# 第5次平时作业

package work5;  
  
/\* Apple类，重写equals(),hashcode(),toString()方法 \*/  
public class Apple {  
 // 只加入了一个重量属性  
 int weight;  
  
 // 构造函数  
 public Apple(int weight){  
 this.weight = weight;  
 }  
  
 /\* 重写的equals()方法\*/  
 @Override  
 public boolean equals(Object obj) {  
 /\* 先判断是否为本身，  
 \* 再判断是否指向为空，  
 \* 然后判断是否属于Apple类的实例，比较weight值  
 \* 上述方法可以增加安全性，提高效率\*/  
 if(this == obj){  
 return true;  
 }  
 if(obj == null){  
 return false;  
 }  
 if(obj instanceof Apple apple){  
 return this.weight == apple.weight;  
 }  
  
 return false;  
 }  
  
 /\* 重写的hashCode()方法，借助特殊素数31进行运算\*/  
 @Override  
 public int hashCode() {  
 int code = 17;  
 code = 31\*this.weight + code;  
 return code;  
 }  
  
 /\* 重写的toString()方法\*/  
 @Override  
 public String toString() {  
 return "this is an Apple and its hashcode: "+this.hashCode()+", besides its wight is "+this.weight;  
 }  
}

package work5;  
  
public class AppleTest {  
  
 public static void main(String[] args) {  
 /\* 实例化5个Apple对象\*/  
 Apple apple1 = new Apple(100);  
 Apple apple2 = new Apple(200);  
 Apple apple3 = new Apple(300);  
 Apple apple4 = new Apple(400);  
 Apple apple5 = new Apple(400);  
  
 /\* 进行方法校验\*/  
 System.*out*.println(apple1.toString());  
 System.*out*.println("apple2's weight: "+apple2.weight+" and apple3's weight: "+apple3.weight);  
 System.*out*.println("are apple2 == apple3 ? "+apple2.equals(apple3));  
 System.*out*.println("apple4's weight is: "+apple4.weight+" and apple5's weight is: "+apple5.weight);  
 System.*out*.println("apple4 == apple5 ? "+apple4.equals(apple5));  
 }  
}