

オブジェクト指向プログラミング 令和4年度 後期中間試験

(2022.12.08 重村 哲至) IE5

番 氏名

模範解答

1 用語

語群から最適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 (2) は設計図にあたる (3) から new 演算子を用いて作成する。 (1) を送信することは、対象になる (2) の (4) を呼び出すことによって行う。

(3) のデータにあたるものは (5) と呼ばれる。 (5) の (6) は private に設定し、 (3) の外部から直接に操作されないようにする。これにより (2) の利用者は決められた窓口だけを利用して (3) を操作するしかなくなる。 (2) の利用者が (7) に依存したプログラミングをすることを防止できる。このようにして (7) を隠蔽することを (8) と言う。

クラス間に「本体と部分」のような関係がある関連は (9) である。 (9) の中でも「本体と部分が同時に生成され同時に消滅する」ような強い結びつきがある場合は (10) である。

語群:

(あ) オブジェクト, (い) カプセル化, (う) クラス, (え) メッセージ, (お) メソッド,
(か) 可視性, (き) 合成集約, (く) 集約, (け) 属性, (こ) 内部構造

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

2 添字の範囲を自由に設定できる配列クラス

付録1にソースリストを掲載した Java プログラムについて答えなさい。なお、付録2にクラス図のサンプルを掲載するので参考にすること。

- 以下に IntArray クラスのクラス図を描きなさい。クラス図はできる限り省略などをしないで正確に描くこと。(10点)

IntArray
- array : int[]
+ <<create>> IntArray(size : int) + set(idx : int, val : int) : void + get(idx : int) : int + length() : int

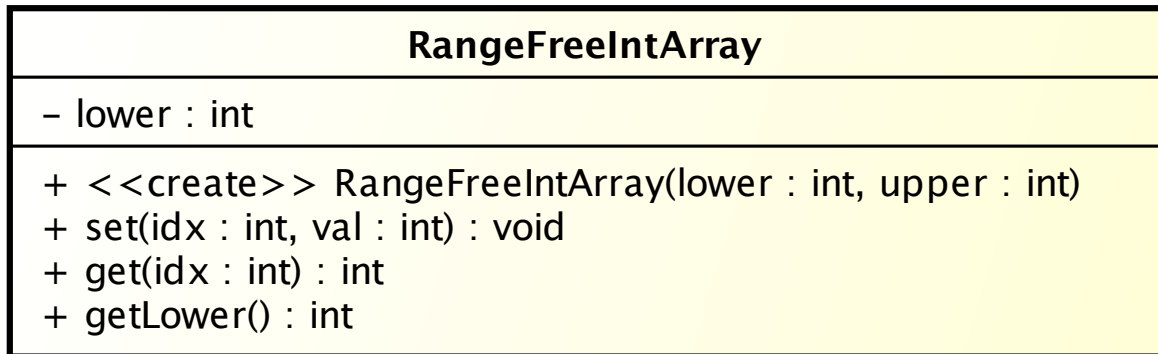
オブジェクト指向プログラミング 令和4年度 後期中間試験

(2022.12.08 重村 哲至) IE5

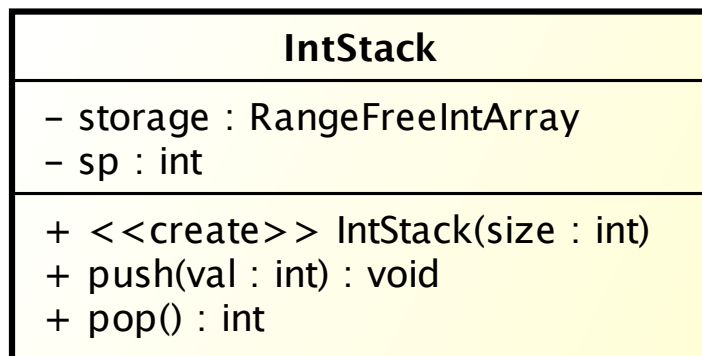
____ 番 氏名

模範解答

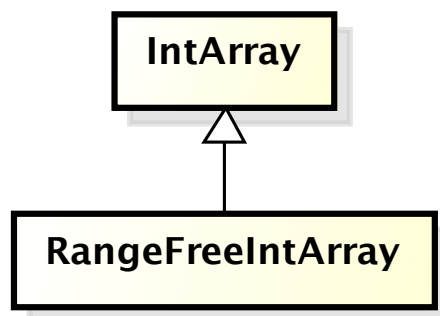
2. 以下に RangeFreeIntArray クラスのクラス図を描きなさい。クラス図はできる限り省略などをしないで正確に描くこと。なお、普通のメソッドとオーバーライドをするメソッドの表現方法の区別は UML のクラス図には無いので、普通のメソッドと同じ描き方をすればよい。(10 点)



