

全国商業高等学校協会主催  
情報処理検定（プログラミング部門）

J a v a

1 級

問題集

## 目 次

オブジェクト指向1	・ ・ ・ ・ 3
オブジェクト指向2	・ ・ ・ ・ 9
二分探索	・ ・ ・ ・ 14
二次元配列	・ ・ ・ ・ 16
ソート	・ ・ ・ ・ 18
ArrayList	・ ・ ・ ・ 25
解答	・ ・ ・ ・ 27

問1. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

コンストラクタを用いて表示する。

入力データ

なし

実行結果

Bunkatsu1_1Hello1
Bunkatsu1_2Hello1
bunkatsu1_2Hello2
bunkatsu1_2Hello3

```
public class Bunkatsu1_1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        System.out.println("Bunkatsu1_1Hello1");
        Bunkatsu1_2 bun2 = new ( 1 );
        bun2. ( 2 );
        bun2. ( 3 );
    }
}

public class Bunkatsu1_2 {
    ( 4 ) {
        System.out.println("Bunkatsu1_2Hello1");
    }
    public void hyouji1() {
        System.out.println("bunkatsu1_2Hello2");
    }
    public void hyouji2() {
        System.out.println("bunkatsu1_2Hello3");
    }
}
```

(1)		(3)	
(2)		(4)	

問2. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

コンストラクタを用いて表示する。

入力データ

文字
××

実行結果

Bunkatsu1_1Hello1
Bunkatsu1_2Hello1
入力してください
こんにちは
bunkatsu1_2Hello2 こんにちは
bunkatsu1_2Hello3

```
import java.util.Scanner;
public class Bunkatsu2_1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        System.out.println("Bunkatsu1_1Hello1");
        Bunkatsu2_2 bun2 = new Bunkatsu2_2();
        String ( 1 );
        Scanner sc = new Scanner(System.in);
        System.out.println("入力してください");
        nyuryoku = ( 2 );
        bun2. ( 3 );
        bun2.hyouji2();
    }
}

public class Bunkatsu2_2 {
    Bunkatsu2_2() {
        System.out.println("Bunkatsu1_2Hello1");
    }
    public void hyouji1(String nyu) {
        System.out.println("bunkatsu1_2Hello2"+nyu);
    }
    public void ( 4 ){
        System.out.println("bunkatsu1_2Hello3");
    }
}
```

処理条件

1. キーボードからデータを入力する。

(1)		(3)	
(2)		(4)	

問3. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

ゲッターセッターを用いる。

入力データ

文字
××

実行結果

GetSet1_1Hello1
セットしました
後ろの文字を入力してください
world
セットしました world

```
import java.util.Scanner;
public class GetSet1 {
    String nyuuryoku;
    public String getNyuuryoku() {
        System.out.println
            ("後ろの文字を入力してください");
        Scanner sc = new Scanner(System.in);
        String se = sc.next();
        nyuuryoku = nyuuryoku + ( 1 );
        return nyuuryoku;
    }
    public void setNyuuryoku(String nyuuryoku) {
        this.nyuuryoku = ( 2 );
        System.out.println(this.nyuuryoku);
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        String hyouji;
        GetSet1 gs1_1 = new GetSet1();
        System.out.println("GetSet1_1Hello1");
        gs1_1.( 3 )"セットしました");
        hyouji = gs1_1.getNyuuryoku();
        System.out.println( ( 4 ) );
    }
}
```

処理条件

1. キーボードからデータを入力する。

(1)		(3)	
(2)		(4)	

問4. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

ゲッターセッターを用いる。

入力データ

文字
××

実行結果

GetSet2_2Hello1
セットしました
後ろの文字を入力してください
world
セットしました world

```
public class GetSet2_1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        String ( 1 );
        GetSet2_2 gs2_2 = new GetSet2_2();
        System.out.println("GetSet2_2Hello1");
        gs2_2.setNyuuryoku("セットしました");
        hyouji = gs2_2.( 2 );
        System.out.println(hyouji);
    }
}

import java.util.Scanner;
public class GetSet2_2 {
    String nyuuryoku;
    public String getNyuuryoku() {
        System.out.println
            ("後ろの文字を入力してください");
        Scanner sc = new Scanner(System.in);
        String se = sc.next();
        nyuuryoku = nyuuryoku + se;
        return ( 3 );
    }
    public void setNyuuryoku(String nyuuryoku) {
        this.nyuuryoku = ( 4 );
    }
}
```

