

2024年度	後期理解度テスト2試験問題 解答		
科目名	JavaプログラミングⅡ		
担当教員	石原真紀夫		
実施日付	1月14日（火）	1時限目	09:00～（1組2組/A11）
持ち込み	禁 止		
情報工学科	年	組	学籍番号
氏名			

点／100 点

【問 1】次は Java に関する記述です。各記述が正しい場合はoを、誤っている場合はxを答えなさい。【各 2 計 50】

- (1) サブクラス型の変数にはスーパークラスのオブジェクトを代入できます
- (2) Java のすべてのクラスは Object クラスのメンバーを継承します
- (3) メソッドのオーバーライドはポリモρφイズムを実現します
- (4) abstract 修飾子がついた変数の値は変更できません
- (5) 抽象メソッドは常に仮引数を持ちません
- (6) 抽象クラスのオブジェクトは生成できません
- (7) インタフェースはコンストラクタを持ちます
- (8) インタフェースを実装してクラスを宣言します
- (9) インタフェースをクラスと組み合わせることを拡張といいます
- (10) 発生した例外に関する情報は例外クラスのオブジェクトにより保持されます
- (11) 例外クラスは Throwable クラスのメンバーを継承します
- (12) プログラマは例外を発生（送出）するコードを書くことができます
- (13) ストリームとは異なる入出力機器から（へ）のデータを統一的に扱う考え方です
- (14) 文字ストリームは文字ベースのデータを読み書きするストリームです
- (15) 標準入力とはマウスのことです
- (16) コマンドライン引数は半角スペースで区切り複数個与えることができます
- (17) コマンドライン引数は String クラスの配列で受け取ることができます
- (18) コマンドライン引数の数の上限は 10 です
- (19) Java は実行できるマルチスレッドの数の上限は 10 です
- (20) Java はシングルスレッド処理のみを記述できます
- (21) GUI アプリケーションは画像を描画することはできません
- (22) GUI アプリケーションは線や円などのグラフィックスを描画することができます
- (23) GUI は Graphical User Interface の略です
- (24) GUI アプリケーションではマルチスレッドを利用できません
- (25) ウィンドウのオープンやクローズを表すイベントは受け取ることはできません

【問 2】次は Java で用いる用語です。各用語に関連する選択肢を 1 つずつ選び記号で答えなさい。【各 2 計 16】

- | | |
|----------------|------------------------------|
| (1) クラスの拡張の禁止 | (5) 同期 |
| (2) 標準入力 | (6) ウィンドウとウィンドウ部品を扱うためのライブラリ |
| (3) スレッド | (7) 例外の送出 |
| (4) マウスのイベント処理 | (8) ファイル入力 |

■選択肢

- | | |
|----------------|-------------------|
| (A) throw | (E) final |
| (B) Thread | (F) System.in |
| (C) AWT | (G) MouseListener |
| (D) FileReader | (H) synchronized |

【問 3】例外処理を含むコード 1 を実行して以下のようにキーボード入力した場合の画面出力を正確に答えなさい。ここで、[改行]はエンターキーを押すことを意味する。何も画面出力がない場合は「なし」と答えなさい。【各 2 計 8】

- | | | | |
|-------------|------------|------------|------------|
| (1) -1 [改行] | (2) 1 [改行] | (3) 2 [改行] | (4) a [改行] |
| 1 [改行] | 1 [改行] | 1 [改行] | |
| 1 [改行] | 0 [改行] | 0 [改行] | |

■コード 1

```
01:import java.io.*;
02:class Final03{
03:    public static void main(String[] args){
04:        BufferedReader br;
05:        br=new BufferedReader(new InputStreamReader(System.in));
06:        int[] data;
07:        int num, index, value;
08:        try{
09:            num = Integer.parseInt(br.readLine());
10:            index = Integer.parseInt(br.readLine());
11:            value = Integer.parseInt(br.readLine());
12:            data = new int[num];
13:            data[index] = value;
14:        }catch(ArrayIndexOutOfBoundsException e){
15:            System.out.println("例外 1 ");
16:        }catch(NumberFormatException e){
17:            System.out.println("例外 2");
18:        }catch(NegativeArraySizeException e){
```

```
19:            System.out.println("例外 3");
20:        }catch(IOException e){
21:            System.out.println("例外 4");
22:        }
23:    }
24: }
```

【問 4】次はスレッドを用いたコード 2 とその実行結果です。空欄を埋めなさい。【各 2 計 8】
■コード 2

```
01:class Sum implements 【 1 】{
02:    private int from, to, sum;
03:    public Sum(int f, int t){
04:        from=f;
05:        to=t;
06:    }
07:    public void 【 2 】(){
08:        sum=0;
09:        for(int i=from;i<=to;i++) sum+=i;
10:    }
11:    public int getSum(){
12:        return sum;
13:    }
14:}
15:class Final04{
16:    public static void main(String[] args){
17:        Sum sm1 = new Sum(1, 5);
18:        Sum sm2 = new Sum(6, 10);
19:        Thread th1=new Thread(sm1);
20:        Thread th2=new Thread(sm2);
21:        th1.【 3 】();
22:        th2.【 3 】();
23:        int sum=0;
24:        try{
25:            th1.【 4 】();
26:            th2.【 4 】();
27:            sum=sm1.getSum()+sm2.getSum();
28:        }catch(InterruptedException e){
29:            sum=-1;
30:        }
31:        System.out.println("総計 "+sum);
32:    }
33:}
■実行結果
```