3. 以下に IntStack クラスのクラス図を描きなさい。クラス図はできる限り省略などをしないで正確に描くこと。また、関連のロール名で表現する方が適切なものも属性として記入すること。(10 点)



4. 以下に IntArray, RangeFreeIntArray クラスとの関係を表現するクラス図を描きなさい。2つのクラスの図は属性と操作を省略しクラス名だけの長方形で表現すること。(10 点)



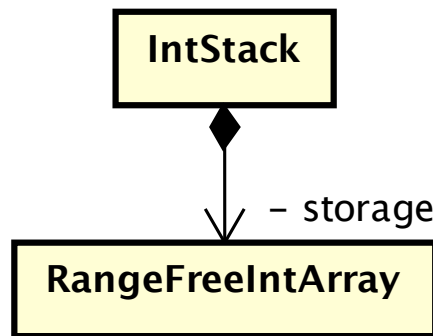
オブジェクト指向プログラミング 令和4年度 後期中間試験

(2022.12.08 重村 哲至) IE5

____ 番 氏名

模範解答

5. 以下に IntStack, RangeFreeIntArray クラスの関連を表現するクラス図を描きなさい。2つのクラスの図は属性と操作を省略しクラス名だけの長方形で表現すること。関連には、「誘導可能性 (矢印等のこと)」、「可視性 (+, - など) を含むロール名」、「集約など」を書き込むこと。なお、上の問題で IntStack クラスの属性として表現したものもロール名を用いて再度、表現すること。(10 点)



6. RangeFreeIntArray クラスは、インスタンス作成時に添え字の最小値と最大値を指定できる整数配列クラスです。例えば次のプログラムのように使えます。

```
int r;
RangeFreeIntArray a = new RangeFreeIntArray(-1, +1); // 要素数3の配列
a.set(-1, 0);    // a[-1] = 0; の代用
a.set( 0, 1);    // a[ 0] = 1; の代用
a.set( 1, 2);    // a[ 1] = 2; の代用
r=a.get(-1);     // r = a[-1]; の代用
r=a.get( 0);     // r = a[ 0]; の代用
r=a.get( 1);     // r = a[ 1]; の代用
```

- (a) RangeFreeIntArray クラスの###(1)###に記入すべきプログラムを以下に書きなさい。(6 点)

```
super.set(idx-lower, val)
```

- (b) RangeFreeIntArray クラスの###(2)###に記入すべきプログラムを以下に書きなさい。(6 点)

```
return super.get(idx-lower)
```

オブジェクト指向プログラミング 令和4年度 後期中間試験

(2022.12.08 重村 哲至) IE5

____ 番 氏名

模範解答

7. 次のような TestMain クラスをコンパイルする際に (1)～(12) の行で、エラーになる行に「×」、ならない行に「○」を次の解答欄に記入しなさい。 (1点×12問=12点)

```
public class TestMain {  
    public static void main(String[] args) {  
        int r;  
        // RangeFreeIntArray について  
        RangeFreeIntArray a = new RangeFreeIntArray(-3, +3);  
        r=a.array[0];           //(1)  
        a.set(-1, 0);           //(2)  
        r=a.get(-1);            //(3)  
        r=a.lower;              //(4)  
        r=a.getLower();         //(5)  
        r=a.length();           //(6)  
  
        // IntStack について  
        IntStack s = new IntStack(10);  
        r=s.sp;                  //(7)  
        s.push(3);               //(8)  
        r=s.pop();               //(9)  
        r=s.set(0, 1);           //(10)  
        r=s.storage.get(0);      //(11)  
        r=s.storage.length();    //(12)  
    }  
}
```

