CENG 1004
Introduction to Object Oriented Programming

Spring 2016

WEEK 14

Overview

Question - 1

```
public class Question1 {

   public static void test(int x){
        if (x > 5){
               System.out.print("A");
        } else if (x==5) {
               System.out.print("B");
        } else {
               System.out.print("C");
        }
   }
   public static void main(String[] args){
        test(4);
        test(5);
        test(6);
   }
}
```

Question - 2

```
public class Question2 {
    static int triType (int a, int b, int c){
          if (a > b) { int t = a; a = b; b = t; }
          if (a > c) { int t = a; a = c; c = t; }
          if (b > c) { int t = b; b = c; c = t; }
          if (a + b <= c){
                    return 0;
          }else{
                    if ((a==b) && (b==c)){
                              return 2;
                    }else if (a*a + b*b == c*c){
                              return 4;
                    } else if ((a==b) || (b==c)){
                              return 3;
                    }
                    return 1;
          }
    }
    public static void main(String args[]){
          System.out.print(triType(5,4,3));
          System.out.print(triType(5,5,3));
          System.out.print(triType(5,2,2));
          System.out.print(triType(5,5,5));
    }
}
```

Question - 3

```
public class Question3 {
   private static void doSomething(int x, int y, int z) {
        x++;
        y++;
        z++;
        System.out.println("x= " + x + ", y= " +y + ", z= " +z );

   }
   public static void main(String args[]) {
        int x=1, y=2, z=3;
        doSomething(x,y,z);
        System.out.println("x= " + x + ", y= " +y + ", z= " +z );
   }
}
```

Question - 4

```
public class Question4 {

   public static void main(String args[]) {
       for(int x=1;x<=5;x++){
            for(int y=1;y<=x;y++)
                System.out.print(x+",");
       }
   }
}
```

Question - 5

```
public class Question5 {
    private static void doSomething(int[] intArray) {
          int n = intArray.length;
          int temp = 0;
          for (int i = 0; i < n; i++)
                    for (int j = 1; j < (n - i); j++)
                              if (intArray[j - 1] > intArray[j]) {
                                        temp = intArray[j - 1];
                                        intArray[j - 1] = intArray[j];
                                        intArray[j] = temp;
                              }
    }
    public static void main(String args[]) {
          int[] items = new int[]{5,90,35,45,150,3};
          doSomething(items);
          for (int i = 0; i<items.length; i++){
                    System.out.print(items[i]);
                    if (i!= items.length-1)
                              System.out.print(",");
          }
    }
}
```

Question - 6

```
public class Question6 {
   private static void doSomething(int[] intArray) {
        intArray = new int[] { 3, 5, 35, 45, 90, 150 };
   }
   public static void main(String args[]) {
        int[] items = new int[] { 5, 90, 35, 45, 150, 3 };
        doSomething(items);

        for (int i = 0; i < items.length; i++) {
               System.out.print(items[i]);
               if (i != items.length - 1)
                      System.out.print(",");
        }
   }
}
```

Question - 7

- **Write a method that calculates the sum of the numbers in a given two dimesional array of integers**

Question - 7

- **Write a method that calculates the sum of the numbers in a given two dimesional array of integers**

```
public static int sum(int [][] multiArray){
    int sum = 0;
    for (int i = 0; i< multiArray.length; i ++){
        for (int j =0; j< multiArray[i].length; j++){
            sum+= multiArray[i][j];
        }
    }
    return sum;
}
```

Question - 8

- **Write a recursion method that prints the Nth Fibonacci number**

Question - 8

- **Write a recursion method that prints the Nth Fibonacci number**

```
public static int fibonacci(int number){
   if(number == 1 || number == 2){
        return 1;
   }
   return fibonacci(number-1) + fibonacci(number -2);
}
```

Question - 9

- **Write a class that implements the following interface to provide FIFO Queue behaviour. You are not allowed to use array structure and any of Collection classes defined in java.util.**

```
interface Queue{
   //Inserts the specified element into this queue
   void add(Object obj);

   //Retrieves and removes the head of this queue.
   Object remove();

   //Returns the size of this queue.
   int size();
}
```

Question - 10

- **Modify both the interface and your classes to support generics**

Question - 11

- **Modify your Queue implementation by adding a constructor which sets the maximum capaity with the given parameter.**

- **Create checked exceptions for your and throw these exceptions when a remove is called when queue is empty and when an add is called when Queue is full.**

Question - 12

```
public class Question12 {
   static int a = 0;
   int b = 0;
   public void incrementCounter() {
        a++;
        b++;
   }
   public static void main(String args[]) {
        Question12 obj1 = new Question12();
        Question12 obj2 = new Question12();
        obj1.incrementCounter();
        obj2.incrementCounter();
        obj2.incrementCounter();
        System.out.println("Obj1.a: " + obj1.a);
        System.out.println("Obj1.b: " + obj1.b);
        System.out.println("Obj2.a: " + obj2.a);
        System.out.println("Obj2.b: " + obj2.b);
   }
}
```

Question -13

	private	no modifier	protected	public
Inside class				
Same Package Class				
Same Package Subclass				
Other Package Class				
Other Package Subclass				

Question - 14

```
public class Question14 {
   int a;
   int b;
   public Question14(int a, int b) {
       a = a;
       this.b = b;
   }
   public static void main(String[] args) {
       Question14 s = new Question14(37, 47);
       System.out.println("a = " + s.a);
       System.out.println("b = " + s.b);
   }
}
```

