Exec01.java

```
class Exec01 {
    public static void main(String[] args) {
        int a = 1, b = 4;
        System.out.println(a | b);

        int x = 0x03, y = 0x05;
        System.out.println(x ^ y);

        int c = -8;
        System.out.println(c >> 1);

        int sum = 0;
        for(int i = 0; i < 10; i++) {
            sum += i++;
        }
        System.out.println(sum);

        int m = 1, n = 2;
        if(m < n++ | m++ < n)
            m += n++;
        System.out.println(m);

        boolean flag = false;
        if(flag = false) {
            System.out.println("TRUE");
        } else {
            System.out.println("FALSE");
        }
    }
}
```

Exec02.java

```
class Exec02 {
    public static void main(String[] args) {
        String[] words = new String[4];
        words[0] = "spring";
        words[1] = "summer";
        words[2] = "autumn";
        words[3] = "winter";
        int[][] n = {{2, 4}, {3, 6, 9}, {5, 10}};

        for(String word : words) {
            System.out.println(word);
        }

        for (int [] array : n) {
            for(int i : array) {
                System.out.print(i + " ");
            }
            System.out.println();
        }

        System.out.println(words.length);
        System.out.println(n.length);
    }
}
```

Exec03.java

```
class Student {
    String name;
    int age;

    void show() {
        System.out.println("名前  " + name);
        System.out.println("年令  " + age);
    }
}

class Exec03 {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student();
        Student s3 = new Student();

        s1.name = "鈴木";
        s1.age = 18;
        s2.name = "田中";
        s2.age = 23;
        s3.name = "吉田";
        s3.age = 31;

        s1.show();
        s2.show();
        s3.show();
    }
}
```

Exec04.java

```
class Student {
    String name;
    int age;

    Student() {
        this.name = "なし";
        this.age = 99;
    }
    Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    void show() {
        System.out.println("名前  " + name);
        System.out.println("年令  " + age);
    }
}

class Exec04 {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student("斉藤", 28);

        s1.show();
        s2.show();
    }
}
```

Exec05.java

```
class Student {
    String name;
    int age;
    static int count;

    Student() {
        this.name = "なし";
        this.age = 99;
    }
    Student(String name, int age) {
        this.name = name;
        this.age = age;
        count++;
    }

    void show() {
        System.out.println("名前　" + name);
        System.out.println("年令　" + age);
    }
}

class Exec05 {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student("斉藤", 28);
        Student s3 = new Student("岡田", 17);

        s1.show();
        s2.show();
        s3.show();
        System.out.println("人数　" + Student.count);
    }
}
```

Exec06.java

```
class Student {
    String name;
    int age;

    Student() {
        this.name = "なし";
        this.age = 99;
    }
    Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    void show() {
        System.out.println("名前  " + name);
        System.out.println("年令  " + age);
    }
}

class Exec06 {
    public static void main(String[] args) {
        Student s1;
        Student s2 = new Student("斉藤", 28);

        s1 = s2;
        s1.show();
        s2.show();

        String str1 = "Hello";
        String str2 = new String("Hello");

        if(str1 == str2)
            System.out.println("同じオブジェクト");
        else
            System.out.println("違うオブジェクト");
        if(str1.equals(str2))
            System.out.println("同じ文字列");
        else
            System.out.println("違う文字列");
    }
}
```

Exec07.java

```
class Student {
    private String name;
    private int age;

    Student() {
        this.name = "なし";
        this.age = 99;
    }
    Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    void show() {
        System.out.println("名前  " + name);
        System.out.println("年令  " + age);
    }
}

class Exec07 {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student("田中", 23);

        s1.name = "鈴木";
        s1.age = 18;

        s1.show();
        s2.show();
    }
}
```

Exec08.java

```
class Student {
    private String name;
    private int age;

    Student() {
        this.name = "なし";
        this.age = 99;
    }
    Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    void set(String name) {
        this.name = name;
    }
    void set(int age) {
        this.age = age;
    }
    void show() {
        System.out.println("名前  " + name);
        System.out.println("年令  " + age);
    }

}

class Exec08 {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student("斉藤", 28);

        s1.set("鈴木");
        s1.set(18);
        s1.show();
        s2.show();
    }
}
```

Exec09.java

```
class Exec09 {
    public static void main(String[] args) {
        String str = new String("Hello");

        System.out.println("文字数　" + str.length());
        System.out.println("文字位置　" + str.indexOf("e"));
        System.out.println("文字列の比較　" + str.equals("hello"));
        System.out.println("大文字に変換　" + str.toUpperCase());
        System.out.println("小文字に変換　" + str.toLowerCase());
        System.out.println("文字列の比較　" + str.equals("hello"));
        str = str.toLowerCase();
        System.out.println("文字列の比較　" + str.equals("hello"));

        Integer n = new Integer(100);

        System.out.println("16進表示　$" + n.toHexString(n));
    }
}
```