処理条件

1. キーボードからデータを入力する。

(1)		(3)	
(2)		(4)	

問5. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

継承を用いる。

入力データ

文字
××

実行結果

Keisyou1_3 からの表示です
Keisyou1_2 からの表示です
Keisyou1_1 からの表示です

処理条件

1. キーボードからデータを入力する。

```
public class Keisyou1_1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        Keisyou1_3 kei1_3 = new Keisyou1_3();
        kei1_3.hyouji();
        kei1_3.(1);
        System.out.println("Keisyou1_1 からの表示です");
    }
}

public class Keisyou1_2 {
    public void hyouji() {
        System.out.println("Keisyou1_2 からの表示です");
    }
}

public class Keisyou1_3 (2) Keisyou1_2{
    public void hyouji() {
        System.out.println("Keisyou1_3 からの表示です");
    }
}
```

(1)		(3)	
(2)		(4)	

問6. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

継承を用いる。

入力データ

文字
××

実行結果

いくつの階乗にしますか?
3
6

処理条件

1. キーボードからデータを入力する。

```
import java.util.Scanner;
public class Keisyou2_1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        int kotae;
        Scanner sc = new Scanner(System.in);
        Keisyou2_3 kei2_3 = new Keisyou2_3();
        System.out.println("いくつの階乗にしますか?");
        int kazu = sc.nextInt();
        kotae=kei2_3.kaijyou(kazu);
        kei2_3.hyouji( ( 1 ) );
    }
}

public class Keisyou2_2 { //スーパークラス
    void hyouji(int hkotae){
        System.out.println(hkotae);
    }
}

public class Keisyou2_3 extends Keisyou2_2{ //サブクラス
    int ( 2 );
    int kaijyou(int suuti) {
        kaikotae = 1;
        for(int n=suuti;n>=1;n--){
            kaikotae *= ( 3 );
        }
        return ( 4 );
    }
}
```

(1)		(3)	
(2)		(4)	



問7. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

抽象クラスを用いる。

入力データ

なし

実行結果

11

```
abstract class Keisan { //抽象クラス
    abstract void hyouji(int kotae); //抽象メソッド
    int n,m;
    void setsuuti(int n,int m){ //セッター
        this.n=( 1 );
        this.m=m;
    }
    int getsuutin(){ //ゲッター
        return n;
    }
    int getsuutim(){
        return ( 2 );
    }
}
class Tasizan extends Keisan{
    void hyouji(int kotae){
        System.out.println( ( 3 ) );
    }
}
class Abstract1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        Tasizan tasi = new Tasizan();
        tasi.setsuuti(5,6);
        int kotae = tasi.getsuutin()+( 4 );
        tasi.hyouji(kotae);
    }
}
```

(1)		(3)	
(2)		(4)	

問8. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

参照を用いる。

入力データ

文字
××

実行結果

```

入力してください
あんどろ
1_2
入力してください
いとう
1_2
入力してください
えんどろ
1_2
入力してください
かとう
1_2
入力してください
さとう
1_2
あんどろさん
いとうさん
えんどろさん
かとうさん
さとうさん
    
```

```

import java.util.Scanner;
public class Sansyou1_1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        Scanner sc = new Scanner(System.in);
        Sansyou1_2[] san = new Sansyou1_2[5];
        int i;
        String name;
        for (i = 0; i<=4; i++) {
            System.out.println("入力してください");
            name = sc.next();
            ( 1 ) = new Sansyou1_2(name);
        }
        String hyouji;
        for (i = 0; i<=4; i++) {
            ( 2 ) = san[i].getNamae();
            System.out.println(hyouji);
        }
    }
}

public class Sansyou1_2 {
    String namae;
    public Sansyou1_2(String namae) {
        // TODO 自動生成されたメソッド・スタブ
        System.out.println("1_2");
        ( 3 ) = namae;
    }
    public String tuika(){
        return namae+"さん";
    }
    public String getNamae() {
        return ( 4 );
    }
}
    
```

