get(depth);
    return val;
  }
}
```

オブジェクト指向プログラミング 令和２年度 後期中間試験

(2020.12.08 重村 哲至) IE5 ＿＿＿番 氏名＿＿＿＿＿＿模範解答＿＿＿＿＿＿

4. ExtInfArray クラスを継承し，FIFO を表現する InfFifo クラスを Java で記述しなさい．InfFifo クラスは現在のデータ件数を記録する cnt 変数と，FIFO が空なら true を返す isEmpty() メソッド，FIFO に int 型の値を書き込む in() メソッド，FIFO から値を取り出す out() メソッドを持ちます．FIFO が空の時，out() はゼロを返すものとします．InfFifo クラスの使用例は付録２の Main3 クラスに示します．なお，InfFifo クラスは無駄に配列を大きくしないように作成する必要があります．(20 点)




```
public class InfFifo extends ExtInfArray {
  private int cnt = 0;
  public boolean isEmpty() {
    return cnt==0;
  }
  public void in(int val) {
    set(cnt, val);
    cnt++;
  }
  public int out() {
    if (cnt<1) return 0;
    cnt--;
    int val = get(0);
    delete(0);
    return val;
  }
}
```

付録１

リスト 1: InfArray クラス

```
1  // ゼロで初期化された無限長配列クラス
2  // 有効範囲より後ろはゼロのデータが無限に続いていると考える
3  public class InfArray {
4    private int[] array = new int[10];      // 実際の配列
5    private int lastIdx = 0;                // 配列の有効範囲
6
7    // 配列を延長する（呼び出し回数を減らすために10%大きく延長する）
8    private void stretchArray(int idx) {
9      int[] b = new int[idx+idx/10];        // 少し長い新しい配列
10     for (int i=0; i<array.length; i++) { // 新しい配列に
11       b[i] = array[i];                    //    要素をコピー
12     }
13     array = b;                            // 配列を置き換える
14   }
15
16   // idx番目の要素が変更された時，配列の有効範囲を更新する
17   private void updateLastIdx(int idx) {
18     if (idx>lastIdx && array[idx]!=0) {  // 有効範囲が広くなる
19       lastIdx = idx;
20     } else if (idx==lastIdx) {           // 有効範囲が狭くなる
21       while (idx>0 && array[idx]==0)
22         idx--;
23       lastIdx = idx;
24     }
25   }
26
27   // idx番の要素にvalを書き込む
28   public void set(int idx, int val) {
29     if (array.length<=idx) {             // 配列の範囲外
30       if (val==0) return;                // 範囲外のゼロは無視
31       stretchArray(idx);                 // 配列を伸ばす
32     }
33     array[idx] = val;                    // 配列に値を書き込む
34     updateLastIdx(idx);
35   }
36
37   // idx番の配列要素を読み出す
38   public int get(int idx) {              // 配列要素を読み出す
39     if (array.length<=idx) return 0;     // 配列外はゼロを返す
40     return array[idx];                   // 配列要素を返す
41   }
42
43   // ゼロ以外の値を持った最後の要素の番号を返す（空の場合は0を返す）
44   public int last() {                    // 配列の最終データ位置
45     return lastIdx;
46   }
47 }
```

リスト 2: Main クラス

```
1  import java.awt.EventQueue;
2  // ... 省略 ...
3  public class Main {
4      private JFrame frame;
5      private JTextField idxField;
6      private JTextField valField;
7      private InfArray array = new InfArray();
8      private JScrollPane scrollPane;
9      private ArrayPanel panel;
10     public static void main(String[] args) {
11         // ... 省略 ...
12     }
13     public Main() {
14         initialize();
15         panel.setArray(array);
16     }
17     private void initialize() {
18         frame = new JFrame();
19         frame.setBounds(100, 100, 450, 150);
20         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         scrollPane = new JScrollPane();
22         frame.getContentPane().add(scrollPane, BorderLayout.CENTER);
23         panel = new ArrayPanel();
24         scrollPane.setViewportView(panel);
25         // ... 省略 ...
26         idxField = new JTextField();
27         // ... 省略 ...
28         valField = new JTextField();
29         // ... 省略 ...
30         JButton setBtn = new JButton("set");
31         setBtn.addActionListener(new ActionListener() {
32             public void actionPerformed(ActionEvent e) {
33                 int idx = Integer.parseInt(idxField.getText());
34                 int val = Integer.parseInt(valField.getText());
35                 array.set(idx, val);
36                 panel.setArray(array);              // なぜか, repaint()では
37                 scrollPane.setViewportView(panel);  // スクロールに切替わらない
38             }
39         });
40         ctlPanel.add(setBtn);
41     }
42 }
```

リスト 3: ArrayPanel クラス

```
1  import java.awt.Dimension;
2  import java.awt.Graphics;
3  import javax.swing.JPanel;
4
5  @SuppressWarnings("serial")
6  public class ArrayPanel extends JPanel {
7      private final int am = 10;                    // "array"の左マージン
8      private final int w = 40;                     // 箱の幅
9      private final int h = 15;                     // 箱の高さ
10     private final int lm = 50;                    // 配列の左マージン
11     private final int rm = 20;                    // 配列の右マージン
12     private final int tm = 20;                    // 配列の上マージン
13     private final int bm = 20;                    // 配列の下マージン
14     private final int d = 3;                      // 文字列のマージン
15     private InfArray array;
16
17     // 配列の 1 要素を描く
18     private void draw(Graphics g, int idx, int val) {
19         int x0 = w*idx + lm;                      // 原点 x
20         int y0 = tm;                              // 原点 y
21         g.drawRect(x0, y0, w, h);
22         g.drawString(idx+"", x0+d, y0-d);
23         g.drawString(val+"", x0+d, y0+h-d);
24     }
25
26     @Override
27     protected void paintComponent(Graphics g) {
28         super.paintComponent(g);
29         if (array==null) return;
30         g.drawString("array", am, tm+h-d);
31         for (int i=0; i<=array.last(); i++) {
32             draw(g, i, array.get(i));
33         }
34     }
35
36     public void setArray(InfArray array) {
37         this.array = array;
38         int aw = (array.last()+1)*w;  // 配列部分の幅
39         Dimension di = new Dimension(lm+aw+rm, tm+h+bm);
40         setPreferredSize(di);
41     }
42 }
```

リスト 4: ExtInfArray クラス

```
1  // 挿入・削除ができる無限長配列クラス
2  public class ExtInfArray extends InfArray{
3    public void insert(int idx, int val) { // idx番目にvalを挿入
4      for (int i=last(); i>=idx; i--) {    // データを後ろに
5        set(i+1, get(i));                  //  ずらす
6      }
7      set(idx ,val);                       // valを書き込む
8    }
9
10   public void delete(int idx) {          // idx番目の要素を削除
11     for (int i=idx; i<last(); i++) {     // データを前に
12       set(i, get(i+1));                  //   ずらす
13     }
14     set(last(), 0);                      // 最後のデータを消す
15   }
16 }
```

付録２

リスト 5: Main3 クラス

```
1  public class Main3 {
2    private InfFifo f = new InfFifo();
3    private void test(String[] args) {
4      for (int i=0; i<args.length; i++) {
5        f.in(Integer.parseInt(args[i]));
6      }
7      while (!f.isEmpty()) {
8        System.out.printf("%d ",f.out());
9      }
10     System.out.printf("\n");
11   }
12
13   public static void main(String[] args) {
14     Main3 m = new Main3();
15     m.test(args);
16   }
17 }
```