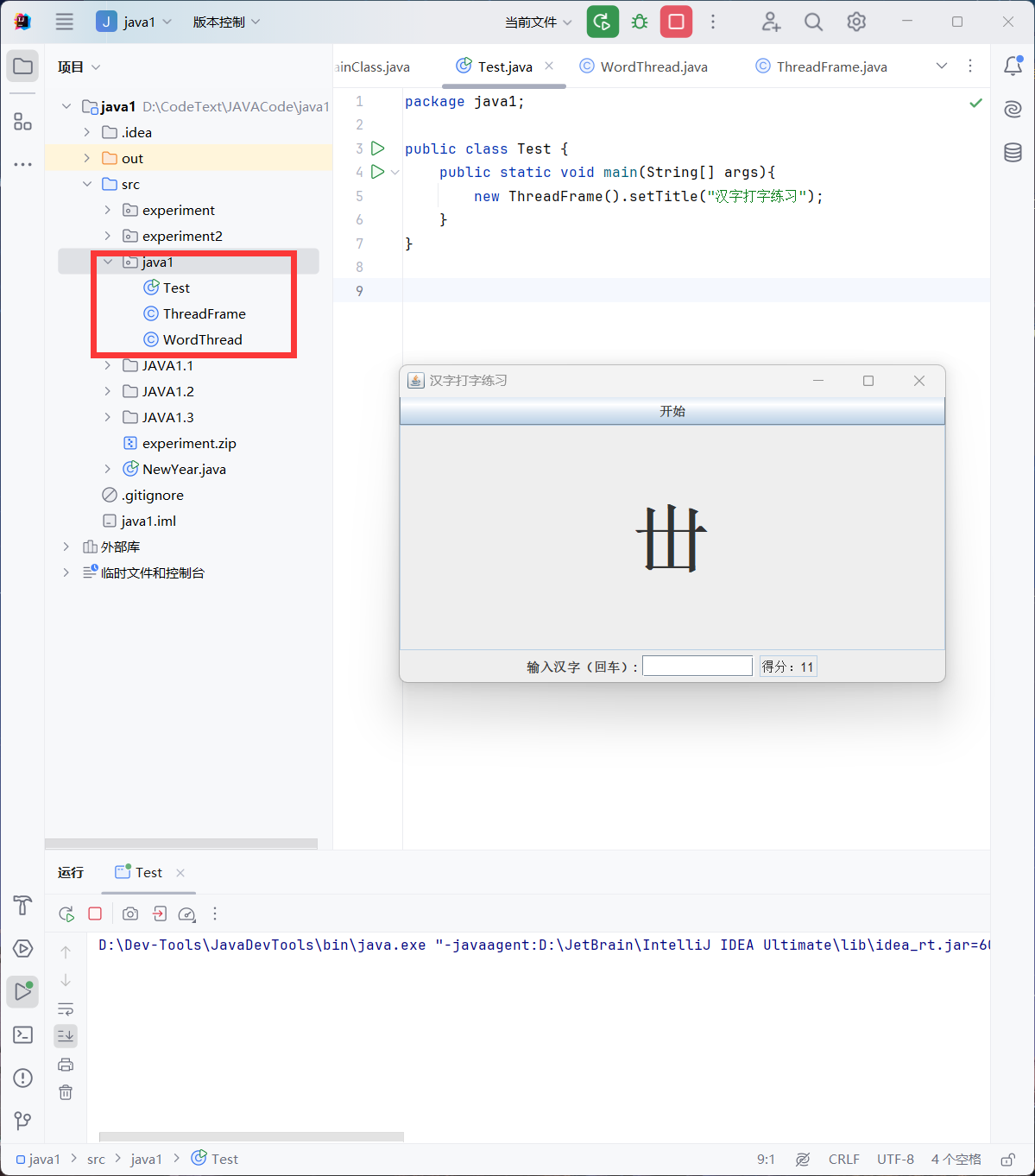
