# **問題１：Java言語の基礎知識**

Java言語について以下の設問に答えよ。

**＜設問1-1＞**

以下の文章はJava言語について述べたものである。空欄に入れるべき適切な語句を**解答欄1-1**に記入せよ。

Ａ．Javaプログラムコードを保存するときのファイル名は　　①　　＋拡張子（.java）となる。

Ｂ．プログラムコードはメソッドの　　②　　と　　③　　の間に記述しなければならない。通常、　　②　　と　　③　　に挟まれた部分は少し右へずらす（通常はタブ1つ分）。これを　　④　　と呼ぶ。

Ｃ．　　⑤　　と　　⑥　　を使うと　　⑤　　と　　⑥　　の間に挟まれた部分がコメントになる。　　⑦　　を使うと  
　　⑦　　から行末までがコメントになる。

Ｄ．　　⑧　　または　　⑨　　を使うと画面に文字を出力することができる。　　⑧　　は文字を出力したあと改行をするが、　　⑨　　は改行しない。

Ｅ．文字列は　　⑩　　で囲む。文字（文字コード）は　　⑪　　で囲む。

Ｆ．　　⑫　　クラスを使うとキーボードからの入力を取得することができる。　　⑫　　クラスを使うためにはあらかじめ  
　　⑬　　をインポートする。文字列の入力を受け取るには　　⑭　　メソッドを使う。int型の数値を受け取るには  
　　⑮　　メソッドを使う。

**解答欄1-1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ① | ② | ③ | ④ | ⑤ |
| ⑥ | ⑦ | ⑧ | ⑨ | ⑩ |
| ⑪ | ⑫ | ⑬ | ⑭ | ⑮ |

**＜設問1-2＞**

以下の文章はJava言語の変数の型について述べたものである。空欄に入れるべき適切な語句を**解答欄1-2**に記入せよ。

Ａ．宣言した変数を使うためには値を代入する必要がある。宣言と同時に値を代入することを　　①　　と呼ぶ。

Ｂ．変数に値を代入するには　　②　　を使う。　　②　　の　　③　　の値が　　④　　に代入される。

Ｃ．int型の数値をdouble型の変数に代入すると自動的に　　⑤　　型に変換される。double型の数値をint型の変数に代入するには　　⑥　　が必要で数値の前に　　⑦　　を付ける。このとき小数点以下の数値は　　⑧　　。

Ｄ．long型もdouble型もデータサイズは　　⑨　　であるが、扱える数値の範囲は　　⑩　　型の方が大きい。ただし数値が大きくなると扱える値が飛び飛びになる。

Ｅ．if文やwhile文の条件式の値は　　⑪　　型である。　　⑪　　型に代入できる値は　　⑫　　と　　⑬　　である。

Ｆ．　　⑭　　型は主に文字（文字コード）を扱うときに使う。Aの文字コードが65のときBの文字コードは　　⑮　　である。

**解答欄1-2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ① | ② | ③ | ④ | ⑤ |
| ⑥ | ⑦ | ⑧ | ⑨ | ⑩ |
| ⑪ | ⑫ | ⑬ | ⑭ | ⑮ |

※⑪と⑫は逆でもOK

**＜設問1-3＞**

以下の文章は制御構造について述べたものである。空欄に入れるべき適切な語句を**解答欄1-3**に記入せよ。

Ａ．　　①　　文では　　②　　の値によって処理が分岐する。　　②　　の値が　　③　　のとき処理Aを実行し、  
　　④　　のとき処理Bを実行する。

　　①　　 (条件式) {

（処理A）

} else {

（処理B）

}

Ｂ．　　⑤　　文では　　⑥　　の値によって処理が繰り返される。　　⑥　　の値が　　⑦　　の間、処理Aを繰り返す。

　　⑤　　 (条件式) {

（処理A）

}

Ｃ．　　⑧　　～　　⑨　　文では処理Aを実行した後に、　　⑩　　の値が　　⑪　　だと処理Aを繰り返す。

　　⑧　　 {

（処理A）

} 　　⑨　　 (条件式);

Ｄ．while文で条件式を　　⑫　　とすると無限ループ（永久ループ）になる。　　⑬　　を使うとwhile文を抜け出すことができる。　　⑭　　を使うとループ処理の先頭に戻る。