処理条件

1. キーボードからデータを入力する。

(1)		(3)	
(2)		(4)	

問9. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

オーバーロード用いる。

入力データ

なし

実行結果

2つのかけ算 2
3つのかけ算 6
4つのかけ算 24

```
public class OverLoad1_1 {
    public static void main(String[] args) {
        OverLoad1_2 ove1_2 = new OverLoad1_2();
        int kekka;
        ( 1 ) = ove1_2.jyou(1, 2); //オーバーロード
        System.out.println("2つのかけ算"+kekka);
        kekka = ove1_2.jyou(1, 2, 3);
        System.out.println("3つのかけ算"+kekka);
        kekka = ove1_2.jyou(1, 2, 3, 4);
        System.out.println("4つのかけ算"+ ( 2 ));
    }
}

public class OverLoad1_2{
    int kotae;
    public int jyou(int s1, int s2) {
        return ( 3 ) = s1 * s2;
    }
    public int jyou(int s1, int s2, int s3) {
        return kotae = s1 * s2 * s3;
    }
    public int jyou(int s1, int s2, int s3, int s4) {
        return kotae = s1 * s2 * s3 * ( 4 );
    }
}
```

(1)		(3)	
(2)		(4)	

問 10. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

オーバーライドを用いて文字を表示する。

入力データ

なし

実行結果

```
GoodMorning1_2
GoodEvening1_2
GoodMorning1_3
ここから super です
GoodMorning1_2
GoodEvening1_2
GoodEvening1_2
```

```
public class OverRide1_1 {
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        OverRide1_2 ovr1_2 = new OverRide1_2();
        OverRide1_3 ovr1_3 = new OverRide1_3();
        ovr1_2.( 1 );
        ovr1_2.hiruhyouji();
        ovr1_3.asahyouji();
        ovr1_3.hiruhyouji();
    }
}

public class OverRide1_2 {
    void asahyouji(){
        System.out.println("GoodMorning1_2");
    }
    void ( 2 ){
        System.out.println("GoodEvening1_2");
    }
}

public class OverRide1_3 extends OverRide1_2{
    void ( 3 ) { //再定義
        System.out.println("GoodMorning1_3");
        System.out.println("ここから super です");
        super.asahyouji();
        ( 4 ).hiruhyouji();
    }
}
```

( 1 )		( 3 )	
( 2 )		( 4 )	

問 11. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

ポリモルフィルズを用いて入力したデータを表示する。

入力データ

文字
××

実行結果

9
---

処理条件

1. キーボードからデータを入力する。

```
abstract class TasizanP{ //抽象クラス
    int n,m;
    abstract int kotae(); //抽象メソッド
    void setsuuti(int n, int m){
        this.n=n;
        this.m= ( 1 );
    }
    public int getN() {
        return ( 2 );
    }
    public int getM() {
        return m;
    }
    void hyouji() {
        System.out.println(kotae());
    }
}

class KeisanP extends TasizanP{
    int kotae() {
        return getN()+ ( 3 );
    }
}

public class PolymorPhism1 {
    //一つのオブジェクトやメソッドが多くの形態をもつこと
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        int n=4,m=5;
        TasizanP tas = new KeisanP();
        tas.setsuuti(n,m);
        tas.( 4 );
    }
}
```

(1)		(3)	
(2)		(4)	