Exec10.java

```
class Student {
    private String name;
    private int age;

    Student() {
        this.name = "なし";
        this.age = 99;
    }
    Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    void show() {
        System.out.println("名前  " + name);
        System.out.println("年令  " + age);
    }
}

class ExStudent extends Student {
    private String country;

    ExStudent(String name, int age, String country) {
        super(name, age);
        this.country = country;
    }

    void show() {
        super.show();
        System.out.println("国籍  " + country);
    }
}

class Exec10 {
    public static void main(String[] args) {
        Student s1 = new Student("田中", 23);
        ExStudent s2 = new ExStudent("渡辺", 25, "日本");

        s1.show();
        s2.show();
    }
}
```

Exec11.java

```
abstract class Figure {
    protected double a;

    Figure(double a) {
        this.a = a;
    }

    public abstract double Area();
}

class Square extends Figure {
    Square(double a) {
        super(a);
    }

    public double Area() {
        return a * a;
    }
}

class Circle extends Figure {
    Circle(double a) {
        super(a);
    }

    public double Area() {
        return Math.PI * a * a;
    }
}

class Exec11 {
    public static void main(String[] args) {
        Figure s = new Square(2.0);
        Figure c = new Circle(2.0);

        System.out.println("正方形の面積は" + s.Area());
        System.out.println("円の面積は" + c.Area());
    }
}
```

Exec12.java

```
interface Geometry {
    final int n = 4;
    double Area();
    double Perimeter();
}

class Square implements Geometry {
    private double a;

    Square(double a) {
        this.a = a;
    }

    public double Area() {
        return a * a;
    }
    public double Perimeter() {
        return n * a;
    }
}

class Exec12 {
    public static void main(String[] args) {
        Geometry s = new Square(2.0);

        System.out.println("正方形の面積は" + s.Area());
        System.out.println("正方形の周長は" + s.Perimeter());
    }
}
```

Exec13. java

```
class Exec13 {
    enum Color {red, blue, green};
    Color dressColor, hairColor;

    public static void main(String[] args) {
        Exec13 obj = new Exec13();
        obj.dressColor = Color.red;
        obj.hairColor = Color.blue;
        System.out.println(obj.dressColor + " " + obj.hairColor);

        switch(obj.hairColor) {
            case red:
                System.out.println("red");
                break;
            case blue:
                System.out.println("blue");
                break;
            case green:
                System.out.println("green");
                break;
        }
    }
}
```

Exec14. java

```
import java.util.*;

class Exec14 {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>();

        list.add("10");
        list.add("20");
        list.add("30");
        for(int i = 0; i < list.size(); i++) {
            System.out.println(list.get(i));
        }

        LinkedList<String> lklist = new LinkedList<String>();

        lklist.add("X");
        lklist.addLast("Y");
        lklist.addFirst("Z");
        for(String s : lklist) {
            System.out.println(s);
        }

        HashMap<String, Integer> goods = new HashMap<String, Integer>();

        goods.put("A", 34);
        goods.put("B", 17);
        goods.put("A", 28);
        for(String str: goods.keySet()) {
            System.out.println(str);
        }
    }
}
```

Exec15. java

```
class Exec15 {
    public static void main(String[] args) {
        int x = 1, y = 0;

        try {
            System.out.println(x / y);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error A");
        } catch (ArithmeticException e) {
            System.out.println("Error B");
        } finally {
            System.out.println("Always");
        }

        String str = "e123";

        try {
            double d = Double.parseDouble(str);
            System.out.println(d);
        } catch (NumberFormatException e) {
            System.out.println(2);
        } catch (Exception e) {
            System.out.println(3);
        }
    }
}
```

Exec16.java

```
interface Calc {
    int VOL_A = 10;
    void calc( int val1, int val2 );
}

class TestCase1 implements Calc {
    int z;
    String MyExp;

    TestCase1() {
        System.out.println("TestCase1 のコンストラクタ");
    }

    public void calc(int x, int y) {
        this.z = x * y;
    }
}

class TestCase2 extends TestCase1 {
    int z;

    public void calc(int x, int y) {
        this.z = x + y;
        super.calc(x, y);
        System.out.println(super.z + "," + this.z);
    }
}

class Exec16 {
    public static void main(String[] args) {
        TestCase2 tc = new TestCase2();

        tc.calc(2, 3);
    }
}
```