総計 55

【問 5】次はファイル入力を用いたコード 3 と入力ファイル in.txt の内容です。入力ファイルを次のようにした場合の画面出力を正確に答えなさい。【各 2 計 8】

- | | | | |
|--------|-------|-------|-------|
| (1) -1 | (2) 0 | (3) 4 | (4) 6 |
| 2 | 2 | 2 | 2 |
| 3 | 3 | 3 | 3 |
| 6 | 6 | 6 | 6 |
| 4 | 4 | 4 | 4 |
| 1 | 1 | 1 | 1 |

■コード 3

```
01:import java.io.*;
02:class Final05{
03:    public static void main(String[] args) throws IOException{
04:        BufferedReader br;
05:        br=new BufferedReader(new FileReader("in.txt"));
06:        int num=Integer.parseInt(br.readLine());
07:        int i,sum=0;
08:        for(i=0;i<num;i++){
09:            String data;
10:            if((data=br.readLine())==null)break;
11:            sum+=Integer.parseInt(data);
12:        }
13:        if(i==num){
14:            System.out.println(sum);
15:        }else{
16:            System.out.println("err");
17:        }
18:        br.close();
19:    }
20:}
```

【問 6】GUI アプリケーションに関する次のコード 4 を実行したら、実行結果のような動作をした。選択肢から 1 つずつ選び、空欄を埋めなさい。【各 2 計 10】

■コード 4

```
01:import java.awt.*;
02:import java.awt.event.*;
```

```
03:class Final06 extends 【 1 】{
04:    private Button bt;
05:    private Label sw;
06:    private int state=0;
07:    private String[] msg={"遮断","通電"};
08:    public Final06(){
09:        super("スイッチ");
10:        bt=new Button(msg[state==0?1:0]+"に変更");
11:        sw=new Label(msg[state]);
12:        Panel pl=new Panel();
13:        pl.add(bt);
14:        pl.add(sw);
15:        add(pl);
16:        addWindowListener(new MyWindowListener());
17:        bt.addActionListener(new MyActionListener());
18:        setVisible(true);
19:    }
20:    class MyWindowListener extends 【 2 】{
21:        public void windowClosing(WindowEvent e){
22:            System.exit(0);
23:        }
24:    }
25:    class 【 3 】 implements ActionListener{
26:        public void actionPerformed(【 4 】 e){
27:            state=(state==0?1:0);
28:            【 5 】.setText(msg[state]);
29:            bt.setLabel(msg[state==0?1:0]+"に変更");
30:        }
31:    }
32:    public static void main(String[] args){
33:        Final06 obj=new Final06();
34:    }
35:}
```

参考

クラス Panel：2 つのボタンのように複数の GUI 部品を配置するには Panel クラスを用います。Panel クラスは複数の GUI 部品を配置できるパネルです。このパネルに 2 つのボタンを配置し、ウィンドウに配置するには次のようにします。

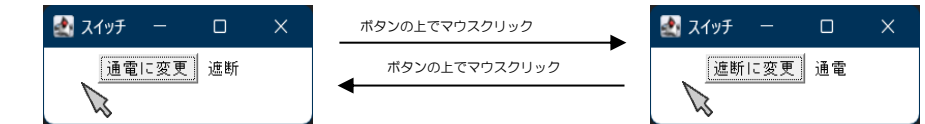
Panel p = new Pane();

p.add(new Button("ボタン 1")); // パネルに 1 つ目のボタンを配置

p.add(new Button("ボタン 2")); // パネルに 2 つ目のボタンを配置

add(p); // Frame にパネルを配置

■実行結果



■選択肢

Frame Thread Runnable implements extends sw bt WindowListener WindowAdapter
MouseListener MouseAdapter MyWindowListener MyActionListener WindowEvent
MouseEvent ActionEvent

解答欄

【問 1】

(1) ×	(2) ○	(3) ○	(4) ×	(5) ×
(6) ○	(7) ×	(8) ○	(9) ×	(10) ○
(11) ○	(12) ○	(13) ○	(14) ○	(15) ×
(16) ○	(17) ○	(18) ×	(19) ×	(20) ×
(21) ×	(22) ○	(23) ○	(24) ×	(25) ×

【問 2】

(1)	E	(2)	F	(3)	B	(4)	G	(5)	H	(6)	C	(7)	A	(8)	D
-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---	-----	---

【問 3】

(1)	例外 3	(2)	例外 1	(3)	なし	(4)	例外 2
-----	------	-----	------	-----	----	-----	------

【問 4】

(1)	Runnable	(2)	run	(3)	start	(4)	join
-----	----------	-----	-----	-----	-------	-----	------

【問 5】

(1)	err	(2)	0	(3)	15	(4)	err
-----	-----	-----	---	-----	----	-----	-----

【問 6】

(1)	Frame	(2)	WindowAdapter	(3)	MyActionListener	(4)	ActionEvent	(5)	sw
-----	-------	-----	---------------	-----	------------------	-----	-------------	-----	----

お疲れ様でした!!