|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学校** | 吉林大学珠海学院 |  | **班级** | **软件7班** |  | **科目** | JAVA面向对象程序设计 |
| **学院** | 计算机学院 |  | **姓名** | 张子恒 |  | **日期** | 2023.\*.\* |
| **专业** | 软件工程 |  | **学号** | 04222119 |  | **实验名称** |  |

|  |
| --- |
| **实验内容**  第1题  代码（要有必要的注释说明程序思路）  package com.ex12;   public class MyThread extends Thread {  private String name;  private int count;   public MyThread(String name) {  this.name = name;  this.count = 0;  }   @Override  public void run() {  while (count < 5) {  count++;  System.*out*.println("你好，我的名字是" + name + "，第" + count + "次来这儿");  }  } }  package com.ex12;  public class test01 {  public static void main(String[] args) {  MyThread thread1 = new MyThread("张三");  MyThread thread2 = new MyThread("李四");   thread1.start();  thread2.start();  } }  package com.ex12;  public class MyRunnable implements Runnable {  private String name;  private int count;   public MyRunnable(String name) {  this.name = name;  this.count = 0;  }   @Override  public void run() {  while (count < 5) {  count++;  System.*out*.println("你好，我的名字是" + name + "，第" + count + "次来这儿");  }  } }  package com.ex12;  public class test02 {  public static void main(String[] args) {  MyRunnable runnable1 = new MyRunnable("张三");  MyRunnable runnable2 = new MyRunnable("李四");   Thread thread1 = new Thread(runnable1);  Thread thread2 = new Thread(runnable2);   thread1.start();  thread2.start();  }  }  运行结果截图      第2题  代码（要有必要的注释说明程序思路）  package com.ex12;  public class test03 {  public static void main(String[] args) throws InterruptedException {  Student student1 = new Student("student1", 10);  Student student2 = new Student("student2", 60);  Teacher teacher = new Teacher();   student1.start();  student2.start();   // 等待student1和student2线程启动  Thread.*sleep*(1000);   teacher.start();   // 等待teacher线程完成  teacher.join();  } }  class Student extends Thread {  private String name;  private int sleepTime;   public Student(String name, int sleepTime) {  this.name = name;  this.sleepTime = sleepTime;  }   @Override  public void run() {  try {  System.*out*.println(name + "准备睡觉" + sleepTime + "分钟");  Thread.*sleep*(sleepTime \* 1000);  System.*out*.println(name + "已经睡了" + sleepTime + "分钟，可以开始上课了");  } catch (InterruptedException e) {  e.printStackTrace();  }  } }  class Teacher extends Thread {   @Override  public void run() {  System.*out*.println("上课！");  Thread student1Thread = findThread("student1");  if (student1Thread != null) {  student1Thread.interrupt();  }  Thread student2Thread = findThread("student2");  if (student2Thread != null) {  student2Thread.interrupt();  }  }   private Thread findThread(String name) {  ThreadGroup group = Thread.*currentThread*().getThreadGroup();  while (group.getParent() != null) {  group = group.getParent();  }  Thread[] threads = new Thread[group.activeCount()];  group.enumerate(threads);  for (Thread thread : threads) {  if (thread != null && thread.getName().equals(name)) {  return thread;  }  }  return null;  } }  运行结果截图    第3题  代码  package com.ex12;  public class test04 {  public static void main(String[] args) {  Warehouse warehouse = new Warehouse();  Loader loader = new Loader(warehouse);  Driver driver = new Driver(loader);   driver.start();  } }  class Driver extends Thread {  private Loader loader;   public Driver(Loader loader) {  this.loader = loader;  }   @Override  public void run() {  try {  System.*out*.println("运货司机正在等待装运工完成工作");  loader.join();  System.*out*.println("运货司机开始运货");  } catch (InterruptedException e) {  e.printStackTrace();  }  } }  class Loader extends Thread {  private Warehouse warehouse;   public Loader(Warehouse warehouse) {  this.warehouse = warehouse;  }   @Override  public void run() {  try {  System.*out*.println("装运工正在等待仓库管理员打开仓库");  warehouse.open();  System.*out*.println("装运工开始装运货物");  } catch (InterruptedException e) {  e.printStackTrace();  }  } }  class Warehouse {  private boolean isOpen = false;   public synchronized void open() throws InterruptedException {  while (!isOpen) {  wait();  }  }   public synchronized void unlock() {  isOpen = true;  notifyAll();  } }  运行结果截图    第4题  代码  运行结果截图  第5题  代码  运行结果截图 |
| **小结** |