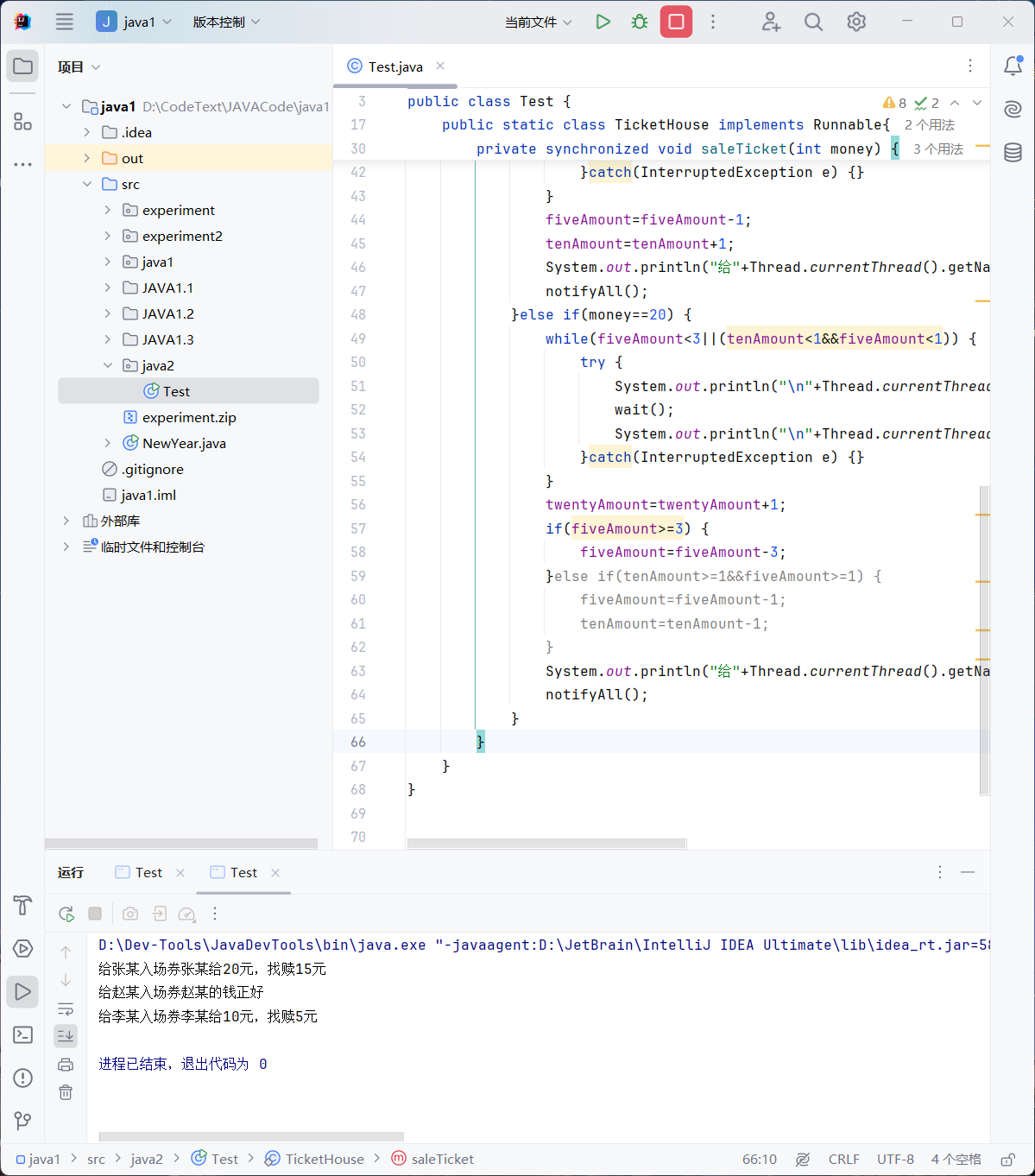
package java1;  
  
