案例1：画窗体

2022年7月5日 8:34

一、方法二：封装思想创建窗体

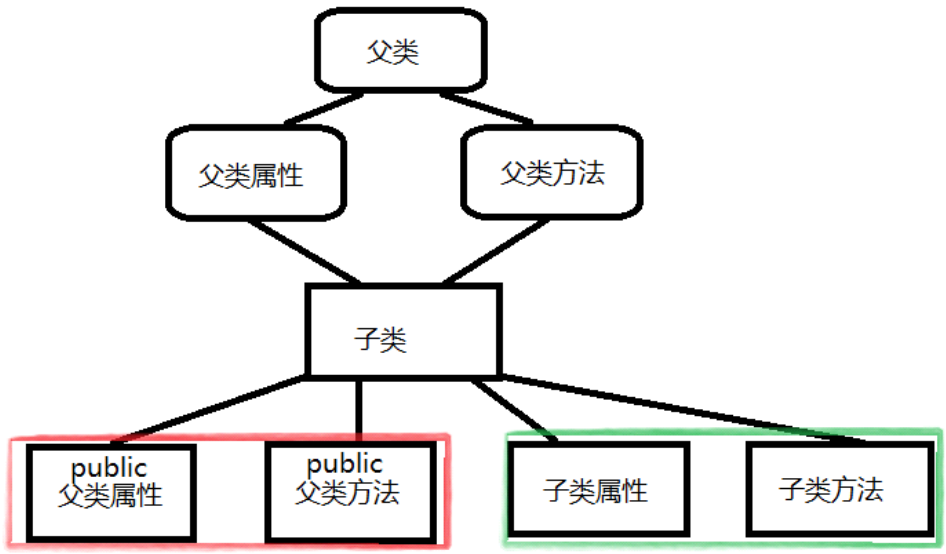
1. 普通方法：类——JFrame类型的属性——showMe方法中完成JFrame的实例化
   a. 调用：main方法，创建当前类的对象，并通过对象名.方法名()调用方法，完成窗体的创建
2. 构造方法：类——JFrame类型的属性-成员变量——构造方法中完成JFrame的实例化
   a. 调用：main方法，创建当前类的对象，隐含着系统自动调用当前类的无参构造方法
   b. 特点：
      i. 和当前类同名
      ii. 没有返回值类型
      iii. 在通过new关键字创建对象时，默认调用类的无参构造；例如new MyFrameBCons()
      iv. 开发者未定义任何构造方法时，系统会默认提供一个无参构造；也即public Type(){}
      v. 通常在构造方法中完成对属性的初始化；也即需要在类实例化的同时完成初始化的属性则在构造方法中实现
      vi. 有参构造：属于构造方法的重写，并且一旦定义了有参构造，系统就不会再提供默认的无参构造了；也即一旦自行定义了构造方法，则需要同时定义一个无参构造以供类实例化时调用

二、方法三：继承JFrame类

1. 创建一个MyFrameC的类继承JFrame类，在MyFrameC类中创建一个showMe(MyFrameC mfc)方法，在此方法中设置MyFrameC 窗体的属性（大小、标题、默认关闭），最后在main方法中实例化MyFrameC类的对象，通过对象调用showMe方法，传入MyFrameC 类对象，实现窗体的设置。

```
class MyFrameC  extends JFrame{
    public void showMe(MyFrameC  mfc){
        //设置大小
    }
    public static void main(String []args){
        MyFrameC  mfc = new MyFrameC();
        mfc.showMe(mfc);
    }
}
```

