1. **程序阅读题，阅读程序并手写给出程序输出，然后上机验证，提交作业时应将手写答案照片放入word文档**

1、

**public** **class** ArrayManipulator {

**private** **int**[] arr;

**public** ArrayManipulator(**int**[] arr) {

**this**.arr = arr;

}

**public** **void** manipulateArray() {

**for** (**int** i = 0; i < arr.length; i++) {

arr[i] += (i % 2 == 0) ? ++arr[i] : --arr[i];

}

}

**public** **void** printArray() {

**for** (**int** i = 0; i < arr.length; i++) {

System.***out***.print(arr[i] + " ");

}

System.***out***.println();

}

**public** **static** **void** main(String[] args) {

**int**[] nums = {1, 2, 3, 4, 5};

ArrayManipulator manipulator = **new** ArrayManipulator(nums);

System.***out***.print("Initial Array: ");

manipulator.printArray();

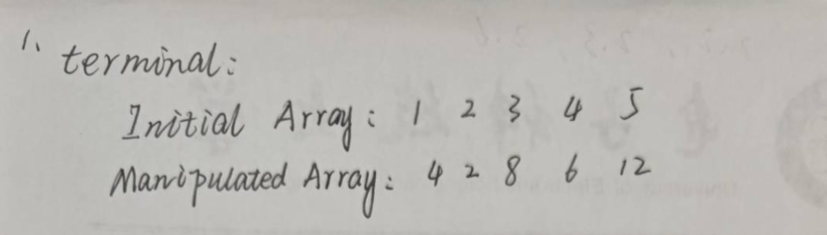
manipulator.manipulateArray();

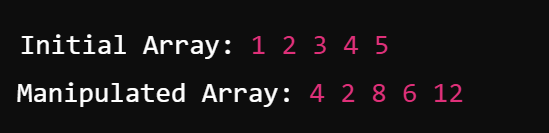
System.***out***.print("Manipulated Array: ");

manipulator.printArray();

}

}





2、注意，本题main方法有一句有问题，需要先改错，然后给出输出

interface Printer {

void print();

}

class BasePrinter implements Printer {

public void print() {

System.out.println("BasePrinter printing...");

}

}

class AdvancedPrinter extends BasePrinter {

public void print() {

System.out.println("AdvancedPrinter printing...");

}

public void advancedPrint() {

System.out.println("Advanced printing features enabled.");

}

}

public class PrinterTest {

public static void main(String[] args) {

Printer p1 = new BasePrinter();

Printer p2 = new AdvancedPrinter();

BasePrinter p3 = new AdvancedPrinter();

p1.print();

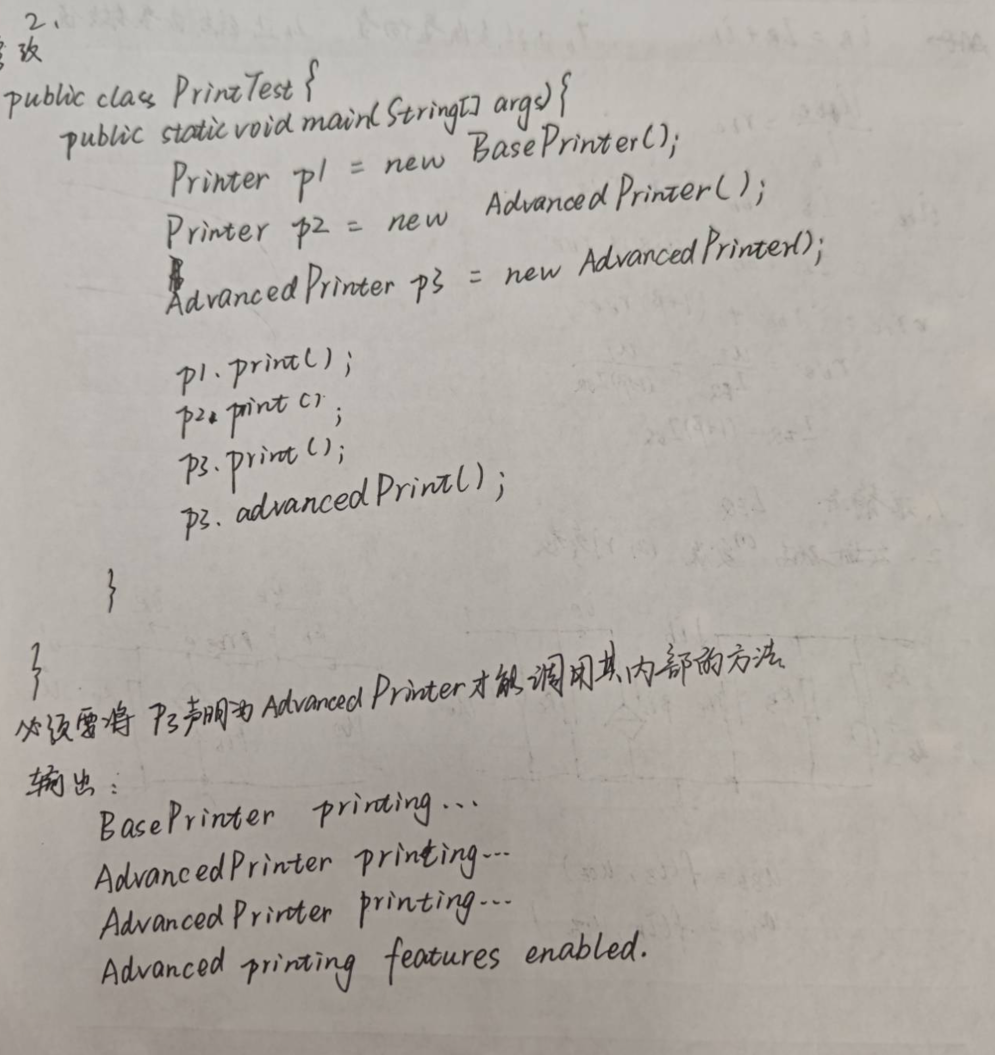
p2.print();

