1.

import java.util.ArrayList;

import java.util.List;

public class Course {

private String ClassName;

private int ClassNumber;

private List<Integer> PreClassList;

public Course() {

PreClassList = new ArrayList<>();

}

public Course(String className, int classNumber, List<Integer> preClassList) {

ClassName = className;

ClassNumber = classNumber;

PreClassList = preClassList;

}

public String getClassName() {

return ClassName;

}

public void setClassName(String className) {

ClassName = className;

}

public int getClassNumber() {

return ClassNumber;

}

public void setClassNumber(int classNumber) {

ClassNumber = classNumber;

}

public List<Integer> getPreClassList() {

return PreClassList;

}

public void setPreClassList(List<Integer> preClassList) {

PreClassList = preClassList;

}

public void printClassMessage() {

System.out.println("课程名：" + ClassName);

System.out.println("课程编号：" + ClassNumber);

System.out.println("预先课程编号：");

for (Integer pre:

PreClassList){

System.out.println(pre + " ");

}

System.out.println();

}

}

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) {

Course c1 = new Course(

"睡觉艺术", 114514,

new ArrayList<Integer>(){{add(1);add(12);add(123);}});

Course c2 = new Course();

List<Integer> list = new ArrayList<>();

list.add(12);

list.add(33);

list.add(44);

c2.setClassName("药水艺术");

c2.setClassNumber(123);

c2.setPreClassList(list);

c1.printClassMessage();

c2.printClassMessage();

}

}

2.

public interface Shape {

double Area();

}

public class Circle implements Shape {

private double area = 0, r = 0;

public Circle(double r) {

this.r = r;

}

private void cacuArea() {

area = r \* r \* Math.PI;

}

@Override

public double Area() {

cacuArea();

return area;

}

}

public class Square implements Shape {

double area = 0, x = 0;

public Square(double x) {

this.x = x;

}

private void cacuArea() {

area = x \* x;

}

@Override

public double Area() {

cacuArea();

return area;

}

}

public class Triangle implements Shape {

double area = 0, board = 0, height = 0;

public Triangle(double board, double height) {

this.board = board;

this.height = height;

}

private void cacuArea() {

area = (board \* height) / 2.0;

}

@Override

public double Area() {

cacuArea();

return area;

}

}

3.

public class MyException extends Exception {

public MyException(String message) {

super(message);

}

}

public class Main {

public static void main(String[] args) throws MyException {

int x = 0;

if (x == 0) {

throw new MyException("1");

}

}

}