

学籍番号_____ 氏名_____

問題1 Q1、Q2、Q3、Q4、Q5 の実行結果を記述してください（80点）。ただし、Q3 から Q5 は Template Method パターンで設計されています。

<pre>public class Q00 { public static void main(String[] args) { Item item = new C(); item.back(); System.out.println(item.getCharactor()); Item[] is = {new C(), new TL()}; is[1].front(); System.out.print(is[0].getCharactor() + is[1].getCharactor()); } }</pre>		<pre>> java Q00 × 青</pre>
<pre>public class C implements Item { private String state = null; public C() { this.interior(); } public void front() { this.state = "o"; } public void interior() { this.state = " "; //全角の空白 } public void back() { this.state = "x"; } public String getCharactor() { return this.state; } }</pre>	<pre>public class TL implements Item { private String state = null; public TL() { this.interior(); } public void front() { this.state = "青"; } public void interior() { this.state = "黄"; } public void back() { this.state = "赤"; } public String getCharactor() { return this.state; } }</pre>	<pre>public interface Item { /** * 表にします */ public void front(); /** * 中間にします */ public void interior(); /** * 裏にします */ public void back(); /** * 文字列を得る */ public String getCharactor(); }</pre>

Q1 の実行結果

<pre>public class Q1 { public static void main(String[] args) { Item[] is = {new C(), new TL(), new C()}; for(int i = 0; i < is.length/*3*/; i++) { is[i].front(); System.out.print(is[i].getCharactor()); } } }</pre>	<pre>public interface Display { /* 縦の行数を得る */ public int height(); /* 横の列数を得る */ public int width(); /* 横x番目、縦y番目のItemオブジェクトを得る */ public Item get(int x, int y); /* 次の状態に移行する */ public void next(); /* 状態の番号を返却する(0, 1, 2の何れかを返却) */ public int getIndex(); /* 表示する */ public void show(); }</pre>
---	---

Q2の実行結果

<pre>public class Q2 { public static void main(String[] args) { Display d = new SDisplay(); d.show(); d.next(); d.show(); d.next(); d.show(); } }</pre>	<pre>public abstract class AbstractDisplay implements Display { private int index = 0; public abstract int height(); public abstract int width(); public abstract Item get(int x, int y); public void next() { this.index = (this.index + 1) % 3; } public int getIndex() { return this.index; } public void show() { for(int y = 0; y < this.height(); y++) { for(int x = 0; x < this.width(); x++) { Item item = this.get(x, y); System.out.print(item.getCharactor()); } System.out.println(); } } }</pre>
<pre>public class SDisplay extends AbstractDisplay { private Item item = new TL(); public SDisplay() { this.item.interior(); } public int height() { return 1; } public int width() { return 1; } public Item get(int x, int y) { if(this.getIndex() == 0) { this.item.front(); } else if(this.getIndex() == 1) { this.item.interior(); } else if(this.getIndex() == 2) { this.item.back(); } return this.item; } }</pre>	

Q3の実行結果

```
public class Q3 {
    public static void main(String[] args) {
        Display d = new YDisplay();
        d.next();
        d.show();
    }
}
```

Q4の実行結果

```
public class Q4 {
    public static void main(String[] args) {
        Display d = new WDisplay();
        d.show();
    }
}
```

```
public class WDisplay extends AbstractDisplay {
    private Item[] line0 = {new C(), new C(), new C()};
    private Item[] line1 = {new C(), new C(), new C()};
    private Item[] line2 = {new C(), new C(), new C()};
    public WDisplay() {
        this.line0[1].back();
        this.line1[0].back();
        this.line1[2].back();
        this.line2[1].back();
    }
    public int height() {
        return 3;
    }
    public int width() {
        return 3;
    }
    public Item get(int x, int y) {
        if(y == 0) {
            return this.line0[x];
        } else if(y == 1) {
            return this.line1[x];
        } else if(y == 2) {
            return this.line2[x];
        }
        return null;
    }
}
```

Q5の実行結果

```
public class Q5 {
    public static void main(String[] args) {
        Display d = new TDisplay(new YDisplay());
        d.next();
        d.show();
    }
}
```

```
public class YDisplay extends AbstractDisplay {
    private Item item = new TL();
    public YDisplay() {
        this.item.front();
    }
    public int height() {
        return 3;
    }
    public int width() {
        return 2;
    }
    public Item get(int x, int y) {
        if(this.getIndex() == 0) {
            item.front();
        } else if(this.getIndex() == 1) {
            item.interior();
        } else if(this.getIndex() == 2) {
            item.back();
        }
        return this.item;
    }
}
```

```
public class TDisplay implements Display {
    private Display display;
    public TDisplay(Display display) {
        this.display = display;
    }
    public int height() {
        return this.display.width();
    }
    public int width() {
        return this.display.height();
    }
    public Item get(int x, int y) {
        return this.display.get(y, x);
    }
    public void next() {
        this.display.next();
    }
    public int getIndex() {
        return this.display.getIndex();
    }
    public void show() {
        for(int y = 0; y < this.height(); y++) {
            for(int x = 0; x < this.width(); x++) {
                Item item = this.get(x, y);
                System.out.print(item.getCharactor());
            }
            System.out.println();
        }
    }
}
```

問題2 Q00 および Q1 から Q5を除くクラス間の関係のみ記述してください（20点）。必要なクラスは予め印刷してあります。ただし、状態、振る舞い（メソッド）は省略してください。

学籍番号_____ 氏名_____

問題 1 の解答、実行結果はマスに一文字を記述すること。ただし、全角の空白はなににも記述しないこと。

Q1の実行結果

Q2の実行結果

[illegible][illegible]

Q3の実行結果

Q4の実行結果

[illegible][illegible]

Q5の実行結果

[illegible]

問題2の解答

<<interface>> Item

<<interface>> Display

C

TL

<i>AbstractDisplay</i>

YDisplay

SDisplay

WDisplay

TDisplay