Exam

Explanations

Carefully formulate your answers, code, and images

Formulate your answers precise and to the point.

Code shall be written so that it is easy to follow and understand. In some situations suitable comments can contribute to understanding. Small syntactical errors can be tolerated. If some parts of code cannot be exactly produced, it is possible that well-formed pseudocode may provide a solution. Do not write more code than necessary; if just a method is requested there is no need to create a whole class. All program code is to be written in Java.

When an array (vector) or an object is drawn, it must be clearly visible by which reference the array or object is referred to, and what data is located inside it. When an array or object contains a reference, the resource that is referred to (an object or array) shall be drawn. All references shall have relevant labels.

Points and grading

In total: 42 points
For grade E at least: 21 points
For grade D at least: 25 points
For grade C at least: 29 points
For grade B at least: 33 points
For grade A at least: 37 points
.

Tasks

Task 1 (3 points + 3points)

```
int[][]    b = { {0, 1, 2},
                 {3, 2, 1},
                 {2, 3, 4} };

int[]    u = new int[b.length];
int    m = 0;
for (int i = 0; i < u.length; i++)
{
    m = b[i][0];
    for (int j = 1; j < b[i].length; j++)
        if (b[i][j] > m)
            m = b[i][j];
    u[i] = m;
}

int[][]    v = new int[2][];
v[0] = b[2];
v[1] = b[1];
```

a) Draw the array referred to by reference `u`.

b) Draw the array referred to by reference `v`.

Task 2 (3 points + 3 points + 3 points)

The class `Triangle` represents a triangle:

```
class Triangle
```

```
{
    // the side lengths of the triangle
    private double    a;
    private double    b;
    private double    c;

    public Triangle (double a, double b, double c)
    {
        this.a = a;
        this.b = b;
        this.c = c;
    }

    // perimeter returns the circumference of the triangle
    public double perimeter ()
    {
        return  a + b + c;
    }

    // area returns the area of the triangle
    public double area ()
    {
        double    s = (a + b + c) / 2;
        double    ar = Math.sqrt (s * (s - a) * (s - b) * (s - c));

        return ar;
    }
}
```

a) A static method, `maxTriangle`, accepts an array of triangles (objects of type `Triangle`), and returns that triangle in the array which has the greatest circumference. Create this method.

b) A static method, `totalArea`, accepts an array of triangles (objects of type `Triangle`), and returns the total area of these triangles. Create this method.

c) Create an array of triangles (objects of type `Triangle`), and call the methods `maxTriangle` and `totalArea` in some way.

Task 3 (3 points + 3 points + 3 points)

The class `DigitHandler` manages a sequence of digits in different ways:

```
class DigitHandler
{
    private byte[]    digits;

    public DigitHandler (String digits)
    {
        this.digits = new byte[digits.length ()];
        for (int pos = 0; pos < this.digits.length; pos++)
            this.digits[pos] = (byte) (digits.charAt(pos) - 48);
     }

    public String toString ()
    {
        StringBuilder    sb = new StringBuilder ("");
        for (byte digit : digits)
            sb.append (digit);

        return sb.toString ();
    }

    public void circularShiftLeft ()
    // code is missing here


    public void circularShiftRight ()
    // code is missing here
```

```
}
```

An instance of class `DigitHandler` is created and used like this:

```
DigitHandler    handler = new DigitHandler ("12345");  (1)
System.out.println (handler);
System.out.println ();

handler.circularShiftLeft ();
System.out.println (handler);
handler.circularShiftLeft ();
System.out.println (handler);
handler.circularShiftRight ();
System.out.println (handler);
handler.circularShiftRight ();
System.out.println (handler);
```

When this code fragment is executed, the following output is generated:

```
12345

23451
34512
23451
12345
```

a) What does the object referred to by reference `handler` look like when statement (1) has been executed? Draw the object.

b) Implement the method `circularShiftLeft`.

c) Implement the method `circularShiftRight`.

Task 4 (3 points + 3 points + 3 points)

The class `Word` represents a word. Characters in the word are stored in a sequence of linked nodes.

```
class Word
{
    private class Node
    {
        public char    c;
        public Node    next;

        public Node (char c)
        {
            this.c = c;
            this.next = null;
        }
    }


    // the first node in the node sequence
    private Node    first = null;


    // Word creates a word from an array of characters
    // code is missing here


    public String toString ()
    {
        StringBuilder    sb = new StringBuilder ("[");
        Node    node = first;
        if (node != null)
        {
            while (node.next != null)
```

```
            {
                sb.append (node.c);
                node = node.next;
            }
            sb.append (node.c);
        }
        sb.append ("]");

        return sb.toString ();
    }

    public void combine (Word w)
    {
        if (w.first != null)
        {
            Node    n1 = this.first;
            Node    n2 = w.first;
            if (this.first == null)
            {
                this.first = new Node (n2.c);
                n1 = this.first;
                n2 = n2.next;
             }

             while (n1.next != null)
                n1 = n1.next;

             while (n2 != null)
             {
                n1.next = new Node (n2.c);
                n1 = n1.next;
                n2 = n2.next;
             }
        }
     }
}
```

The class `Word` is used like this:

```
char[]   v = {'L', 'i', 'f', 'e'};
Word    w = new Word (v);
System.out.println (w);
```

When this code fragment is executed, the following output is generated:

```
[Life]
```

a) Create the constructor in class `Word`.

b) Draw the object referred to by reference `w`.

c) Which output is generated if class `Word` is used like this?

```
char[]   v1 = {'L', 'i', 'g', 'h', 't'};
Word    w1 = new Word (v1);
char[]   v2 = {'L', 'o', 'v', 'e'};
Word    w2 = new Word (v2);
w1.combine (w2);
System.out.println (w1);
```

Task 5 (3 points + 3 points + 3 points)

An algorithm sorts a sequence of elements that can be compared with the operator *less* (<). The algorithm is used below to sort a sequence of integers:

```
public static void sort (int[] sequence)
{
```

```
    int    lastPos = sequence.length - 1;
    int    i = 0;
    for (int pos = 0; pos < lastPos; pos++)
    {
        for (int p = pos + 1; p <= lastPos; p++)
            if (sequence[p] < sequence[pos])
            {
                i = sequence[pos];
                sequence[pos] = sequence[p];
                sequence[p] = i;
            }

        System.out.println (java.util.Arrays.toString (sequence));
     }
}
```

a) The method `sort` is called like this:

```
int[]   seq = {5, 4, 3, 2, 1};
System.out.println (java.util.Arrays.toString (seq));
System.out.println ();

sort (seq);
System.out.println ();
System.out.println (java.util.Arrays.toString (seq));
```

Which output is generated when this code fragment is executed?

b) Specify the preconditions and postconditions of the algorithm. Represent steps in the algorithm in the form of pseudocode.

c) Let n denote the number of elements being sorted.

Determine the time complexity of the algorithm in terms of the number of element comparisons. Categorize the corresponding complexity function: to which Θ–set does it belong?

Also determine the worst case time complexity of the algorithm in terms of the number of element exchanges. To which Θ–set does the corresponding complexity function belong?