PPL FINAL SOLUTION

SPRING 2014

Phạm Tuấn Anh

2c12

Hanoi University

Part I: multiple choice questions

This part has 3 questions. Each question is worth **5 marks**.

Question 1

Which of the followings is the output of this code?

```
int x = 0;

int y = 0;


while (y < 10) {

  x = x + 2 * y;

  y++;

}


int z = x/y ;

System.out.printf("%d, %d, %d%n", x, y, z);
```

a. 100, 10, 10

b. 90, 10, 9

c. 70, 10, 3

d. run-time error

Đáp án : 1 b

Question 2

Which of the followings is the output of this code?

```
for (int i = 0; i < 5; i++) {
  for (int j = 0; j < 20; j++) {
    if (j % 5 == i) {
      System.out.printf("%3d", j);
    }
  }
  System.out.prinln();
}
```

a.

0 1 2 3 4

5 6 7 8 9

10 11 12 13 14

15 16 17 18 19

b.

0 1 2 3 4 5

6 7 8 9 10

11 12 13 14 15

16 17 18 19 20

c.

0 5 10 15

1 6 11 16

2 7 12 17

3 8 13 18

4 9 14 19

d. run-time error

Đáp án 2 c

Question 3

Which of the followings is the output of this code?

```
String string = "To be or not to be";


for (char c = 'a'; c <= 'z'; c++) {
  for (int i = 0; i < string.length(); i++) {
    if (c == string.charAt(i)) {
      System.out.printf("%-2s", c);
      break;
    }
  }
}
```

a. b e n o r t

b. T r o n e b t

c. T o b e r n t

d. run-time error

Đáp án 3a

Part II :task-based questions

This part has 5 questions.

Question 4 [15 marks]

Specify and implement a procedure named `frequencyIf`, which computes and returns an array containing the numbers of the occurrences of the elements of an integer array that satisfy a given condition. The condition must be a string of the form `op val` where `op` is one of the followings operators: <=, >=, |, `val` is an integer value. The operator | means "divisible by". Frequencies of the elements that do not satisfy the condition are set to 0. For example:

`frequencyIf([1,1,2,3,2,4,5], "| 2") = [0,0,2,0,2,1,0]`, because only 2 and 4 are divisible by 2 and there are two occurrences of 2 and one occurrence of 4. Similarly,

`frequencyIf([1,1,2,3,2,4,5], ">= 2") = [0,0,2,1,2,1,1]` and

`frequencyIf([1,1,2,3,2,4,5], "<= 2") = [2,2,2,0,2,0,0].`

Đáp án :

```
public static int[] frequentlyIf(int[] a){
    int k = 1;
    int[] b = new int[a.length];
    for( int i=0 ;i < a.length ; i++ ){
       for( int j = 0; j < a.length; j++){
          if(i==j) continue;
```

```
            else{
                if(a[i]==a[j]){
                    k+=1;
                    b[i] = k;
                }
                else if(a[i]!=a[j]){
                    b[i] = k;
                }
            }
        }
    }
    return b;
    }
```

Question 5 [15 marks]

Given below is the specification and Java code of a procedure. Add suitable mid-conditions to the code and use these to determine the loop invariant. Write the invariant in the block comment that is marked in the code.

```
/**
  * @requires a != null /\ a.length > 0
  * @effects return n in a: (for all y. y in a -> y <= n),
  *  e.g. method1([2, 1, 3]) = 3
  */
```

```
public static int method1(int[] a){
  int n = a[0];

  /*
   * Loop invariant:
   *
   */
  int x;
  for (int i = 0; i < a.length; i++){
    x = a[i];
    if (x > n){
      n = x;
    }
  }
  return n;
}
```

Đáp án :

```
/**
 * @requires a != null /\ a.length > 0
 * @effects return n in a: (for all y. y in a -> y <= n),
 *  e.g. method1([2, 1, 3]) = 3
```

```
 */
public static int method1(int[] a){
    int n = a[0];

    /*
     * Loop invariant: *
     */
    int x;
    for (int i = 0; i < a.length; i++){
    // 0 <= i < a.length /\ ( i >= 1) -> (i += 1 /\ n = a[i] | a[i] > n)
        x = a[i];
          // x = a[i];
        if (x > n){
    // 0 <= i < a.length /\ x = a[i] /\
    // (i >= 1) -> (i += 1 /\ n = a[i] | a[i] > n)
            n = x;
        }
        // n = a[k] | a[k] >= a[i] | 0 <= k <= i
    }
    // i = a.length - 1 /\ n = a[k] | a[k] >= a[i] | 0 <= k <= i
```

```
    return n;
}
```

Question 6 [35 marks]

Given below is a listing of partial design specification of a class named `IntegerArray`. Given also that the statement `import` at the top imports the annotation class `DomainConstraint` for use in this class. Answer the following questions to complete the design and code of this class. The `TODO` block comments in the listing mark the regions where you need to write your answers.

