

平成28年度 後期理解度テスト2試験問題
 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23日(月) 5時限目(1組 A36/2組 A37)
 持ち込み 許可・禁止
 情報工学科 年 組 学籍番号
 氏名

_____点/102点

【問1】次はJavaに関する記述です。各記述が正しい場合は○を、誤っている場合は×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) 抽象メソッドとは処理本体が未定義のメソッドです
- (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) 抽象クラスはコンストラクタを宣言できます
- (7) インタフェースのメンバーは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) クラスを拡張してインタフェースを宣言します
- (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) 独自の例外を表わすクラスは宣言できません
- (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) プログラマは例外を発生させるコードを書くことはできません
- (13) 発生した例外に関する情報はスレッドクラスのオブジェクトにより管理されます
- (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) 標準入力とはディスプレイのことです
- (16) ストリームはディスプレイやファイルなど異なる入出力機器を統一的な方法で扱うためのしくみです
- (17) 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
- (18) コマンドライン引数の数は最大2個です
- (19) Javaはマルチスレッド処理を記述できます
- (20) 複数のスレッドは常にそれを開始した順番に終わります
- (21) アプレットは線や円などのグラフィックスを描画することはできません
- (22) スレッド間では同期をとることができません
- (23) アプレットはWebブラウザ上で動作します
- (24) マウスのクリックやドラッグなどのイベントに関する処理をイベント処理といいます
- (25) アプレットはメインメソッドを宣言しなくても動作します

【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
- (2) バッファ機能つき入力
- (3) 同期
- (4) オブジェクトの属するクラスの検査
- (5) 例外の送出
- (6) 抽象クラス
- (7) スレッド
- (8) マウスのイベント処理

■選択肢

- (A) final
- (B) Thread
- (C) throw
- (D) abstract
- (E) instanceof
- (F) synchronized
- (G) ActionListener
- (H) BufferedReader

【問3】例外処理に関する下のコード1を以下のように実行した場合の画面出力を正確に答えなさい。【各2計10】

- (1) java Final3
- (2) java Final3 2 1 3
- (3) java Final3 -1 0 1
- (4) java Final3 1 Hello
- (5) java Final3 3 2 3 0

■コード1

```
1: class Final3{
2:     public static void main(String[] args){
3:         if(args.length>=1){
4:             try{
5:                 int[] num=new int[Integer.parseInt(args[0])];
6:                 for(int i=0;i<num.length;i++){
7:                     num[i]=Integer.parseInt(args[i+1]);
8:                 }

```

```
9:         int sum=0;
10:        for(int i=0;i<num.length;i++){
11:            sum+=num[i];
12:        }
13:        System.out.println(sum);
14:    }catch(NegativeArraySizeException e){
15:        System.out.println("不正な配列サイズ");
16:    }catch(NumberFormatException e){
17:        System.out.println("不正な整数データ");
18:    }
19:    }else{
20:        System.out.println("パラメータエラー");
21:    }
22: }
23: }
```

NegativeArraySizeException 例外：負のサイズを持った配列を作成しようとした場合

NumberFormatException 例外：文字列を数値型に変換するとき文字列の形式が正しくない場合

【問4】メソッドのオーバーライドに関する下のコードをそれぞれ実行した場合の画面出力を正確に答えなさい。クラスBuildingとSchoolの宣言はコード2に示す。【各2計6】

- (1) Building a=new Building(12);
System.out.println(a.toString());
- (2) Building b=new School(6,20);
System.out.println(b.toString());
- (3) School c=new School(25,125);
System.out.println(c);

■コード2

```
1: class Building{
2:     private int story;
3:     public Building(int s){
4:         story=s;
5:     }
6:     public String toString(){
7:         return story+"階建";
8:     }
9: }
10:
11: class School extends Building{
12:     private int lecturerroom;
13:     public School(int s, int r){
14:         super(s);
15:         lecturerroom=r;
16:     }
17:     public String toString(){
18:         return super.toString()+"/"+lecturerroom+"講義室";
19:     }
20: }
```

