

氏名

点/104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カプセル化を実現します
- (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戻り値を持たないメソッドです
- (6) 抽象クラスのオブジェクトは生成できません
- (7) インタフェースはメンバーとして処理内容が定義されたメソッドをもつことができます
- (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実装といいます
- (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) 例外が発生したら、プログラムは常に強制終了します
- (12) プログラムは例外を発生(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(ハ)のデータを统一的に扱う考え方で
- (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) 標準入力とはディスプレイのことです
- (16) コマンドライン引数はメインメソッドの仮引数で受け取ります
- (17) コマンドライン引数は文字列の配列としてメインメソッドに渡されます
- (18) 1つのスレッドは1つの実行箇所をもつ一連の処理の流れです
- (19) Javaはマルチスレッド処理を記述できます
- (20) マルチスレッド処理では複数のスレッドが並行して実行されます
- (21) GUIアプリケーションは線や円を描画することはできません
- (22) マウスのクリックやドラッグなどのイベントを受け取る処理をイベント処理といいます
- (23) GUIはGraphical User Interfaceの略です
- (24) GUIアプリケーションはボタンやチェックボックスなどのウィンドウ部品を配置できます
- (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) 例外の送出                    |
| (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) 同期         | (7) スレッド                     |
| (4) マウスのイベント処理 | (8) ファイル入力                   |

■選択肢

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

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

■コード1

```
1: import java.io.*;
2: class Final3{
3:     public static void main(String[] args){
4:         BufferedReader br;
5:         br=new BufferedReader(new InputStreamReader(System.in));
6:         int[] data;
7:         int num, index, value;
8:         try{
9:             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。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2

```
1: class Company{
2:     private int sum = 0;
3:     public int add(int a){
4:         sum += a;
5:     }
6:     public int getSum(){return sum;}
7:     public void setSum(int s){sum = s;}
8: }
9: class Driver implements Runnable{
10:    private Company comp;
11:    public Driver(Company c){
12:        comp = c;
13:    }
14:    public void run(){
15:        for(int i=0;i<3;i++){
16:            comp.add(50);
17:        }
18:    }
19: }
20: class Final4{
21:     public static void main(String[] args){
22:         Company cmp = new Company();
23:         Thread th1 = new Thread(new Driver(cmp));
24:         Thread th2 = new Thread(new Driver(cmp));
25:         th1.start();
26:         th2.start();
27:         try{
28:             th1.join();
29:             th2.join();
30:         }catch(InterruptedException e){
31:             cmp.setSum(-1);
32:         }
33:         System.out.println("利益"+cmp.getSum()+"円");
34:     }
35: }
```

■実行結果

利益 300円

■選択肢

sleep start run Runnable Thread join paint throw throws  
 IOException synchronized static try catch finally

【問5】入出力に関する次の説明文の空欄を選択肢から選びなさい。【各2計10】

〔入出力に関する説明文〕

データの出入力は、( 1 ) という概念を用います。( 1 ) は異なる機器からのデータを统一的に扱う考え方で、文字ストリームとバイトストリームがあります。バイトストリームは、画像データや音声データなどの( 2 ) をありのままに読み書きすることができます。

( 3 ) とは、ユーザが実行時に入力する文字列を受け取ってプログラムの処理に利用する機能です。与えられた文字列は( 4 ) メソッドの引数として渡されます。たとえば、プログラム Sample を実行する場合、

■java Sample data.txt

とすると、( 5 ) という文字列がプログラムに渡されます。

■選択肢

スーパー サブ 抽象 ストリーム 配列 コンストラクタ ポリモーフィズム  
 テキストデータ バイナリデータ コマンドライン引数 Thread Object Graphics  
 Runnable run() paint() main() java Sample data.txt

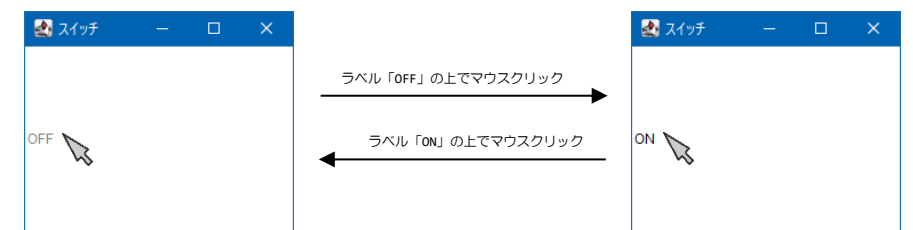
【問6】GUIアプリケーションに関する次のコード3を実行したら、実行結果のような動作をした。選択肢から1つずつ選び、空欄を埋めなさい。【各2計10】

■コード3

```
1: import java.awt.*;
2: import java.awt.event.*;
3: class Final6 extends JPanel{
4:     private Label sw;
5:     private int state=0;
6:     private String[] msg={"OFF","ON"};
7:     private Color[] clr={Color.GRAY,Color.BLACK,};
8:     public Final6(){
9:         super("スイッチ");
10:        sw=new Label(msg[state]);
```

```
11:        sw.setForeground(clrs[state]);
12:        add(sw);
13:        setSize(250,200);
14:        setVisible(true);
15:        addWindowListener(new MyWindowListener());
16:        sw.addMouseListener(new MyMouseListener());
17:    }
18:    class MyWindowListener extends WindowAdapter{
19:        public void windowClosing(WindowEvent e){
20:            System.exit(0);
21:        }
22:    }
23:    class MyMouseListener extends MouseAdapter{
24:        public void mouseClicked(MouseEvent e){
25:            state = state == 0 ? 1 : 0;
26:            sw.setText(msg[state]);
27:            sw.setForeground(clrs[state]);
28:        }
29:    }
30:    public static void main(String[] args){
31:        Final6 obj=new Final6();
32:    }
33: }
```

■実行結果



■選択肢

Frame Thread Runnable implements extends MouseListener MyMouseListener  
 MyWindowListener Final6 MouseEvent WindowEvent start run paint main

解答欄

【問1】

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

【問2】

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

【問3】

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

【問4】

(1) synchronized	(2) Runnable	(3) run	(4) start	(5) catch
------------------	--------------	---------	-----------	-----------

【問5】

(1) ストリーム	(2) バイナリデータ	(3) コマンドライン引数	(4) main()	(5) data.txt
-----------	-------------	---------------	------------	--------------

【問6】

(1) Frame	(2) extends	(3) MyMouseListener	(4) MouseEvent	(5) Final6
-----------	-------------	---------------------	----------------	------------

お疲れ様でした!!