Question - 15

```
public class Question15 {
   public static void main(String[] args) {
        ChildClass c = new ChildClass();
        c.showID();
   }
}
class ParentClass {
   int id = 1;

   void showID() {
        System.out.println(id);
   }
}
class ChildClass extends ParentClass {
   int id = 2;
}
```

Question - 16

```
public class Question16 extends A {
    void f() {
        System.out.println("Question16.f");
    }
    void g() {
        f();
        super.f();
    }
    public static void main(String[] args) {
        Question16 o = new Question16();
        o.g();
    }
}
class A {
    void f() {
        System.out.println("A.f");
    }
}
```

Question - 17

```
class Base {
    static void f() {
          System.out.println("Base.f");
    }
    void g() {
          System.out.println("Base.g");
    }
}
public class Sub extends Base {
    static void f() {
          System.out.println("Sub.f");
    }
    void g() {
          System.out.println("Sub.g");
    }
    public static void main(String[] args) {
          Base ref = new Sub();
          ref.f();
          ref.g();
    }
}
```

Question - 18

Write a cylinder class which extends the given Circle class

```
// The superclass Circle
public class Circle {
   private double radius;
   // Constructor
   public Circle(double radius) {
      this.radius = radius;
   }
   // Getter
   public double getRadius() {
      return this.radius;
   }
   // Return the area of this circle
   public double getArea() {
      return radius * radius * Math.PI;
   }
   // Describe itself
   public String toString() {
      return "Circle[radius=" + radius + "]";
   }
}
```

Question - 19

- Write a Drawing class which contains rectangles and circles.
  - The class should have a method to add rectangles and circles
  - The class should have method that claculates the total area of shapes it contain
- Your drawing class should be extensible so that whenever a new type of shape is introduced, you do not need to modify the drawing class

Question - 20

```
public class Question20 implements Comparator<Integer> {
   public int compare(Integer o1, Integer o2) {
         int result = (o1 % 10) - (o2 % 10);
         if (result == 0 && !o1.equals(o2))
                  return o1 - o2;
         return result;

   }
   public static void main(String[] args) {
         Set<Integer> map = new TreeSet<>(new Question20());
         map.add(12);
         map.add(21);
         map.add(30);
         map.add(41);

         System.out.println(map);
   }
}
```

Question - 21

```
public class Question21 {
   public static void main(String[] args) {
        Set<Integer> set = new LinkedHashSet<>();
        set.add(5);
        set.add(4);
        set.add(3);
        set.add(4);
        set.add(2);
        System.out.println(set);
   }
}
```

Question - 22

```
import java.util.ArrayList;
import java.util.Collection;
public class Question22 {
   public static void main(String[] args) {
        Collection<Integer> list = new ArrayList<>();
        list.add(5);
        list.add(7);
        list.add(2);
        list.add(10);
        for (int i=0; i<list.size(); i++){
               list.remove(i);
        }
        System.out.println(list);
   }
}
```

Question - 23

```
public class Question23 {
   public static void main(String[] args) {
        try{
            int x = 0;
            int y = 5 / x;
        }catch (Exception e){
            System.out.println("Exception");
        } catch (ArithmeticException ae){
            System.out.println(" Arithmetic Exception");
        }
        System.out.println("finished");       }
}
```

Question - 24

```
public class Question24{
   public static void main(String[] args) {
         try {
                  badMethod();
                  System.out.print("A");
         } catch (RuntimeException ex) {
                  System.out.print("B");
         } catch (Exception ex1) {
                  System.out.print("C");
         } finally {
                  System.out.print("D");
         }
         System.out.print("E");

   }
   public static void badMethod() {
          int x = 0;
          int y = 5 / x;
   }
```

Question - 25

- **Below the contents of tab seperated scores.txt file is given. Write a program that reads this file and generates sortedscores.txt in which students and their scores are sorted by descending score order.**

```
Student ID  Score
140709003   28
140709004   21
140709005   43
140709006   74
140709007   44
140709008   64
140709009   56
140709011   74
```

Question - 25

```
public class Student {
   private String id;
   private int score;
   public Student(String id, int score) {
         super();
         this.id = id;
         this.score = score;
   }
   public String getId() {
         return id;
   }
   public int getScore() {
         return score;
   }
}
```

Question - 25

```
public class StudentComparator implements Comparator<Student> {

   @Override
   public int compare(Student o1, Student o2) {
        if ((o1.getScore() > o2.getScore())){
                return -1;
        }else if(o2.getScore() > o1.getScore()){
                return 1;
        }else{
                return o1.getId().compareTo(o2.getId());
        }
   }
}
```

Question - 25

```
public class Question25 {
    public static void main(String[] args) throws IOException {
          Set<Student> students = new TreeSet<>(new StudentComparator());
          try (Scanner scanner = new Scanner(new BufferedReader(new
    FileReader("scores.txt")));
                              Formatter formatter = new Formatter(new
    BufferedWriter(new FileWriter("out.txt")))) {
                    String[] columns = scanner.nextLine().split("\t");
                    scanner.useDelimiter("\t|\r\n");
                    while (scanner.hasNext()) {
                              Student student = new Student(scanner.next(),
    scanner.nextInt());
                              System.out.println(student.getId());
                              students.add(student);
                    }
                    formatter.format("%s\t%s%n", columns[0], columns[1]);
                    for (Student s : students) {
                              formatter.format("%s\t%d%n", s.getId(), s.getScore());
                    }
          }
    }
}
```

Question - 26

- Make your Queue implementation for Question 9 thread safe.

Question - 27

- Write a Producer Thread which generates random numbers and put them into Queue.
  - generate 20 numbers

- Write a Consumer Thread which removes numbers from the Queue and prints them.