

1. 次の文の空欄に適切な語句を語群より記号で選びなさい。

(3 点 × 5 問 = 15 点)

オブジェクト指向プログラミングとは、相互に (1) を送り合う (2) の集まりとしてプログラムを構成する技法である。オブジェクト指向プログラミング言語の一種である Java 言語では、(1) の送信を次のように記述する。

(3) . (4) ((5))

語群：(あ) クラス、(い) メソッド、(う) 変数、(え) インスタンス、(お) 引数、(か) メッセージ、(き) フィールド、(く) メンバー、(け) オブジェクト

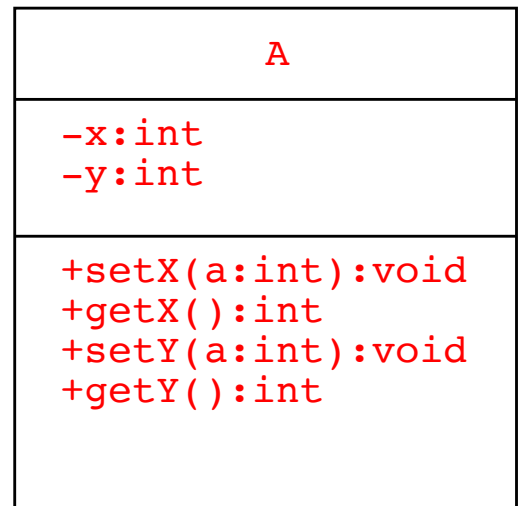
(1)	か	(2)	け	(3)	え	(4)	い	(5)	お
-----	---	-----	---	-----	---	-----	---	-----	---

```
// C.java
public class C {
    private B p1 = new B(1,2,3);
    private B p2 = new B(4,5,6);
    C(int x1,int y1, int z1,
        int x2,int y2, int z2) {
        p1 = new B(x1,y1,z1);
        p2 = new B(x2,y2,z2);
    }
    public void print() {
        System.out.printf(
            "(%d,%d,%d)-(%d,%d,%d)\n",
            p1.getX(),p1.getY(),p1.getZ(),
            p2.getX(),p2.getY(),p2.getZ());
    }
}
```

2. 次の三つの Java プログラムをよく読んで解答しなさい。

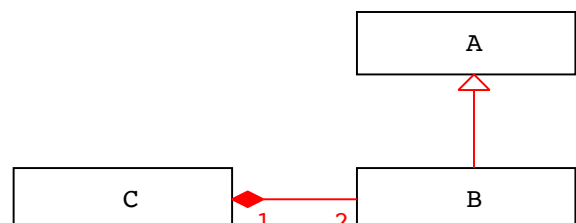
```
// A.java
public class A {
    private int x;
    private int y;
    public void setX(int a) { x = a; }
    public int getX() { return x; }
    public void setY(int a) { y = a; }
    public int getY() { return y; }
}
```

```
// B.java
public class B extends A {
    private int z;
    B(int a, int b, int c) {
        setX(a);
        setY(b);
        z = c;
    }
    public void setZ(int a) { z = a; }
    public int getZ() { return z; }
}
```



(1) A のクラス図を完成しなさい。(5 点)

(2) 次のクラス図に関連を追記しなさい。(5 点)

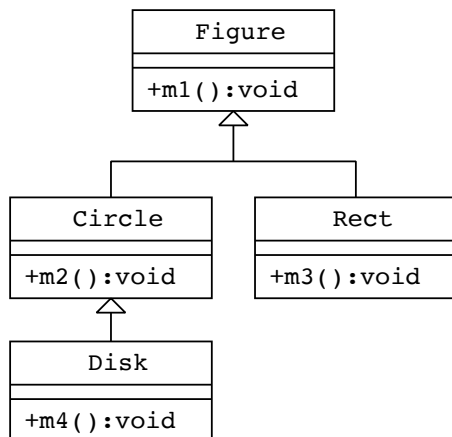


(3) クラス C は 3 次元空間上の半直線を表現しています。クラス C に追加する、直線を x 軸方向に移動するメソッドを書きなさい。(5 点)

(追加メソッド:「public void moveX(int dx);」直線を dx だけ x 軸方向に移動する。)

```
void moveX(int dx) {
    p1.setX(p1.getX()+dx);
    p2.setX(p2.getX()+dx);
}
```

3. 次のクラス図とプログラムをよく見て問に答えなさい。



```
Figure f;
Circle c;
Rect r;
Disk d;
```

```
Figure x;
Circle y;
```

