

# Java 类使用

## [作业要求]:

- 提交形式：通过给定的二维码提交 pdf 作业报告；
- 提交截止时间：2022 年 11 月 1 日零点之前；
- 作业过时不补!!!
- 提交作业二维码：



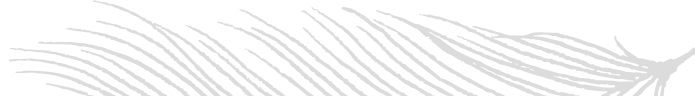
## [报告要求]:

- 需要有封面（封面形式按照思源学堂上指定的文件）；
- 必须是 pdf 文件，否则扣分。


## 题目 1：创建一个用来表示时间的类

创建一个用来存储时间数据的 MyTime 类类型，当创建完这个类以后，应该保证下面的程序可以运行，且运行的结果如下图所示。

```
1. package homework2;
2. public class TestTime {
3.     public static void main(String[] args) {
4.         MyTime t1 = new MyTime();
5.         MyTime t2 = new MyTime(2);
6.         MyTime t3 = new MyTime(21,34);
7.         MyTime t4 = new MyTime(12, 25, 42);
8.         MyTime t5 = new MyTime(t4);
9.
10.        System.out.println("Constructed with:");
11.        System.out.println("t1: all arguments defaulted");
12.        System.out.printf("    %s\n", t1.toUniversalString());
13.        System.out.printf("    %s\n", t1.toString());
```



```
14.
15. System.out.println("t2: hour specified; minute and second defaulted");
16. System.out.printf("    %s\n", t2.toUniversalString());
17. System.out.printf("    %s\n", t2.toString());
18.
19. System.out.println("t3: hour and minute specified; second defaulted");
20. System.out.printf("    %s\n", t3.toUniversalString());
21. System.out.printf("    %s\n", t3.toString());
22.
23. System.out.println("t4: hour ,minute and second specified");
24. System.out.printf("    %s\n", t4.toUniversalString());
25. System.out.printf("    %s\n", t4.toString());
26.
27. System.out.println("t5: MyTime object t4 specified");
28. System.out.printf("    %s\n", t5.toUniversalString());
29. System.out.printf("    %s\n", t5.toString());
30.
31. //when initialize t6 with invalid values,please output error information
32. MyTime t6 = new MyTime(15, 74, 99);
33. System.out.println("t6: invalid values");
34. System.out.printf("%s\n", t6.toUniversalString());
35. }
36. }
```



```
Constructed with:
t1: all arguments defaulted
    00:00:00
    12:00:00 AM
t2: hour specified; minute and
second defaulted
    02:00:00
    02:00:00 AM
t3: hour and minute specified;
second defaulted
    21:34:00
    09:34:00 PM
t4: hour ,minute and second
specified
    12:25:42
    12:25:42 PM
t5: MyTime object t4 specified
    12:25:42
    12:25:42 PM
t6: invalid values
minute must be 0-59
second must be 0-59
```

另外，再为 MyTime 类增加三个成员方法：

```
public void incrementHour();  
public void incrementMinute();  
public void incrementSecond();
```

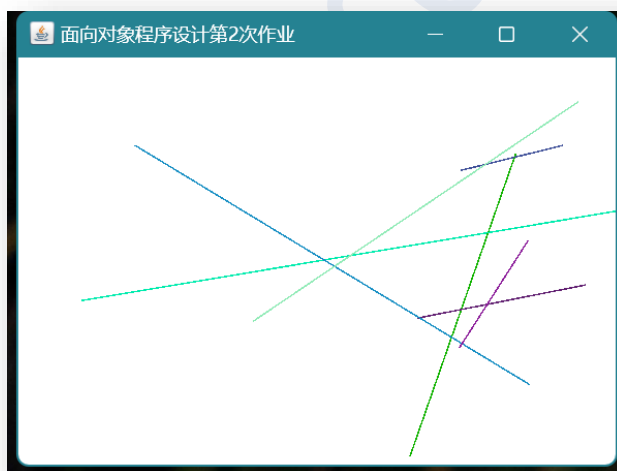
每个方法都是在原有的对应数据上进行加 1 操作，但是一定要注意这种对时间的加 1 操作的合法性，比如考虑如下场景：

MyTime 对象中的 second 值为 59，此时调用 incrementSecond()方法之后，second 值会成为 0，同时 minute 值也应该加 1。

将所有的特殊情况都考虑全面，并给出测试类。

## 题目 2：创建自己的 GUI 图形

打开 homework.zip 文件，在 homework2 的文件夹下面有 5 个类：TestDraw、DrawPanel、MyLine、MyCircle 和 MyRectangle 类。其中 TestDraw 类、DrawPanel 类和 MyLine 类里面的内容都已经写好。创建一个 Java 工程，将 homework2 文件夹拷贝到新建的工程的 src 文件夹下，执行 TestDraw 程序，此时应该能够显示如下运行结果：



在阅读 TestDraw、DrawPanel 和 MyLine 类定义的基础上，为 MyCircle 和 MyRectangle 添加相应的代码，以便让 TestDraw 程序可以绘制圆形和长方形。

