

\_\_\_\_\_点 / 104点

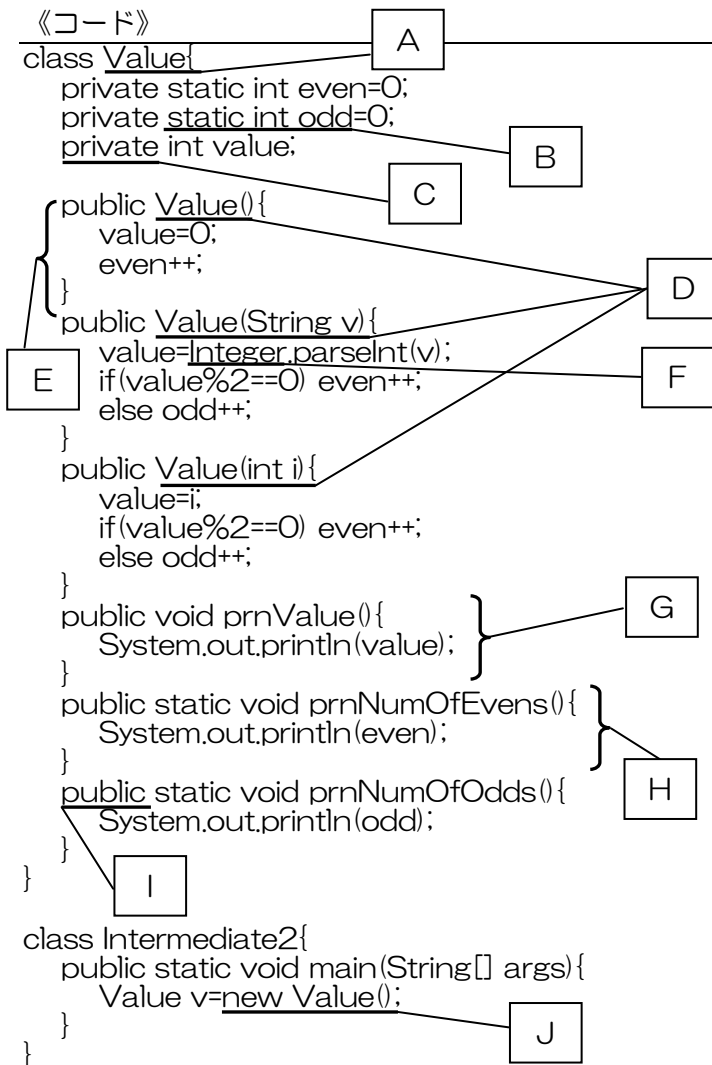
【問1】次はJavaに関する記述である。各記述が正しい場合は○を、誤っている場合は×を解答欄に記入しなさい。【各2点 計50点】

- (1) クラスはフィールドのみをもつことができる
- (2) メソッドの戻り値は常に int 型である
- (3) メソッドの戻り値は return 文を用いて返す
- (4) メソッドの引数は0個でもよい
- (5) 実引数に変数を指定する場合、仮引数の変数の名前を同じにする必要がある
- (6) クラスの protected メンバは、そのクラスからのみアクセス可能なメンバである
- (7) クラスは private メンバと public メンバを必ず同時に持たなければならない
- (8) クラスのメソッドのみが public メンバになることができる
- (9) クラスのカプセル化はメソッドのオーバーロードにより実現する
- (10) コンストラクタの名前はクラス名と同じである
- (11) コンストラクタはオーバーロードができない
- (12) コンストラクタは仮引数を持たない
- (13) クラス変数とはクラス単位で準備される変数である
- (14) クラスメソッド内ではインスタンス変数をアクセスできない
- (15) インスタンス変数はオブジェクト毎に異なる値を保持できる
- (16) ラップクラスとは、変数の基本型に関する様々な便利機能を集約したクラスである
- (17) クラス型の変数は基本型である
- (18) 参照型変数はオブジェクトへの参照を保持する
- (19) 参照型変数はメソッドの仮引数にはできない
- (20) 参照されなくなったオブジェクトはガーベッジコレクションによりメモリから削除される
- (21) 既存のクラスを拡張して新しいクラスを宣言することはできない
- (22) 新しいクラスが既存のクラスのメンバを受け継ぐことをオーバーロードという
- (23) 継承における既存のクラスをサブクラスという
- (24) コンストラクタは継承されない
- (25) コンストラクタは必ず明示的に1つ以上宣言しなければならない