(1) 変数 x に型変換しないで代入できる全ての変数を列挙しなさい。(但し、x、y は除きます。)(5 点)

f, c, r, d

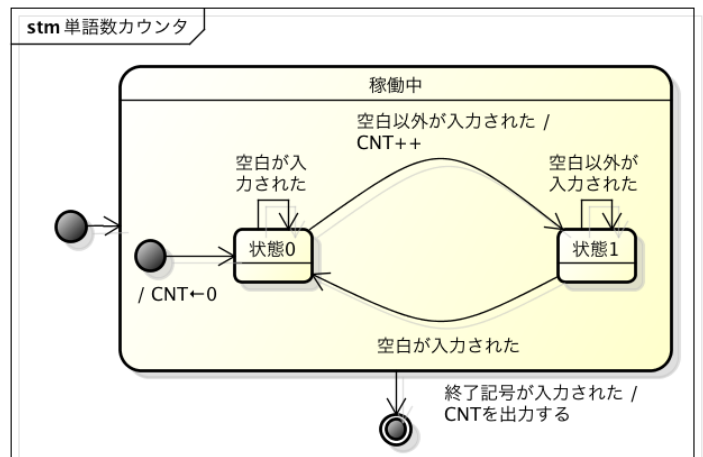
(2) 変数 y に型変換しないで代入できる全ての変数を列挙しなさい。(但し、x、y は除きます。)(5 点)

c, d

(3) 次のプログラムでエラーになるものに「×」、エラーにならないものに「○」をつけなさい。(2 点 × 12 問 = 24 点)

番号	プログラム	○/×
(a)	f.m1();	○
(b)	f.m2();	×
(c)	f.m3();	×
(d)	f.m4();	×
(e)	c.m1();	○
(f)	c.m2();	○
(g)	c.m3();	×
(h)	c.m4();	×
(i)	d.m1();	○
(j)	d.m2();	○
(k)	d.m3();	×
(l)	d.m4();	○

4. 次の図は入力された文字列に含まれる単語数をカウントするステートマシンを表しています。入力は 1 文字ずつステートマシンへ入力され、それをトリガーにステートが変化します。入力文字列に含まれる文字は空白(' '), 英字('A' ~ 'Z'), 終了記号('#') のみとします。単語とは空白で区切られた英字列です。図をよく見て以下の問に答えなさい。



(1) 上記のステートマシンを Fsm クラスとして実装します。下のプログラムを完成しなさい。(20 点)

```
public class Fsm {  
    private int state = 0;           // ステートマシンの状態  
    private int cnt    = 0;           // 単語数のカウンタ  
  
    public void input(char ch) { // ステートマシンへ文字を入力  
        if (ch=='#') {  
            System.out.println(cnt);  
            state = -1;           // -1 は終了状態のつもり  
        } else {  
  
            switch (state) {  
            case 0:  
                if (ch!=' ') {  
                    cnt++;  
                    state = 1;  
                }  
                break;  
            case 1:  
                if (ch==' ') {  
                    state = 0;  
                }  
                break;  
            }  
        }  
    }  
}
```

5. 次のプログラムをよく見て問に答えなさい。(16 点)

以下は、授業で作成した電卓プログラムの Eclipse が生成した、initialize メソッド内部の一部です。ボタン「1」を生成したあと、アクションリスナーを登録しています。なお、keisan は Keisan クラスのインスタンスであり、電卓の計算機能 (モデル) を実装しています。

```
 JButton btn1 = new JButton("1");
 btn1.addActionListener(new ActionListener() {
     public void actionPerformed(ActionEvent e) {
         keisan.keyIn('1');
     }
 });
 panel.add(btn1);
```

同様の記述が 16 個のキーについて繰返され無駄が多いように見えます。そこで、次のように使用できる DentakuButton クラスを作ります。下に、DentakuButton クラスを完成しなさい。なお、import で始まる行は省略して良い。

```
DentakuButton btn1 = new DentakuButton('1', keisan);
panel.add(btn1);
```

```
public class DentakuButton extends JButton {
```

```
    private char moji;
```

```
    private Keisan keisan;
```

```
    public DentakuButton(char c, Keisan k) {
```

```
        super(c+"");
```

```
        moji = c;
```

```
        keisan = k;
```

```
        this.addActionListener(new ActionListener() {
```

```
            public void actionPerformed(ActionEvent e) {
```

```
                keisan.keyIn(moji);
```

```
            }
```

```
        });
```

```
    }
```

```
}
```