(1)	×	(2)	○	(3)	○	(4)	×	(5)	○	(6)	○
(7)	×	(8)	○	(9)	○	(10)	×	(11)	×	(12)	×

8. Main クラスからこのプログラを実行することができます。「\$ java Main」でプログラムを実行したときの出力を以下に書きなさい。 (6点)

3
2
1

付録 1

リスト 1: IntArray クラス

```
// IntArray.java
public class IntArray {
    private int[] array;
    public IntArray(int size) {
        array = new int[size];
    }
    public void set(int idx, int val) { array[idx] = val; }
    public int get(int idx) { return array[idx]; }
    public int length() { return array.length; }
}
```

リスト 2: RangeFreeIntArray クラス

```
// RangeFreeIntArray.java
public class RangeFreeIntArray extends IntArray {
    private int lower;
    public RangeFreeIntArray(int lower, int upper) {
        super(upper-lower+1);
        this.lower = lower;
    }
    @Override
    public void set(int idx, int val) { ###(1)###; }
    @Override
    public int get(int idx) { ###(2)###; }
    public int getLower() { return lower; }
}
```

リスト 3: IntStack クラス

```
// IntStack.java
public class IntStack {
    private RangeFreeIntArray storage;
    private int sp; // Stack Pointer
    public IntStack(int size) {
        storage = new RangeFreeIntArray(-size, -1);
        sp = 0;
    }
    public void push(int val) {
        sp--;
        storage.set(sp, val);
    }
    public int pop() {
        int n = storage.get(sp);
        sp++;
        return n;
    }
}
```

リスト 4: Main クラス

```
// Main.java
public class Main {
    private IntStack stack = new IntStack(3);
    private Main() {
        stack.push(1);
        stack.push(2);
        stack.push(3);
        System.out.println(stack.pop());
        System.out.println(stack.pop());
        System.out.println(stack.pop());
    }
    static public void main(String[] args) {
        new Main();
    }
}
```

付録2

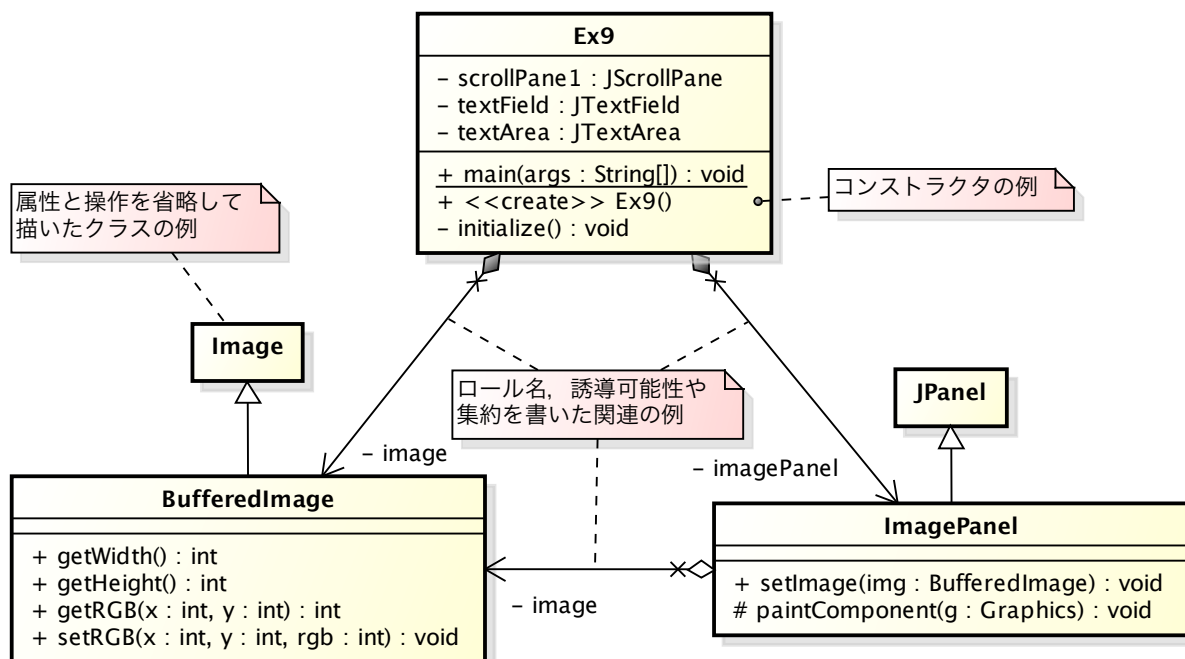


図 1: クラス図のサンプル