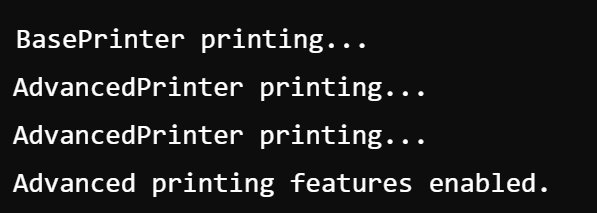
p3.print();

p3.advancedPrint();

}

}





3、

class MyCustomException extends Exception {

public MyCustomException(String message) {

super(message);

}

}

class Calculator {

public int divide(int a, int b) throws MyCustomException {

if (b == 0) {

throw new MyCustomException("Division by zero is not allowed.");

}

return a / b;

}

}

public class ExceptionTest {

public static void main(String[] args) {

Calculator calculator = new Calculator();

try {

System.out.println("Result 1: " + calculator.divide(10, 2));

System.out.println("Result 2: " + calculator.divide(5, 0));

} catch (MyCustomException e) {

System.out.println("Caught an exception: " + e.getMessage());

} finally {

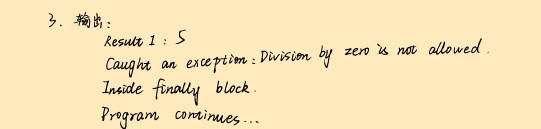
System.out.println("Inside finally block.");

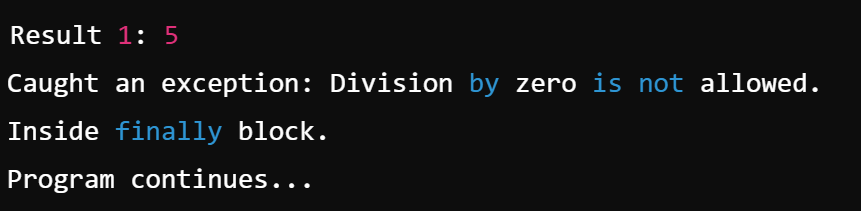
}

System.out.println("Program continues...");

}

}





4、

**class** Counter {

**private** **int** count = 0;

**public** **synchronized** **void** increment() {

count++;

System.***out***.println(Thread.*currentThread*().getName() + " incremented count to: " + count);

}

**public** **int** getCount() {

**return** count;

}

}

**class** Worker **implements** Runnable {

**private** Counter counter;

**public** Worker(Counter counter) {

**this**.counter = counter;

}

**public** **void** run() {

**for** (**int** i = 0; i < 3; i++) {

counter.increment();

}

}

}

**public** **class** ThreadSyncTest {

**public** **static** **void** main(String[] args) {

Counter counter = **new** Counter();

Thread t1 = **new** Thread(**new** Worker(counter), "Thread 1");

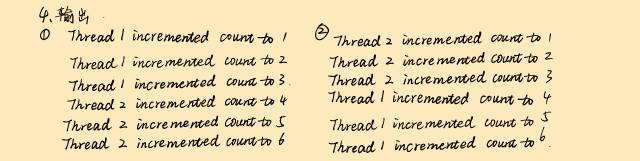
Thread t2 = **new** Thread(**new** Worker(counter), "Thread 2");

