# **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．int型の数値をdouble型の変数に代入すると自動的に　　double　　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　　型変換(キャスト)　　が必要で数値の前に　　(int)　　を付ける。このとき小数点以下の数値は　　切り捨てられる　　。

Ｂ．long型もdouble型もデータサイズは　　64　　ビットであるが、扱える数値の範囲は　　double　　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。

Ｃ．配列の要素数は　　配列名.length　　で取得できる。インデックスは　　0　　から始まるため、　　配列名.length　　番目の要素は存在しない。

Ｄ．if文では　　条件式　　の値によって処理が分岐する。　　条件式　　の値が　　true　　のとき処理Aを実行し、　　false　　のとき処理Bを実行する。

if (　　条件式　　) {

（処理A）

} else {

（処理B）

}

Ｅ．while文では　　条件式　　の値が　　true　　の間、処理Aを繰り返す。　　break　　を使うとwhile文を抜け出すことができる。　　continue　　を使うとループ処理の先頭に戻る。

while(　　条件式　　) {

（処理A）

}

Ｆ．for文では最初に1回だけ　　A　　の処理を実行する。その後、　　B　　に書かれた条件式の値が　　true　　のとき、処理Dを実行したのち、　　C　　の処理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1-1**

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

**＜設問1-2＞**

以下の文章はクラス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．メンバ（フィールドやメソッド）は外部に対して公開するのか非公開にするのか設定することができる。　　public　　を指定すると外部に対して公開するためクラス外からでも使える。　private　を指定すると外部に対し非公開となるためクラス内でしか使えない。　　protected　　を指定するとサブクラスに対しては公開となるためサブクラスからは使うことができる。

Ｂ．クラスを使う側に必要なメンバだけを公開しクラスの中でしか使わないメンバを隠すことを　　カブセル化　　という。

Ｃ．Javaではメソッド名が同じでも　　引数(シグネチャ)　　の仕様が異なればメソッドを複数定義できる。これをメソッドの　　オーバーロード　　と呼ぶ。

Ｄ．フィールドやメソッドの宣言に　　static　　を付けると、インスタンスを生成していなくても使える　　クラス　　メンバとなる。クラス全体でひとつだけ作られ同じクラスであればどのインスタンスからでも参照できる。クラス外から　　　　クラス　　メンバを使うには　　クラス名・メンバ名　　と記述する。

Ｅ．フィールドやメソッドの宣言に　　static　　を付けないと、　　インスタンス　　メンバとなる。　　インスタンス　　　メンバはインスタンスごとに作られるのでインスタンスを生成しないと使えない。クラス外から　　インスタンス　　　メンバを使うには　　インスタンス名・メンバ名　　と記述する。

Ｅ．クラスにはインスタンス生成時に自動的に呼び出される　　コンストラクタ　　と呼ばれるメソッドがある。　　コンストラクタ　　に　　戻り値　　は記述せず、　　メンソッド名　　はクラス名と同じにする。

Ｆ．Javaでは　　new　　でインスタンスの生成を行う。

Ｇ．Javaのデータ型は　　基本型　　（int型やdouble型など）と　　参照型　　（配列やクラスなど）に分けられる。　　基本型　　の変数にはそのデータ型の　　値　　が入る。　　参照型　　の変数にはそのデータ型の　　参照　　が入る。

**解答欄1-2**

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

**＜設問1-3＞**

以下の文章はクラスの継承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．Javaではクラスを継承して新しいクラスを作ることができる。このとき継承元のクラスを　　スーパークラス　　、継承先のクラスを　　サブクラス　　と呼ぶ。　　サブクラス　　には新たにフィールドやメソッドを追加することができる。

Ｂ．クラスを継承するにはクラス宣言で「　　extends　　 スーパークラス名」と記述する。

Ｃ．スーパークラスと同じ仕様のメソッドをサブクラスに定義することができる。これをメソッドの　　オーバーライド　　と呼ぶ。

Ｄ．スーパークラス型の変数でサブクラスのインスタンスを指すことができる。ただし使えるのは　　スーパークラス　　のメンバのみとなる。

Ｅ．スーパークラスのメソッドをサブクラスでオーバーライドすると、スーパークラス型の変数を使ってサブクラス側のメソッドを呼び出すことができるようになる。スーパークラス型の変数が指すサブクラスによって異なる動作をさせることができる。このように呼び出し方が同じでも異なる動作をすることを　　ポリモーフィズム　　と呼ぶ。

