

計算機ネットワーク I (Java 編) ・ テスト問題用紙

('07 年 8 月 3 日 ・ 13:00 ~ 14:30)

解答上、その他の注意事項

- I. 問題は、問 I ~ III までである。
- II. 解答用紙の右上の欄に学籍番号・名前を記入すること。
- III. 解答欄を間違えないよう注意すること。
- IV. 解答中の文字 (特に a と d) がはっきりと区別できるよう注意すること。
- V. 持ち込みは不可である。
- VI. 合格は 50 点満点中 30 点以上とする。
(配点 — 期末テスト 40 点, レポート 10 点)

I. 次の各選択問題に答えよ。解答は各問の指示する選択肢から選べ。ただし、特に指定しない限り、選ぶべき選択肢は必ずしも 1つとは限らない。

(i) 次のうち、Java のクラスの名前として使用できないのは、どれか？

- (A). Apple1 (B). Lemon.A (C). Orange_X (D). Peach-0

(ii) 次の Java に関する文章のうち間違っているものはどれか？

- (A). Java では public なクラスの名前と、そのクラスを定義しているファイルの名前は一致させなければならない。
(B). Java のコードが HTML ファイルなどにソースファイルの形で埋め込まれたものが JavaScript である。
(C). Java は C 言語に上位互換なオブジェクト指向言語である。
(D). Java は中間言語方式を取り、機種依存性を避けながら効率の良い実行方式を目指している。

(iii) 要素の型が String (文字列) 型であるような、ArrayList 型 (サイズ変更可能な配列の型) の変数 a を宣言したい。正しい書き方を 1つ、以下の選択肢から選べ。

- (A). ArrayList<String> a = new ArrayList<String>();
(B). String<ArrayList> a = new String<ArrayList>();
(C). String<> a = new String<>();
(D). ArrayList.String<> a = new ArrayList.String<>();

(iv) 次のプログラムは、try ~ catch 文を使用するプログラムの例である。(Java では、0 を割る数として整数の割算を行なうと、ArithmeticException という種類の例外が発生する。)

```
public class ExceptionTest {
    public static void main(String[] args) {
        int i;
        for(i=0; i<=10; i++) {
            try {
                System.out.printf("%d ", 60/(i-5));
            } catch (ArithmeticException e) {
                System.out.printf("!? ");
            }
        }
        System.out.println();
    }
}
```

このプログラムの出力として正しいものを 1つ、以下の選択肢から選べ。

- (A). -12 -15 -20 -30 -60
(B). -12 -15 -20 -30 -60 !?
(C). -12 -15 -20 -30 -60 60 30 20 15 12 !?
(D). -12 -15 -20 -30 -60 !? 60 30 20 15 12

- (v) 次のプログラムはホスト名(コンピュータ名)をコマンドライン引数として受け取り、そのコンピュータのHTTPサーバ(80番ポート)に接続して、index.htmlというファイルを受信して表示する。

```
import java.net.*;
import java.io.*;

public class NetworkTest {
    public static void main(String[] argv) {
        byte[] buff = new byte[1024];
        try {
            (1) rwSocket = new (1) (argv[0], 80);
            InputStream instrm = rwSocket.getInputStream();
            OutputStream outstr = rwSocket.getOutputStream();

            outstr.write("GET /index.html HTTP/1.0\r\n\r\n".getBytes());
            // getBytes() は String 型から byte[] 型に変換するメソッド

            while(true) {
                int n = instrm.read(buff);
                if (n==-1) break;
                System.out.write(buff, 0, n);
            }
        } catch (Exception e) {
            System.out.println("error!");
        }
    }
}
```

例えば

```
java NetworkTest stwww.eng.kagawa-u.ac.jp
```

のように起動すると、<http://stwww.eng.kagawa-u.ac.jp/index.html> の内容を表示する。(1)の空欄に当てはまるものを1つ、以下の選択肢から選べ。

- (A). InputStream (B). OutputStream (C). Socket (D). DatagramSocket

(vi) 0 から 4 までの数とその数に 10 を足した数を次のよう出力したい。

```
x が 0 のとき x+10 は 10
x が 1 のとき x+10 は 11
x が 2 のとき x+10 は 12
x が 3 のとき x+10 は 13
x が 4 のとき x+10 は 14
```

次のプログラム:

```
int i;
for(i=0; i<5; i++) {
    [???];
}
```

空欄 [???] にはいる式を下の選択肢の中から 1 つ選べ。

- (A). System.out.println("xが+i+のとき "x"+10は+(i+10)+)
- (B). System.out.println("xが"+i+"のとき x+10は"+i+10)
- (C). System.out.println("xが¥"i¥"のとき x+10は¥"i+10¥")
- (D). System.out.println("xが"+ i +"のとき x+10は" + (i+10))

II. 次のプログラムは、コマンドライン引数の和を計算する。

```
public class CommandLineSum {
    public static void main(String[] args) {
        int i, sum=0;
        for(i=0; i< [ (i) ]; i++) {
            sum += [ (ii) ];
        }
        System.out.printf("答えは %d です。 %n", sum);
    }
}
```

実行例は以下のようになる。

```
% java CommandLineSum 10 20 30 40 50
答えは 150 です。
% java CommandLineSum 1 2 4 8 16
答えは 31 です。
```

(i) ~ (ii) の空欄にふさわしい式を以下の選択肢から選べ。

(i) の選択肢

- (A). args.length (B). sizeof(args) (C). sizeof.args (D). length(args)

(ii) の選択肢

- (A). String.valueOf(args[i]) (B). String.valueOf(args(i))
- (C). Integer.parseInt(args[i]) (D). Integer.parseInt(args(i))

III. 次の文章は String クラスの replace メソッドの説明の Java™ API 仕様からの抜粋である。

```
public String replace(char oldChar, char newChar)
```

この文字列内のすべての oldChar を newChar に置換した結果生成される、新しい文字列を返します。

文字 oldChar がこの String オブジェクトによって表される文字列内にはない場合は、この String オブジェクトへの参照が返されます。そうでない場合は、この String オブジェクトによって表される文字列と同じ文字列を表す、新しい String オブジェクトが生成されます。ただし、文字列内の oldChar はすべて newChar に置換されます。

例:

```
"mesquite in your cellar".replace('e', 'o') では  
"mosquito in your collar"が返される。
```

```
"the war of baronets".replace('r', 'y') では  
"the way of bayonets"が返される。
```

```
"sparring with a purple porpoise".replace('p', 't') では  
"starring with a turtle tortoise"が返される。
```

```
"JonL".replace('q', 'x') では "JonL"(変更なし)が返される。
```

パラメータ:

oldChar – 以前の文字

newChar – 新しい文字

戻り値:

この文字列内のすべての oldChar を newChar に置換することによって生成された文字列

このメソッドを使用し、テストするプログラムを次のように作成する。

ファイル名: ReplaceTest.java

```
public class ReplaceTest {  
    public static void main(String[] args) {  
        String str = args[0];  
          
        System.out.println(str);  
    }  
}
```

このプログラムは、コマンドライン引数を 1 つ受け取り、その中の '?' という文字を、'!' という文字に、 '_' という文字を、空白文字に置換して表示する。

実行例は次のようになる。

```
prompt> java ReplaceTest ?_?_?_Thomas  
!!! Thomas  
prompt> java ReplaceTest ?_?_?_Gordon  
!!! Gordon
```

上のプログラムの空欄 を埋めよ。