Ｅ．for文では最初に1回だけAの処理を実行する。その後、　　⑮　　に書かれた条件式の値が　　⑯　　のとき、処理Dを実行したのち、　　⑰　　を実行してからループの先頭に戻る。処理Dの途中で　　⑱　　を使うと処理Dを中断してfor文を抜け出すことができる。　　⑲　　を使うと処理Dを中断して　　⑳　　を実行してからループの先頭に戻る。

for ( A; B; C) {

（処理D）

}

**解答欄1-3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ① | ② | ③ | ④ | ⑤ |
| ⑥ | ⑦ | ⑧ | ⑨ | ⑩ |
| ⑪ | ⑫ | ⑬ | ⑭ | ⑮ |
| ⑯ | ⑰ | ⑱ | ⑲ | ⑳ |

**＜設問1-4＞**

以下の文章はメソッドと配列について述べたものである。空欄に入れるべき適切な語句を**解答欄1-4**に記入せよ。

Ａ．配列の要素数は　　①　　で表される。インデックスは　　②　　から始まるため、　　①　　番目の要素は　　③　　。

Ｂ．String型の文字数は　　④　　メソッドで取得できる。i番目の文字は　　⑤　　メソッド取得できる。String型の文字列同士がが同じかどうか比較するには　　⑥　　メソッドを使う。

Ｃ．メソッド名nobita、引数なし、戻り値なしのメソッドを記述せよ。

public static 　　⑦　　 　　⑧　　(　　⑨　　) {

（処理）

}

Ｄ．メソッド名dekisugi、引数n（int型）、戻り値がboolean型のメソッドを記述せよ。

public static 　　⑩　　 　　⑪　　(　　⑫　　) {

（処理）

}

Ｅ．メソッド名sizuka、引数array（int型配列）、戻り値なしのメソッドを記述せよ。

public static 　　⑬　　 　　⑭　　(　　⑮　　) {

（処理）

}

**解答欄1-4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ① | ② | ③ | ④ | ⑤ |
| ⑥ | ⑦ | ⑧ | ⑨ | ⑩ |
| ⑪ | ⑫ | ⑬ | ⑭ | ⑮ |

# **問題２：Javaプログラムの基礎**

Javaプログラミングに関する以下の設問に答えよ。

**＜設問2-1＞ リスト2-1：名前の入力と表示（ファイル「Name.java」）**

import java.util.Scanner;

public class Name {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

　　A

in.close();

}

}

**リスト2-1**はキーボードから入力した名前を画面に表示するプログラムである。　　A　　に**コード①**から**コード④**のそれぞれを入力したときの実行結果を**解答欄2-1**に記入せよ。なお、実行できないときは「×」を記入すること

**コード① コード②**

System.out.println("名前を入力してください＞");

String name = in.next();

System.out.println(name + "さん、こんにちは！");

System.out.print("名前を入力してください＞");

System.out.println(name + "さん、こんにちは！");

String name = in.next();

**コード③ コード④**

System.out.print("名前を入力してください＞");

String name = in.next();

System.out.println(name + "さん、こんにちは！");

String name = in.next();

System.out.println("名前を入力してください＞");

System.out.println(name + "さん、こんにちは！");

**解答欄2-1**

|  |  |
| --- | --- |
| **コード①の**  **実行結果** |  |
| **コード②の**  **実行結果** |  |
| **コード③の**  **実行結果** |  |
| **コード④の**  **実行結果** |  |

**＜設問2-2＞**

**リスト2-2**は変数のさまざまな演算プログラムである。それぞれの変数の最終的な値を答えよ。

|  |  |
| --- | --- |
| 変数名 | 値 |
| eat |  |
| rest |  |
| apple |  |
| person |  |
| a |  |
| b |  |
| c |  |
| f |  |
| pos |  |
| data[0] |  |
| data[1] |  |
| data[2] |  |
| data[3] |  |
| data[4] |  |
| s2 |  |

**リスト2-2**

public class Arithmetic {

public static void main(String[] args) {

int apple = 13;

int person = 4;

int eat = apple / person;

int rest = apple % person;

int a = 5;

int b = a + 7;

int c = a;

a = b;

b = c;

boolean f = (c == a);

int[] data = {7, 5, 3, 8, 2};

int pos = 0;

for (int i = 1; i < data.length; i++) {

if (data[pos] < data[i]) pos = i;

}

int tmp = data[0];

data[0] = data[pos];

data[pos] = tmp;

int s0 = 0x0f;

int s1 = 0x4c;

int s2 = ~s0 & s1;

}

}