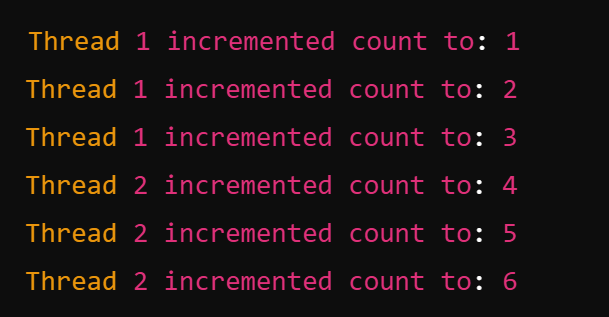
t1.start();

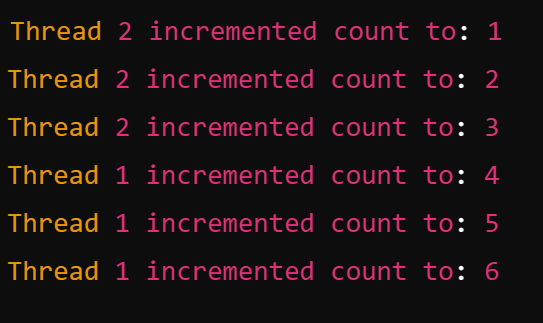
t2.start();

}

}





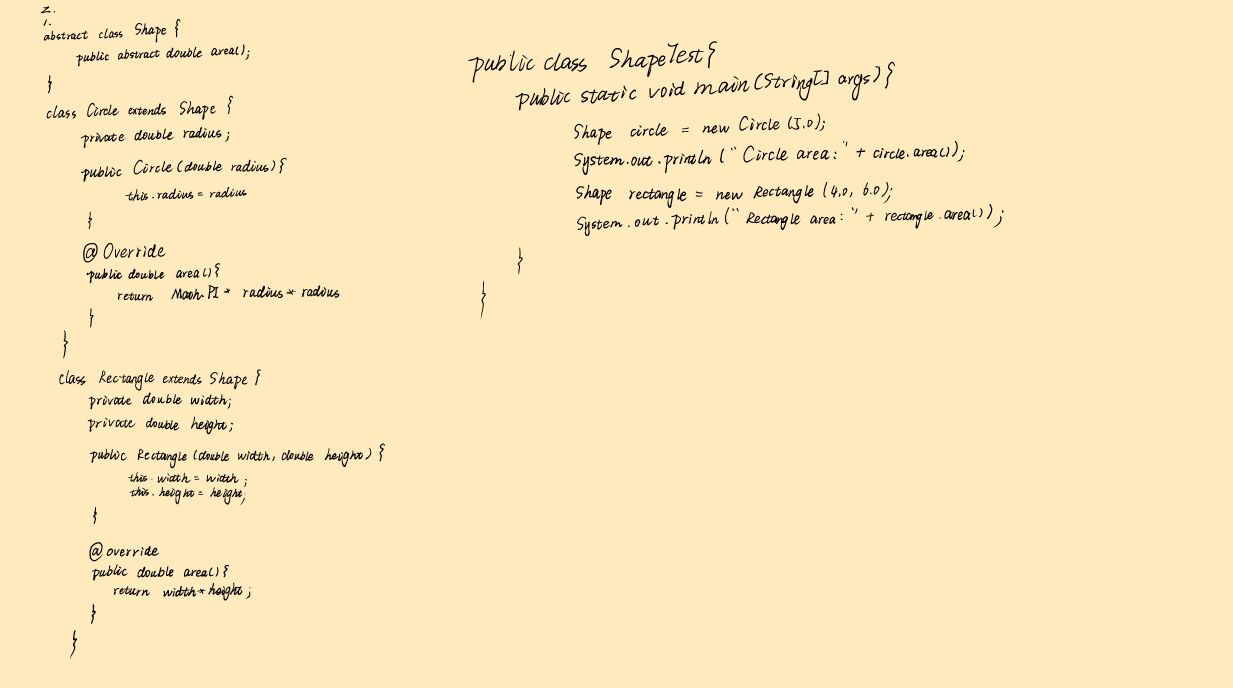


**二、程序设计题，要求在纸上手写程序并将照片贴入word文档，然后上机验证设计程序的正确性**

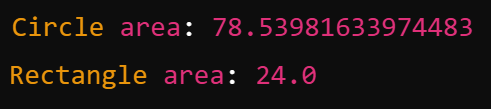
1、

设计一个包含类和继承的程序，要求：

* 定义一个父类 Shape，其中包含一个抽象方法 area()，用于计算形状的面积。
* 定义两个子类 Circle 和 Rectangle，分别继承 Shape 类并实现 area() 方法。
* 在主程序中创建不同类型的形状对象，并输出它们的面积。