【問2】次はJavaで用いる用語である。それぞれの用語に対応する箇所を下のコード中から1つずつ選び記号を解答欄に記入しなさい。【各2点 計24点】

- (1) インスタンスメソッドの宣言
- (2) クラス名
- (3) private メンバの宣言のための修飾子
- (4) public メンバの宣言のための修飾子
- (5) オーバーロード
- (6) クラス変数の宣言
- (7) クラスメソッドの宣言
- (8) デフォルトコンストラクタ
- (9) ラップクラス
- (10) 引数なしのコンストラクタの宣言
- (11) オーバーロードされた他のコンストラクタの呼び出し

(12) オブジェクトの生成



【問3】次の各文に対応するメソッドの宣言を下の選択肢より記号で答えなさい。複数の宣言が対応する場合はすべて答えなさい。また、1つも対応しない場合は×を解答欄に記入しなさい。【各2点 計10点】

- (1) 引数を1つもち、戻り値がないメソッド
- (2) 引数がなく、戻り値が double 型のメソッド
- (3) 引数を2つもちメソッド
- (4) 引数を1つもち、戻り値が double 型のメソッド
- (5) オーバーロードで文法エラーとなる一組

《選択肢》

- (a) void func(int i){...}
- (b) int func(){...}
- (c) double func(int i, double j){...}
- (d) void func(double i){...}
- (e) double func(){...}

【問4】次の各コードを実行したときの画面出力を正確に答えなさい。クラス Value の宣言は問2のものとし、クラスA~C の宣言は下に示すものとする。【各2点 計20点】

```
(1) class Intermediate4_1 {
    public static void main(String[] args) {
        Value v=new Value();
        v.prnValue();
    }
}
```

- (2) class Intermediate4\_2 {
 public static void main(String[] args) {
 Value v=new Value("3");
 v.prnValue();
 }
 }
- (3) class Intermediate4\_3 {
 public static void main(String[] args) {
 Value v=new Value(0);
 Value w=v;
 v=new Value(2);
 w.prnValue();
 }
 }
- (4) class Intermediate4\_4 {
 public static void main(String[] args) {
 Value.prnNumOfEvens();
 }
 }
- (5) class Intermediate4\_5 {
 public static void main(String[] args) {
 Value v=new Value(3);
 Value w=new Value(5);
 Value.prnNumOfOdds();
 }
 }
- (6) class Intermediate4\_6 {
 public static void main(String[] args) {
 A a=new A();
 a.func();
 }
 }
- (7) class Intermediate4\_7 {
 public static void main(String[] args) {
 A a=new A();
 a.func(3,5);
 }
 }
- (8) class Intermediate4\_8 {
 public static void main(String[] args) {
 B b=new B();
 }
 }
- (9) class Intermediate4\_9 {
 public static void main(String[] args) {
 C c=new C();
 }
 }
- (10) class Intermediate4\_10 {
 public static void main(String[] args) {
 C c=new C(3);
 }
 }

《コード》

```
class A {
    public void func() {
        System.out.println("A0");
    }
    public void func(int i) {
        func();
        System.out.println("A1");
    }
    public void func(double d) {
        this.func();
        System.out.println("A2");
    }
}
```

```
class B {
    public B() {
        System.out.println("B0");
    }
    public B(int i) {
        this();
        System.out.println("B1");
    }
}

class C extends B {
    public C() {
    }
    public C(int i) {
        super(i);
    }
}
```

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)	(9)	(10)
(11)	(12)			

【問3】

(1)	(2)	(3)
(4)	(5)	と

【問4】

(1)	(2)	(3)	(4)
(5)	(6)	(7)	(8)
(9)	(10)		

お疲れ様でした

◆お知らせ◆

1組の皆さんへ 明日火曜日の講義はお休みです