問 12. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

コードを入力し表示する。

入力データ

コード
××

実行結果

コード入力：終わりは 999
103
103 は本宮
コード入力：終わりは 999

処理条件

1. キーボードからデータを入力する。
2. 結果を実行結果のようにディスプレイに表示する。
3. 二分探索を用いること。

```
import java.util.Scanner;
public class NibuTansaku {
    Scanner sc = new Scanner(System.in);
    int[] kode =
        {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
    String[] eki = {"郡山", "日和田", "五百川", "本宮",
        "二本松", "安達", "松川", "金谷川", "南福島", "福島"};
    int inkode;
    int n;
    int jogen;
    int kagen;
    int cyuuou;
    public void tansaku() {
        System.out.println
            ("コード入力：終わりは 999");
        inkode = sc.nextInt();
        int sw;
        while(inkode != 999) {
            sw =0;
            jogen=9;
            kagen=0;
            cyuuou = ( 1 );
            while(inkode != kode[cyuuou]&&
                kagen<=jogen) {
                if (kode[cyuuou]>inkode) {
                    jogen = cyuuou-1;
                }else if(kode[cyuuou]<inkode) {
                    kagen = ( 2 );
                }
                cyuuou = (jogen+kagen)/2;
            }
            if(kagen>jogen) {
                sw = ( 3 );
            }
            hyouji(sw);
            System.out.println
                ("コード入力：終わりは 999");
            inkode = sc.nextInt();
        }
    }
    public void hyouji(int hsw) {
        if(hsw==0) {
            System.out.println
                (inkode+"は"+eki[cyuuou]);
        }else{
            System.out.println("エラーです");
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        NibuTansaku nibu = new NibuTansaku();
        nibu.( 4 );
    }
}
```

(1)		(3)	
(2)		(4)	

問 13. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

コードと点数を入力し集計して表示する。

入力データ

コード	点数
××	××

実行結果

```

コード入力：終わりは 999
102
点数を入力してください
78
(省略)
101 は 90
102 は 78
(省略)
109 は 60
    
```

処理条件

1. キーボードからデータを入力する。
2. 結果を実行結果のようにディスプレイに表示する。
3. 二分探索を用いること。

```

import java.util.Scanner;
public class NibuTansaku2 {
Scanner sc = new Scanner(System.in);
int[] kode =
    {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
int[] tensu = new int[10];
int inkode, intensu, soeji, n, jogen, kagen, cyuuou;
public void syukei() {
    System.out.println
        ("コード入力：終わりは 999");
    inkode = sc.nextInt();
    while(inkode != 999) {
        System.out.println
            ("点数を入力してください");
        intensu = sc.nextInt();
        jogen= ( 1 );
        kagen=0;
        cyuuou = (jogen+kagen)/2;
        while(inkode != ( 2 ) ){
            if (kode[cyuuou]>inkode) {
                jogen = cyuuou-1;
            }else if(kode[cyuuou]<inkode) {
                ( 3 ) = cyuuou+1;
            }
            cyuuou = (jogen+kagen)/2;
        }
        tensu[cyuuou] += intensu;
        System.out.println
            ("コード入力：終わりは 999");
        inkode = sc.nextInt();
    }
}
public void hyouji() {
    for (n=0;n<tensu.length; ( 4 ) ){
        System.out.println(kode[n]+"は"+tensu[n]);
    }
}
public static void main(String[] args) {
    // TODO 自動生成されたメソッド・スタブ
    NibuTansaku2 nibu = new NibuTansaku2();
    nibu.syukei();
    nibu.hyouji();
}
}
    
```

(1)		(3)	
(2)		(4)	

問 14. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

二次元配列を用いて「Z」の模様を「N」になるように表示する。

入力データ

なし

実行結果

```

ZZZZZ
Z
  Z
    Z
ZZZZZ
Z  Z
ZZ  Z
ZZZ  Z
Z  ZZ
Z   Z

```

```

import java.util.Scanner;
public class Hairetsu2ast {
    Scanner sc = new Scanner(System.in);
    final String[][] mae = {{"Z","Z","Z","Z","Z"},
                            {" "," "," ","Z"," "},
                            {" "," ","Z"," "," "},
                            {" ","Z"," "," "," "},
                            {"Z","Z","Z","Z","Z"};
    String[][] ato = new String[5][5];
    int n,m;
    public void maehyouji() {
        for(n=0;n<mae.length;n++){
            for(m=0;m<( 1 ).length;m++){
                System.out.print(mae[n][m]);
            }
            System.out.println();
        }
    }
    public void atohyouji() {
        for(n=0;n<ato.length;n++){
            for(m=0;m<ato[n].length;m++){
                System.out.print( ( 2 ) );
            }
            System.out.println();
        }
    }
    public void irekae() {
        for(n=0;n<mae.length;n++){
            for(m=0;m<mae[n].length;m++){
                ato[n][m]= ( 3 );
            }
        }
    }
    public static void main(String[] args) {
        Hairetsu2ast hai = new Hairetsu2ast();
        hai.maehyouji();
        hai.( 4 );
        hai.atohyouji();
    }
}

```

(1)		(3)	
(2)		(4)	