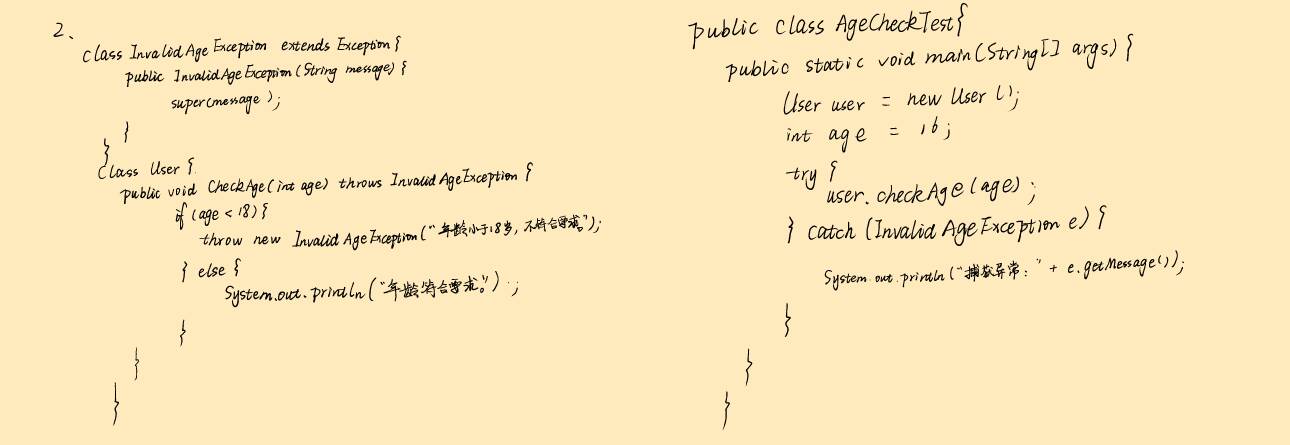
|  |
| --- |
| // 定义抽象父类 Shape  abstract class Shape {  // 抽象方法，用于计算面积  public abstract double area();  }  // 定义子类 Circle，继承自 Shape  class Circle extends Shape {  private double radius;  // 构造方法，初始化半径  public Circle(double radius) {  this.radius = radius;  }  // 实现父类的 area() 方法，计算圆的面积  @Override  public double area() {  return Math.PI \* radius \* radius;  }  }  // 定义子类 Rectangle，继承自 Shape  class Rectangle extends Shape {  private double width;  private double height;  // 构造方法，初始化宽度和高度  public Rectangle(double width, double height) {  this.width = width;  this.height = height;  }  // 实现父类的 area() 方法，计算矩形的面积  @Override  public double area() {  return width \* height;  }  }  // 主程序类  public class ShapeTest {  public static void main(String[] args) {  // 创建 Circle 对象  Shape circle = new Circle(5.0);  System.out.println("Circle area: " + circle.area());  // 创建 Rectangle 对象  Shape rectangle = new Rectangle(4.0, 6.0);  System.out.println("Rectangle area: " + rectangle.area());  }  } |



2、

设计一个包含自定义异常的程序，要求：

* 定义一个自定义异常 InvalidAgeException，当用户输入的年龄小于 18 岁时抛出异常。
* 定义一个类 User，其中包含一个检查年龄的方法 checkAge(int age)，若年龄小于 18 岁，则抛出自定义异常。
* 在主程序中捕获该异常并输出相应的错误信息。



|  |
| --- |
| // 定义自定义异常 InvalidAgeException  class InvalidAgeException extends Exception {  public InvalidAgeException(String message) {  super(message);  }  }  // 定义 User 类  class User {  // 检查年龄的方法  public void checkAge(int age) throws InvalidAgeException {  if (age < 18) {  throw new InvalidAgeException("年龄小于 18 岁，不符合要求。");  } else {  System.out.println("年龄符合要求。");  }  }  }  // 主程序类  public class AgeCheckTest {  public static void main(String[] args) {  User user = new User();  int age = 16; // 可以改变此值测试不同情况  try {  user.checkAge(age);  } catch (InvalidAgeException e) {  System.out.println("捕获异常: " + e.getMessage());  }  }  } |

年龄输入小于18：



年龄输入大于等于18：