**本题第 1 个任务：**创建上面提到的两个类型，包括必要的该类型的数据成员、构造方法（可以利用重载技术，使该类型具有多种构造能力）、以及必不可少的一个成员方法：

```
public void draw(Graphics g) { //.... }
```

上面成员方法中的参数 g 为类型 java.awt.Graphics，使用这个类来绘制所需要的图形，该类型的使用可以在 JDK API 中查找。

**注意：**

- DrawPanel 的构造函数中提供了对 MyCircle 和 MyRectangle 类型数组的支持。

- 尽可能为 `MyCircle` 类和 `MyRectangle` 类提供多样的构造函数，体会重载以及理解构造函数的意义（构造函数之间通过 `this` 传递的方式也要多练）。

当完成上面的任务之后，就应该能够正常绘制长方形和圆形了。同学们应该注意到这样的代码结构：在 `DrawPanel` 类型的代码中，是利用 `switch` 语句判断图形的形状从而决定该调用哪种类型的 `draw` 函数，这种代码的复用率极低。不妨考虑如下场景：如果新增一种图形形状 `MyOval`，这个时候想一想在 `DrawPanel` 类型中都需要修改什么代码？这种修改在增加一种新的图形类型时是无法避免的。

如何能够减少 `DrawPanel` 类型对于增加新的图形类型而作的代码修改工作呢？使用面向对象中的继承和多态技术就可以做到。注意观察：`DrawPanel` 类型中有一个 `paintComponent` 方法，该方法主要的功能就是调用图形类型的 `draw` 函数，用以完成对该图形的绘制工作。

**本题第 2 个任务：**大刀破斧的修改 `DrawPanel` 类的数据成员、构造函数等，使其具有可以接收绘制任何（包括已有的和未来的）具有相同行为的图形类型。为了达到代码复用的目的，可能修改的不止 `DrawPanel` 类。

在实验报告中书写本题的设计环节部分时，请使用 UML 的类图进行阐述。

## 题目 3：接口的威力

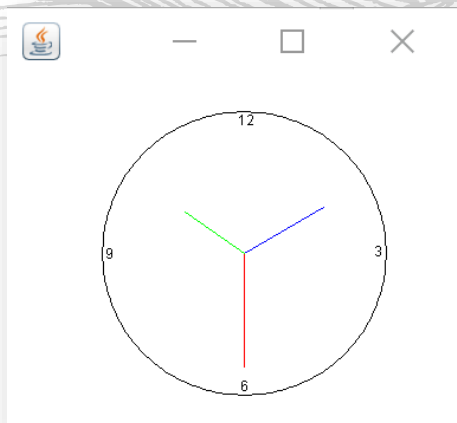
题目 2 中的图形类型的行为能力都是从绘制角度考虑的，但是一个图形类型除了具有被绘制的特点外，还有几何特征，比如面积。解决这个问题看似很简单：为每个图形类型增加一个求解面积的成员方法就可以了。但是会发现：并不是所有能够绘制的图形都具有求解面积的行为，比如 `MyLine`。那如何能够让应该具有面积特征的图形有求解面积的行为，不具有面积特征的图形就不具有求解面积的行为，同时还不破坏它们具有可以被绘制的多态效果的现状呢？

Java 的接口类型可以助力完成上面的目标。

**本题第 1 个任务：**请为该题目修改题目 2 中的图形类型，并绘制出新的图形类型之间的结构图。

### 扩展内容：再大胆一点...

如何让 `DrawPanel` 类型可以绘制更多的东西（并不局限于图形类型的对象）？怎么修改 `DrawPanel` 中的数据成员类型以及构造方法？怎么定义类型使其能在 `DrawPanel` 上进行绘制？**本题第 2 个任务：**依然尝试使用接口，比如创建了一个 `MyTime` 类型（这个类型第 1 个作业中被创建过），在 `DrawPanel` 上就可以绘制一个根据 `MyTime` 对象中的数据所表现出来的一个时钟。（`DrawPanel` 原有的绘制其他图形的能力应该依然保持。）如下图所示：



## 想让时钟活起来吗？（尝试即可）

如果想让时钟活起来，需要使用 Java 的多线程技术。

在保证了本题前半部分能够正确绘制出静态时钟图案的情况下，只需要在主函数代码的框架基础上，加入如下的代码片段即可（假设生成的时钟类型的对象名为 myTimer，此部分在作业中不做任何要求）：

```
Thread t = new Thread() {
    public void run() {
        while (true) {
            try {
                Thread.sleep(1000); //设置绘制的时间间隔为 1 秒
            } catch (InterruptedException e) {
                System.err.println(e);
            }
            //正好也可以用来检验第 1 道题目中时钟类型的这个方法是否正确
            myTimer.incrementSecond();
            //更新绘制图形面板上的内容（也就是绘制的图像）
            panel.updateUI();
        }
    }
};
t.start();
```

## 题目 4：撰写继承、多态和接口方面的知识梳理

代码编写是对理论知识的一种实践。能够编写出正确、合理、有效的代码，理论知识必须得扎实。每个同学都需要沉下心来认真阅读参考资料、课件，再结合课堂所讲和课下所练，总结面向对象中继承、接口以及多态的理解和认识。