【問5】スレッド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8】

〔スレッドに関する説明文〕
 スレッドを構築する方法の1つに(1)クラスの拡張があります。(1)クラスから継承される(2)メソッドをオーバーライドして、並行して実行したい処理を記述します。新しいスレッドを起動するには、拡張されたサブクラスのオブジェクトを生成し、(1)クラスから継承される(3)メソッドを実行します。さらに、(4)メソッドを実行すると、そのスレッドの終了を待つことができます。

■選択肢

Object Thread Runnable Exception Applet Graphics join() run() start() paint()

【問6】アプレットに関する次の設問に答えなさい。

設問1 コード3を実行し、次のようなマウス操作を行った後のアプレットの描画画面を解答欄に図で正確に示しなさい。【各2計6】

- (1) 座標(2,3)で左ボタンを押し、ドラッグして座標(4,1)でボタンを離します
- (2) 座標(1,1)で左ボタンを押し、ドラッグして座標(2,3)でボタンを離します
- (3) 座標(3,4)で左ボタンを押し、ドラッグして座標(1,4)でボタンを離します

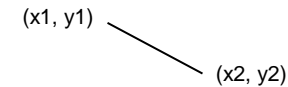
■コード3

```
1: import java.applet.Applet;
2: import java.awt.*;
3: import java.awt.event.*;
4:
5: public class Final6 extends Applet implements ActionListener{
6:     private int[] pos=new int[4];
7:
8:     public void init(){
9:         addMouseListener(this);
10:    }
11:    public void mouseClicked(MouseEvent e){
12:    public void mouseEntered(MouseEvent e){
13:    public void mouseExited(MouseEvent e){

```

```
14:    public void mousePressed(MouseEvent e){
15:        pos[0]=e.getX();
16:        pos[1]=e.getY();
17:    }
18:    public void mouseReleased(MouseEvent e){
19:        pos[2]=e.getX();
20:        pos[3]=e.getY();
21:        repaint();
22:    }
23:    public void paint(Graphics g){
24:        g.drawLine(pos[0],pos[1],pos[2],pos[1]);
25:        g.drawLine(pos[2],pos[1],pos[2],pos[3]);
26:    }
27: }
```

GraphicsクラスのdrawLine(x1, y1, x2, y2)メソッド：次のような直線を描画します



設問2 次はアプレットで使用するクラスまたはメソッドの説明です。それぞれの説明に対応するクラスまたはメソッドを選択肢から選び答えなさい。【各2計6】

- (1) マウス座標やクリックボタンなどマウスイベントに関する情報を管理するクラス
- (2) 点や線、矩形などの図形を描画する機能を持つクラス
- (3) 文字列を描画するメソッド

■選択肢

Graphics MouseEvent MouseListener paint() init() drawString() setFont()

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)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
-----	-----	-----	-----	-----	-----	-----	-----

【問3】

(1)	(2)	(3)	(4)	(5)
-----	-----	-----	-----	-----

【問4】

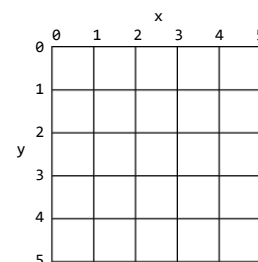
(1)	(2)	(3)
-----	-----	-----

【問5】

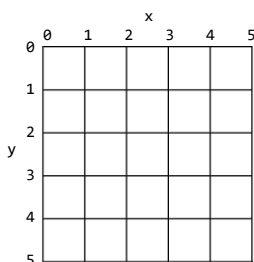
(1)	(2)	(3)	(4)
-----	-----	-----	-----

【問6】設問1

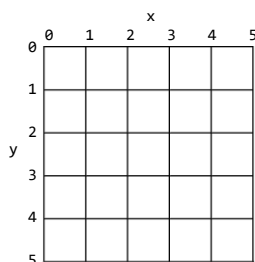
(1) アプレット画面



(2) アプレット画面



(3) アプレット画面



【問6】設問2

(1)	(2)	(3)
-----	-----	-----

お疲れ様でした!!