a) [5 marks] Define the class attributes.

b) [10 marks] Specify and code the necessary constructor operation(s).

c) [15 marks] Specify and code the other necessary operation(s).

d) [5 marks] Code the extra operation given in the listing.

```
import userlib.DomainConstraint;


/**
  * @overview IntegerArray is a fixed sequence of integer numbers.
  *
  * @attributes
  *  array      Integer[]
  *  len        Integer
  *
```

```
  * @object
  *  A typical IntegerArray is [x1,x2, ..., xn], where xi is integer
for all 1 <= i <= n.
  *
  * @abstract_properties
  *  mutable(array)=true /\ optional(array)=false /\
  *  mutable(len)=false /\ optional(len)=false /\ min(len)=1 /\
  *  length(array) = len
  *
  * @rep_invariant
  *  array != null /\
  *  len >= 1 /\
  *  array.length = len
  */
public class IntegerArray{
  /**
    * TODO: ATTRIBUTES
    */


  /**
    * TODO: SPECIFY AND CODE NECESSARY CONSTRUCTOR(S)
    */
```

```
  /**
    * TODO: SPECIFY AND CODE OTHER NECESSARY OPERATION(S)
    */

  /**
    * TODO: CODE THIS EXTRA OPERATION
    */

  /**
    * @effects
    *  return the sum of the array elements, i.e. result =
    *        array[0] + ... + array[array.length - 1]
    * @requires this satisfies the rep invariant
    */
  public int sum()
} // end IntegerArray
```

Đáp án : bài này mình làm gộp vào 1 class code cho tiện

```
import userlib.DomainConstraint;

/**
  * @overview IntegerArray is a fixed sequence of integer numbers.
```

```
    *
    * @attributes
    *  array      |        Integer[]
    *  len        |        Integer
    *
    * @object
    *  A typical IntegerArray is [x1,x2, ..., xn], where xi is integer for all 1 <= i <= n.
    *
    * @abstract_properties
    *  mutable(array)=true /\ optional(array)=false /\
    *  mutable(len)=false /\ optional(len)=false /\ min(len)=1 /\
    *  length(array) = len
    *
    * @rep_invariant
    *  array != null /\
    *  len >= 1 /\
    *  array.length = len
    */
public class IntegerArray{

  @DomainConstraint(type="int[]",mutable=true,optional=false)
```

```
private int[] array;
@DomainConstraint(type="int",mutable=false,optional=false,min=1)
private int len;
private IntegerArray(){
    //do nothing
}
public IntegerArray(int[] array){
    this.array = array;
    len = array.length;
}
public void setArray(int[] array){
    array = new IntegerArray(array);
}
public int[] getArray(){
    return array;
}
public getLength(){
    return len;
}
```

```
    /**
     * TODO: CODE THIS EXTRA OPERATION
     */


    /**
     * @effects
     *  return the sum of the array elements, i.e. result =
     *        array[0] + ... + array[array.length - 1]
     * @requires this satisfies the rep invariant
     */
    public int sum(){
        int n = 0;
        for ( int i = 0; i < len; i++ ){
                n += array[i];
        }
        return n;
    }


} // end IntegerArray
```

Question 7 [10 marks]

Answer the following question about the program given below:

a) What is the state of the program in the stack and heap memories when the code is run?

b) What is the console output after running the program?

```
public class Program1{

  public static void main(String[] args){

    int[] x = {10, 20, 30};

    int[] y = {2, 4, 6, 8};


    int i, j;

    for (i = x.length-1; i >= 0; i--){

      for (j = 0; j < y.length; j++){

        x[i] = x[i] + y[j];

      }

      System.out.println(x[i]);

    }

  }

}
```

Đáp án :

a.

x 10 20 30

y 2 4 6 8

i 0 x[0] = 10 + 2 + 4 + 6 + 8 = 30
j 0 1 2 3
i 1 x[1] = 20 + 2 + 4 + 6 + 8 = 40
j 0 1 2 3
i 2 x[2] = 30 + 2 + 4 + 6 + 8 = 50
j 0 1 2 3

b.
Console output :
30
40
50
(có xuống dòng)

Question 8 (10 marks)

Given below is the partial definition of a class named `Module`. Briefly discuss the design issues concerning the three attributes of this class and the solutions for them.

```
import java.util.Vector;


/**
  * @overview Module is a course module in a university program.
  */
public class Module{
  public int id;
  public String name;
  private Vector students;
```

```
  // ... code omitted ... /
}
```

Đáp án :

```
import java.util.Vector;
// should import domainconstraint
/**
 * @overview Module is a course module in a university program.
// lack @attribute, @object, @abstract_properties, @repvariant,
 */
public class Module{
  // domain constraint lacked
    public int id;
    // should be private instead of public
    // should be auto generated ID started with ID = 1
  // domain constraint lacked
    public String name;
    //should be private instead of public
  // domain constraint lacked
```

```
    private Vector students;


  // … code omitted … /

}
```