問 15. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

グループおよびコードごとに入力した点数を配列に集計後、表示する。

入力データ

コード	グループ	点数
××	××	××

実行結果

```

コード入力：終わりは 999
105
グループ入力：0~9
0
点数を入力してください
67
(省略)
グループ 0 は 00000670000
グループ 1 は 000010000000
(省略)
    
```

処理条件

1. キーボードからデータを入力する。

```

import java.util.Scanner;
public class Hairetsu2Syukei {
    Scanner sc = new Scanner(System.in);
    int[] kode =
        {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
    int[][] tensu = new int[10][10];
    int inkode;
    int ingroup;
    int intensu;
    int soeji;
    int n;
    int m;
    public void syukei() {
        System.out.println
            ("コード入力：終わりは 999");
        inkode = sc.nextInt();
        while(inkode != 999) {
            System.out.println
                ("グループ入力：0~9");
            ingroup = sc.nextInt();
            System.out.println
                ("点数を入力してください");
            intensu = sc.nextInt();
            soeji = inkode - ( 1 );
            tensu[ingroup][soeji] += ( 2 );
            System.out.println
                ("コード入力：終わりは 999");
            ( 3 ) = sc.nextInt();
        }
    }
    public void hyouji() {
        for (n=0;n<tensu.length;n++) {
            System.out.print("グループ"+n+"は");
            for (m=0;m<tensu[n].length;m++) {
                System.out.print( ( 4 ) + " ");
            }
            System.out.println();
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        Hairetsu2Syukei hai = new Hairetsu2Syukei();
        hai.syukei();
        hai.hyouji();
    }
}
    
```

(1)		(3)	
(2)		(4)	

問 16. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

配列 kode の要素を降順にソートする。

入力データ

なし

実行結果

109
108
107
106
105
104
103
102
101
100

```
public class BubbleSort {
    int[] kode =
        {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
    int n;
    int m;
    int taihi;
    public void sort() {
        for(n=8;n>=0; ( 1 ) ){
            for(m=0;m<=n;m++){
                if(kode[m]<kode[m+1]){
                    taihi = kode[m];
                    kode[m]= ( 2 );
                    kode[m+1]=taihi;
                }
            }
        }
    }
    public void hyouji() {
        for (n=0;n<kode.length;( 3 ) ){
            System.out.println(kode[n]);
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        BubbleSort bub = new BubbleSort();
        bub.( 4 );
        bub.hyouji();
    }
}
```

(1)		(3)	
(2)		(4)	

問 17. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

配列 kode の要素を昇順にソートする。

入力データ

なし

実行結果

100
101
102
103
104
105
106
107
108
109

```
public class BubbleSort2 {
    int[] kode =
        {109, 108, 107, 106, 105, 104, 103, 102, 101, 100};
    int n;
    int m;
    int taihi;
    public void sort() {
        for (n=8;n>=0;n--){
            for ( ( 1 ) ){
                if(kode[m]>kode[m+1]){
                    ( 2 );
                }
            }
        }
    }
    public void koukan() {
        taihi = kode[m];
        kode[m]=kode[m+1];
        kode[m+1]= ( 3 );
    }
    public void hyouji() {
        for (n=0;n<kode.length;n++){
            System.out.println(kode[n]);
        }
    }

    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        BubbleSort2 bub = new BubbleSort2();
        bub. ( 4 );
        bub.hyouji();
    }
}
```

( 1 )		( 3 )	
( 2 )		( 4 )	

問 18. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

配列 kode の要素を降順にソートする。

入力データ

なし

実行結果

109  
108  
107  
106  
105  
104  
103  
102  
101  
100

```
public class InsertSort {
    int[] kode =
        {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
    int n;
    int m;
    int taihi;
    public void sort() {
        for (n=1;n<=9;n++){
            m = ( 1 );
            taihi = kode[n];
            while (m >= 0 && kode[m] < taihi) {
                kode[m+1] = kode[m];
                m--;
            }
            ( 2 ) = taihi;
        }
    }
    public void hyouji() {
        for (n=0;n<kode.length;n++){
            System.out.println( ( 3 ) );
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        InsertSort ins = new InsertSort();
        ins.sort();
        ins.( 4 );
    }
}
```

(1)		(3)	
(2)		(4)	