Ｆ．サブクラスでオーバーライドしたスーパークラスのメソッドを（サブクラス内から）呼び出すには　　super.メンソッド名　　と記述する。

Ｇ．　　抽象メンソッド　　とはプログラムコードを実装しないメソッドのことで、メソッドの宣言に　　abstract　　を、プログラムコードを定義する中カッコの代わりに　　セミコロン(;)　　を付ける。

Ｈ．　　抽象メンソッド　　を持つクラスを　　抽象クラス　　と呼び、classの前に　　abstract　　を付ける。　　抽象クラス　　型の　　変数　　を作ることはできるが　　インスタンス　　を作ることはできない。

Ｉ．クラス宣言のclassを　　interface　　と記述するとインターフェイスになる。インターフェイスにするとすべてのメソッドが  
　　抽象メンソッド　　になる。インターフェイスを実装するにはクラス宣言で「　　implements　　 インターフェイス名」と記述する。

Ｊ．継承できるクラスは　　1つ　　であるが、実装できるインターフェイスは　　複数　　である。

**解答欄1-3**

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

# **問題２：Java基本文法**

以下で指示された処理について**実行結果**を参考に**リスト2**の　　A　　の部分のプログラムコードを作成せよ。

**リスト2**

import java.util.Scanner;

public class J2Test13\_2 {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

System.out.print("0より大きい整数を入力してください＞");

int n = in.nextInt();

　　A

in.close();

}

}

1. 入力された数値の桁数を表示する処理を、while文を使って作成せよ。

int count = 0;  
while(n != 0){  
 n /= 10;  
 count++;  
 }  
 System.*out*.println(count + "桁です！");

**実行結果①（太字は入力箇所）「1234」を入力したとき**

0より大きい整数を入力してください＞**1234**

4桁です！

1. 1から入力された数値までを加算して表示する処理を、for文を使って作成せよ。

int sum = 0;  
for(int i = 1 ; i <= n ; i++){  
 sum += i;  
}  
System.*out*.println("1から"+n+"まで加算すると"+sum+"です！");

**実行結果②（太字は入力箇所）「10」を入力したとき**

0より大きい整数を入力してください＞**10**

1から10まで加算すると55です！

1. 入力された数値を5で割ったとき、割り切れるかどうか表示する処理を、if～else文を使って作成せよ。

if(n % 5 == 0){  
 System.*out*.println("5で割り切れます！");  
}else{  
 System.*out*.println("5で割り切れません！");  
}

**実行結果③（太字は入力箇所）「100」を入力したとき 実行結果③（太字は入力箇所）「99」を入力したとき**

0より大きい整数を入力してください＞**99**

5で割り切れません！

0より大きい整数を入力してください＞**100**

5で割り切れます！

1. ③と同じ処理を、switch文を使って作成せよ。

switch(n % 5){  
 case 0:System.*out*.println("5で割り切れます！"); break;  
 default:System.*out*.println("5で割り切れません！");break;  
}

**実行結果④（太字は入力箇所）「100」を入力したとき 実行結果④（太字は入力箇所）「99」を入力したとき**

0より大きい整数を入力してください＞**99**

5で割り切れません！

0より大きい整数を入力してください＞**100**

5で割り切れます！

# **問題３：クラスメンバとインスタンスメンバ**

**リスト3-1**および**リスト3-2**は生成したSheepクラスのインスタンスの数を表示する処理である。**実行結果3**となるように**リスト3-1**および**リスト3-2**のコードを作成せよ。

**リスト3-1：Sheepクラス**

public class Sheep {

private 　　①static int　　 counter; // 生成した羊の数

private 　　②int　　 num; // 羊番号（生成した順に1、2、3、・・・）

public 　　③static void　　 showCounter() {

System.out.println("羊は全部で" + 　　④counter　　 + "匹です！");

}

public 　　⑤Sheep()　　 { // コンストラクタ

　　⑥num　　 = ++　　⑦counter　　;

System.out.println(num + "番の羊がやってきた！");