Exec17.java

```
class Exec17 {
    public static String[][] salesList = {
        {"A025", "330"}, {"E074", "430"}, {"F011", "1203"}, {"R035", "521"},
        {"R074", "857"}, {"N041", "237"}, {"K005", "130"}
    };
    public static void main(String[] args) {
        Record rList = new Record( salesList );
        rList.sortData();
        rList.showData();
    }
}

class Record {
    Staff[] staffList;
    Record(String[][] salesList) {
        staffList = new Staff[salesList.length];
        int i = 0;
        for (String[] sales : salesList) {
            staffList[i++] = new Staff( sales );
        }
    }
    void sortData() {
        int lnum = 0, rnum = staffList.length - 1, ex = 0, i = 0;
        while(rnum > lnum) {
            while(i < rnum) {
                if(staffList[i].getSales() > staffList[i + 1].getSales()) {
                    exchangeData(i);
                    ex = i;
                }
            i++;
            }
            i = rnum = ex;
            while(i > lnum) {
                if(staffList[i].getSales() < staffList[i - 1].getSales()) {
                    exchangeData(i - 1);
                    ex = i;
                }
                i--;
            }
            i = lnum = ex;
        }
    }

    void exchangeData(int i) {
        Staff wk;

        wk = staffList[i];
        staffList[i] = staffList[i + 1];
        staffList[i + 1] = wk;
    }
```

```
    void showData() {
        for(int i = staffList.length - 1; i >=0; i--) {
            System.out.print(staffList[i].getId() + "\t");
            System.out.println(staffList[i].getSales());
        }
    }
}

class Staff {
    String id;
    int sales;

    Staff(String[] salesData) {
        this.id = salesData[0];
        this.sales = Integer.parseInt(salesData[1]);
    }
    String getId() {
        return this.id;
    }
    int getSales() {
        return this.sales;
    }
}
```

Exec18.java

```
import java.lang.StringBuffer;
import java.util.ArrayList;

class Exec18 {
    public static void main(String[] args) {
        Data data = new Data(args);

        System.out.print(data.getString());
    }
}

class Data {
    private int number, cardinalNumber;

    Data(String[] args) {
        this.number = Integer.parseInt(args[0]);
        this.cardinalNumber= Integer.parseInt(args[1]);
    }

    String getString() {
        StringBuffer buff = new StringBuffer(cardinalNumber + "進数:");
        ArrayList<Integer> collection = new ArrayList<Integer>();
        int num = number;
        char digit;

        while( num > 0 ) {
            collection.add(num % cardinalNumber);
            num /= cardinalNumber;
        }
        for(int i = collection.size() - 1; i >= 0; i--) {
            if(collection.get(i) < 10) {
                digit = (char)(collection.get(i) + '0');
            } else {
                digit = (char)(collection.get(i) - 10 + 'A');
            }
            buff.append(digit);
        }
        return buff.toString();
    }
}
```

Exec19. java

```
import java.io.*;
import java.util.*;

class TextLines {
    List<String> linelist;

    public TextLines(String pathName) throws IOException {
        linelist = new ArrayList<String>();
        BufferedReader br = new BufferedReader(new FileReader(pathName));
        String line;
        while((line = br.readLine()) != null) {
            linelist.add(line);
        }
        br.close();
    }
    public List<String> getFindList(String searchWord) {
        List<String> findList = new ArrayList<String>();
        int linenum = 1;

        for(String line : linelist) {
            int findPos;
            int startPos = 0;
            while((findPos = line.indexOf(searchWord, startPos)) != -1) {
                findList.add(linenum + "(" + (findPos + 1) + "):" + line);
                startPos = findPos + searchWord.length();
            }
            linenum++;
        }
        return findList;
    }
}

class Exec19 {
    public static void main(String[] args) {
        if(args.length != 2) {
            System.out.println("Argument error");
            System.exit(0);
        }

        String searchWord = args[0];
        String pathName = args[1];
        try {
            TextLines textdata = new TextLines(pathName);
            List<String> findlist = textdata.getFindList(searchWord);
            for(String find : findlist) {
                System.out.println(find);
            }
        } catch (FileNotFoundException e) {
            System.out.println("File cannot open[" + pathName + "]");
        } catch (IOException e) {
            System.out.println("I/O error");
        }
    }
}
```

Exec19.txt

```
XAAABCDEFGH
AAABCDEFGHI
XAAAABCAAAA
XXXAAAAABCD
AA
```

Exec20.java

```
/*
<applet code = "Exec20.class" width = "400" height = "240">
</applet>
*/
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

public class Exec20 extends Applet implements ActionListener, MouseListener {
    Panel btnPnl; Button clearButton;
    int x, y;
    Dimension d;

    public void init() {
        btnPnl = new Panel();
        clearButton = new Button("CLEAR");
        clearButton.addActionListener(this);

        btnPnl.add(clearButton);
        setLayout(new BorderLayout());
        this.add("South", btnPnl);

        d = getSize();
        x = -100;
        y = -100;
        addMouseListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        if(e.getActionCommand().equals("CLEAR")) {
            Graphics g = getGraphics();
            g.setColor(getBackground());
            g.fillRect(0, 0, d.width, d.height);
            g.dispose();
        }
    }
    public void paint(Graphics p) {
        p.setColor(Color.blue);
        p.drawOval(x - 50, y - 50, 100, 100);
    }
    public void update(Graphics g) {
        paint(g);
    }
    public void mouseClicked(MouseEvent e) {
        x = e.getX();
        y = e.getY();
        repaint();
    }
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mousePressed(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
}
```