問 19. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

入力したコードを降順にソートする。

入力データ

コード
××

実行結果

10 個のコードを入れてください
5
2 個目
4
(省略)
10 個目
2
11 個目
9
9
8
7
6
5
4
3
3
2

```

import java.util.Scanner;
public class InsertSort2 {
    Scanner sc = new Scanner(System.in);
    int[] kode = new int[10];
    int n;
    int m;
    int taihi;
    public void sort() {
        int inkode;
        for(n=1;n<=10;n++){
            inkode = sc.nextInt();
            m=9;
            kode[m]= ( 1 );
            while( ( 2 )&&kode[m-1]<kode[m]){
                taihi = kode[m-1];
                kode[m-1]=kode[m];
                kode[m]=taihi;
                m--;
            }
            System.out.println( ( 3 ) ) + "個目";
        }
    }
    public void hyouji() {
        for (n=0;n<kode.length;n++){
            System.out.println( ( 4 ) );
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        InsertSort2 ins = new InsertSort2();
        System.out.println
            ("10 個のコードを入れてください");
        ins.sort();
        ins.hyouji();
    }
}

```

処理条件

1. キーボードからデータを入力する。

(1)		(3)	
(2)		(4)	

問 20. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

配列 kode の要素を降順にソートする。

入力データ

なし

実行結果

109
108
107
106
105
104
103
102
101
100

```
public class SelectionSort {
    int[] kode =
        {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
    int n;
    int ( 1 );
    int taihi;
    public void sort() {
        for (n=0;n<( 2 );n++){
            for (m=n+1;m<kode.length;m++){
                if(kode[n]<( 3 )){
                    taihi = kode[n];
                    kode[n]=kode[m];
                    kode[m]=taihi;
                }
            }
        }
    }
    public void ( 4 ){
        for (n=0;n<kode.length;n++){
            System.out.println(kode[n]);
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        SelectionSort sele = new SelectionSort();
        sele.sort();
        sele.hyouji();
    }
}
```

(1)		(3)	
(2)		(4)	

問 21. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

配列 kode の要素を降順にソートする。

入力データ

なし

実行結果

109
108
107
106
105
104
103
102
101
100

```
public class SelectionSort2 {
    Scanner sc = new Scanner(System.in);
    int[] kode =
        {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
    int n;
    int m;
    int taihi;
    int soeji;
    int ( 1 );
    public void sort() {
        for (n=0;n<kode.length-1;n++) {
            soeji = n;
            max = kode[n];
            for (m= ( 2 );m<kode.length;m++) {
                if (max<kode[m]) {
                    soeji = m;
                    max = kode[m];
                }
            }
            if (n != ( 3 )) {
                taihi = kode[n];
                kode[n]=kode[soeji];
                kode[soeji]=taihi;
            }
        }
    }
    public void hyouji() {
        for ( ( 4 );n<kode.length;n++) {
            System.out.println(kode[n]);
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        SelectionSort2 sele2 = new SelectionSort2();
        sele2.sort();
        sele2.hyouji();
    }
}
```

(1)		(3)	
(2)		(4)	

問 22. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

配列 kode の要素を降順にソートする。

入力データ

なし

実行結果

109
108
107
106
105
104
103
102
101
100

```
public class SelectionSort3_1 {
    public static void main(String[] args) {
        int[] kode =
            {100, 101, 102, 103, 104, 105, 106, 107, 108, 109};
        SelectionSort3_2 sele3_2 =
            new SelectionSort3_2();
        hyouji(sele3_2.sort( ( 1 ) ));
    }
    public static void hyouji(int[] kode){
        int ( 2 );
        for (n=0;n<kode.length;n++){
            System.out.println(kode[n]);
        }
    }
}
public class SelectionSort3_2 {
    public int[] sort(int[] kode) {
        int ( 3 );
        for (n=0;n<kode.length-1;n++){
            for (m=n+1;m<kode.length;m++){
                if(kode[n]<kode[m]){
                    taihi = kode[n];
                    kode[n]=kode[m];
                    kode[m]=taihi;
                }
            }
        }
        return ( 4 );
    }
}
```

( 1 )		( 3 )	
( 2 )		( 4 )	