}

}

**リスト3-2：Sheepの生成と生成した数の表示**

public class J2Test13\_3 {

public static void main(String[] args) {

　　⑧Sheep.showCounter()　　; // 羊の数を表示

　　⑨Sheep　　 s1 = 　　⑩Sheep()　　; // s1に羊を1匹生成

　　⑧Sheep.showCounter()　　; // 羊の数を表示

　　⑪Sheep[]　　 sn = 　　⑫new Sheep[3]　　; // snに羊の配列（3匹分）を生成

for (int i = 0; 　　⑬i < sn.length　　; i++) {

　　⑭sn[i]　　 = 　　⑮new Sheep()　　; // 配列に羊を生成

}

　　⑧Sheep.showCounter()　　; // 羊の数を表示

}

}

**実行結果3**

羊は全部で0匹です！

1番の羊がやってきた！

羊は全部で1匹です！

2番の羊がやってきた！

3番の羊がやってきた！

4番の羊がやってきた！

羊は全部で4匹です！

# **問題４：継承**

Monsterクラスを継承してFlyMonsterクラスとSwimMonsterクラスを定義する。

**＜設問4-1＞**

クラス図をもとに**リスト4-1、4-2、4-3**のコードを作成せよ。

|  |
| --- |
| **Monster** |
| # name |
| + Monster(name)  + intro() |

|  |
| --- |
| **FlyMonster** |
|  |
| + FlyMonster(name)  + intro() |

|  |
| --- |
| **SwimMonster** |
|  |
| + SwimMonster(name)  + intro() |

**リスト4-1：Monsterクラス**

public class Monster {

1. protected　　 String name; // 名前

　　②public　　 Monster(String name) {

　　③this.name　　 = name;

System.out.println("Monster！");

}

　　④public　　 void intro() {

System.out.println("ぼくの名前は" + name + "。");

}

}

**リスト4-2：FlyMonsterクラス リスト4-3：SwimMonsterクラス**

public class SwimMonster 　⑤extends Monster　　 {

public SwimMonster(String name) {

　　⑥super(name)　　;

System.out.println("SwimMonster！");

}

public void intro() {

　　⑦super.intro()　　; // Monsterのintro

System.out.println("泳ぎが得意さ！");

}

}

public class FlyMonster 　⑤extends Monster　　 {

public FlyMonster(String name) {

　　⑥super(name)　　;

System.out.println("FlyMonster！");

}

public void intro() {

　　⑦super.intro()　　; // Monsterのinro

System.out.println("空をとべるよ！");

}

}

**＜設問4-2＞**

**リスト4-4**はFlyMonsterとSwimMonsterに自己紹介させる処理である。**リスト4-4**を入力し実行結果を確認せよ。

**リスト4-4：FlyMonsterとSwimMonsterの自己紹介**

public class J2Test13\_4 {

public static void main(String[] args) {

Monster fly = new FlyMonster("ムックル");

fly.intro();

Monster swim = new SwimMonster("コイキング");

swim.intro();

}

}

# **問題５：ポリモーフィズム**

|  |
| --- |
| <<interface>>  ***IFairy*** |
|  |
| *intro() : void* |

|  |
| --- |
| **Drakness** |
|  |
| intro() : void |

|  |
| --- |
| **Fire** |
|  |
| intro() : void |

|  |
| --- |
| **Light** |
|  |
| intro() : void |

Light・Darkness・Fireの各クラスはIFairyを実装（継承）している。

**＜設問5-1＞**

IFairyインターフェイスを作成せよ。

**＜設問5-2＞**

**実行結果5**のようになるようにLight・Darkness・Fireの各妖精クラスとJ2Test13\_5クラス（mainメソッド）を作成せよ。

**実行結果5（J2Test13\_5クラス）**

妖精を召喚して自己紹介させます！

誰を召喚しますか？（0：光の妖精、1：闇の妖精、2：炎の妖精、-1：やめる）＞**0**

わたしは光の妖精！この者に祝福を！！

誰を召喚しますか？（0：光の妖精、1：闇の妖精、2：炎の妖精、-1：やめる）＞**1**

わたしは闇の妖精だ！闇の力を思い知れ！！

誰を召喚しますか？（0：光の妖精、1：闇の妖精、2：炎の妖精、-1：やめる）＞**2**

わたしは炎の妖精さ！炎の力は気まぐれなのさ！！

誰を召喚しますか？（0：光の妖精、1：闇の妖精、2：炎の妖精、-1：やめる）＞**-1**