2. 继承：继承可以提升代码的重用性，子类继承父类，继承自什么就是什么类；子类会继承父类的所有公开的属性和方法



3. this关键字：代表着当前对象
    a. 可以通过this. 调用当前对象的方法以及继承自父类的方法
    b. 可以通过this.属性 调用当前对象的属性
    c. 可以通过this()~=MyFrameC()调用当前对象的构造方法
    d. 可以通过this(val)调用当前对象的有参构造
    e. this()一定要在构造方法的方法体的第一行，并且不能在当前构造方法中调用自身构造
    f. 常用于this.成员变量，与方法中同名的局部变量进行区分
4. 总结：
    a. 如何使用一个已经存在的类：通过import关键字导入此类所在的包；快捷键：ctrl+1 || ctrl+shift+O
    b. 如何创建对象：ClassType ct = new ClassType();
    c. 如果访问对象的方法：
        i. 在方法所在的类中调用该方法：this.methodName();
        ii. 在方法所在的类外部调用该方法：先生成对应类的对象，然后再通过对象.methodName()方式调用
    d. 继承：通过extends关键字来继承一个已经存在类，子类继承父类
    e. this关键字：当前对象，调用方法、属性、构造方法
    f. 构造方法：与类同名且无返回值类型，可以重写；通常用于实例化同时初始化属性，在创建对象时被自动调用

案例2：画卡通画-星星

2022年7月5日 11:11

一、案例需求

1. 创建一个图形化的窗口，并把它显示在界面上，需要设置大小、标题、默认关闭操作等。同时将窗体的颜色设置为黑色，然后在窗口中显示一个黄色的星星。
2. 三种方式：
   a. 使用绘制字符串 * 来实现，显示星星
   b. 使用绘制图形的方式来画星星：

   c. 使用在窗口中显示一张星星图片的方式
3. 分析：
   a. 先有窗体，设置大小、背景颜色、标题、默认关闭
   b. 再在窗体上设置一张画布
   c. 在画布上绘制星星：
      i. 方式一：在画布上设法显示出键盘输入的 '*'
      ii. 方式二：在画布上画一个实心的黄色小圆，然后在圆心周围绘制一些直线
      iii. 方式三：在画布上显示出星星图片
4. 具体实现步骤以及代码：
   a. 创建一个星星窗体类：MyStarFrame

```
public class MyStarFrame extends JFrame{
    public void showMe() {
        this.setSize(600, 600);
        this.setTitle("星星窗体");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }
    public static void main(String[] args) {
        MyStarFrame msf = new MyStarFrame();
        msf.showMe();
    }
}
```

   b. Java不允许直接在窗体类上作画的，需要提供一个画布并添加到窗体上；此时同样Java类库提供好了画布类JPanel，如何将系统提供的画布类转换为所需要的画布类；通过继承的方式来产生一个有星星的新画布。创建一个画布类：MyStarPanel

```
public class MyStarPanel extends JPanel{
    public void paint(Graphics g) {
```

```
        }
    }
```

i. 此处的paint(Graphics g)不是随意写的方法，而是在继承了父类的基础上，对父类或者祖父类上的方法进行重写；重写发生在父子类之中，子类中重写后的方法会覆盖掉继承自父类的原有方法

c. 完成窗体对象&画布对象的创建以及将画布添加到窗体上

```
//          创建窗体对象
        MyStarFrame frame = new MyStarFrame();
//          创建画布对象
        MyStarPanel panel = new MyStarPanel();
//          将画布对象添加到窗体上
        frame.add(panel);
//          显示窗体以及画布
        frame.showMe();
```

d. 第一种方式绘制星星：

```
    public void paint(Graphics g) {
//          super调用父类的paint(g)，绘制一个空画布
        super.paint(g);
//          将画布背景颜色设置为黑色
        this.setBackground(Color.black);
//          设置要绘制的图案的颜色为黄色
        g.setColor(Color.yellow);
//          绘制字符串形式的星星：可以设置字体
        Font font = new Font("宋体",Font.BOLD,30);
        g.setFont(font);
//          绘制 *
        g.drawString("*", 40, 40);
//          四个参数分别表示起点&终点的坐标
        g.drawLine(30, 30, 100, 100);
    }
```

5. 知识点讲解：

a. super关键字：代表父类对象，可以通过super调用父类的公开的属性以及方法，通过super()调用父类的构造方法，需要位于整个代码块的第一行，不要出现嵌套调用的情况

b. static关键字：

i. 一个类不论产生多少个对象，静态的只有一份；

ii. 静态的与对象无关，是否有对象或对象有多少个都不影响静态的数量；

iii. 静态的成员变量才被称为类变量

iv. 可以理解为什么有些变量是使用类名来调用的，此时此变量为静态变量

v. 静态变量会随着类的加载而加载到内存中
vi. 修饰方法为静态方法、修饰变量为静态成员变量-类变量（不能修饰局部变量！）