問 23. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

ArrayList を用いて入力した名前を記憶し、表示する。

入力データ

名前
××

実行結果

名前入力：終わりは end
andou
(省略)
katou
名前：終わりは end
end
andou
itou
katou

処理条件

1. キーボードからデータを入力する。

```
import java.util.ArrayList;
import java.util.Scanner;
public class ArrayListInOut {
    Scanner sc = new Scanner(System.in);
    String innamae;
    String naiyou;
    int n;
    ArrayList<String> ( 1 ) = new ArrayList<String>();
    public void nyuryoku() {
        System.out.println
        ("名前入力：終わりは end");
        innamae = sc.next();
        while(innamae. ( 2 ) ("end") != true) {
            list.add(innamae);
            System.out.println
            ("名前入力：終わりは end");
            innamae = ( 3 );
        }
    }
    public void hyouji() {
        for (n=0;n<list.size();n++) {
            String naiyou = list. ( 4 );
            System.out.println(naiyou);
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        ArrayListInOut arr = new ArrayListInOut();
        arr.nyuryoku();
        arr.hyouji();
    }
}
```

(1)		(3)	
(2)		(4)	

問 24. プログラムの説明を読んで、プログラムの ( ) を答えなさい。

<プログラムの説明>

処理内容

ArrayList を用いて入力した名前を記憶し、表示する。

入力データ

名前
××

実行結果

終わりは end
andou
終わりは end
itou
終わりは end
katou
終わりは end
end
Hello1andou
Hello2itou
Hello3katou

処理条件

1. キーボードからデータを入力する。

```
import java.util.ArrayList;
import java.util.Scanner;
public class ArrayListInOut2 {
    Scanner sc = new Scanner(System.in);
    String innamae;
    String naiyou;
    int n;
    ArrayList<String> list = new ArrayList<String>();
    public void nyuryoku() {
        n = ( 1 );
        System.out.println("終わりは end");
        innamae = sc.next();
        while(innamae.equals("end") != true) {
            list.add(kotoba()+n+( 2 ));
            n++;
            System.out.println("終わりは end");
            innamae = sc.next();
        }
    }
    public String kotoba() {
        String koto = "Hello";
        return ( 3 );
    }
    public void hyouji() {
        for (n=0; ( 4 );n++){
            String naiyou = list.get(n);
            System.out.println(naiyou);
        }
    }
    public static void main(String[] args) {
        // TODO 自動生成されたメソッド・スタブ
        ArrayListInOut2 arr = new ArrayListInOut2();
        arr.nyuryoku();
        arr.hyouji();
    }
}
```

(1)		(3)	
(2)		(4)	

## 1 級解答

	1	2	3	4
問 1	Bunkatsu1_2()	hyouji1()	hyouji2()	Bunkatsu1_2()
問 2	nyuryoku	sc. next()	hyouji1(nyuryoku)	hyouji2()
問 3	se	nyuuryoku	setNyuuryoku	hyouji
問 4	hyouji	getNyuuryoku()	nyuuryoku	nyuuryoku
問 5	hyoujioya()	extends		
問 6	kotae	kaikotae	n	kaikotae
問 7	n	m	kotae	tasi.getsuutim()
問 8	san[i]	hyouji	this.name	tuika()
問 9	kekka	kekka	kotae	s4
問 1 0	asahyouji()	hiruhyouji()	asahyouji()	super
問 1 1	m	n	getM()	hyouji()
問 1 2	(jogen+kagen)/2	cyuuou+1	1	tansaku()
問 1 3	9	kode[cyuuou]	kagen	n++
問 1 4	mae[n]	ato[n][m]	mae[m][4-n]	irekae()
問 1 5	100	intensu	inkode	tensu[n][m]
問 1 6	n--	kode[m+1]	n++	sort()
問 1 7	m=0;m<=n;m++	koukan()	taihi	sort()
問 1 8	n-1	kode[m+1]	kode[n]	hyouji()
問 1 9	inkode	m>0	n+1	kode[n]
問 2 0	m	kode.length-1	kode[m]	hyouji()
問 2 1	max	n+1	soeji	n=0
問 2 2	kode	n	n, m, taihi	kode
問 2 3	list	equals	sc. next()	get(n)
問 2 4	1	inname	koto	n<list.size()