public class Test {  
 public static void main(String[] args){  
 new ThreadFrame().setTitle("汉字打字练习");  
 }  
}

package java1;  
  
import javax.swing.\*;  
  
public class WordThread extends Thread {  
 char word;  
 int startPosition = 19968;  
 int endPosition = 32320;  
 JTextField showWord;  
 int sleepLength=6000;  
 public void setJTextField(JTextField t){  
 showWord=t;  
 showWord.setEditable(false);  
 }  
 public void setSleepLength(int n){  
 sleepLength=n;  
 }  
 public void run(){  
 int k=startPosition;  
 while (true){  
 word=(char) k;  
 showWord.setText(""+word);  
 try {  
 Thread.*sleep*(sleepLength);*//代码1* }catch (InterruptedException e){}  
 k++;  
 if (k>=endPosition){  
 k=endPosition;  
 }  
 }  
 }  
}

package java1;  
  
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
*//主线程*public class ThreadFrame extends JFrame implements ActionListener {  
 JTextField showWord;  
 JButton button;  
 JTextField inputText,showScore;  
 WordThread giveWord;*//代码2* JLabel ID=new JLabel();  
 int score=0;  
 ThreadFrame(){  
 showWord=new JTextField(6);  
 showWord.setFont(new Font("",Font.*BOLD*,72));  
 showWord.setHorizontalAlignment(JTextField.*CENTER*);  
 giveWord=new WordThread();*//代码3* giveWord.setJTextField(showWord);  
 giveWord.setSleepLength(5000);  
 button=new JButton("开始");  
 inputText=new JTextField(10);  
 showScore=new JTextField(5);  
 button.addActionListener(this);  
 inputText.addActionListener(this);  
 add(button,BorderLayout.*NORTH*);  
 add(showWord,BorderLayout.*CENTER*);  
 showScore.setEditable(false);  
 JPanel southP=new JPanel();  
 southP.add(new JLabel("输入汉字（回车）:"));  
 southP.add(inputText);  
 southP.add(showScore);  
 add(southP,BorderLayout.*SOUTH*);  
 setBounds(100,100,350,180);  
 setVisible(true);  
 validate();  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 }  
  
 @Override*//重写监视器* public void actionPerformed(ActionEvent e) {  
 *//按钮监视器* if (e.getSource()==button){  
 if(!(giveWord.isAlive())){  
 giveWord=new WordThread();*//代码4* giveWord.setJTextField(showWord);  
 giveWord.setSleepLength(5000);  
 }  
 try{  
 giveWord.start();*//代码5* }catch (Exception exe){  
 }  
 }  
 else if (e.getSource()==inputText){  
 if (inputText.getText().equals(showWord.getText())){  
 score++;  
 }  
 showScore.setText("得分："+score);  
 inputText.setText(null);  
 }  
 }  
}



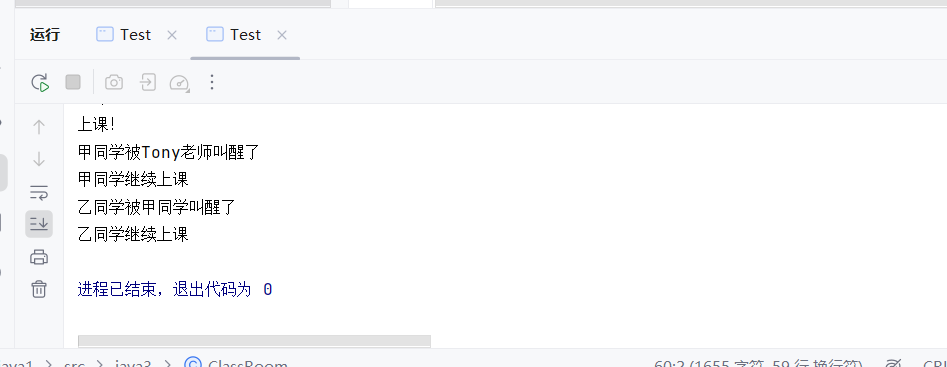
package java2;  
  
public class Test {  
 public static void main(String args[]) {  
 TicketHouse officer = new TicketHouse();  
 Thread zhangmo,limo,zhaomo;  
 zhangmo = new Thread(officer);  
 zhangmo.setName("张某");  
 limo = new Thread(officer);  
 limo.setName("李某");  
 zhaomo = new Thread(officer);  
 zhaomo.setName("赵某");  
 zhangmo.start();  
 limo.start();  
 zhaomo.start();  
 }  
 public static class TicketHouse implements Runnable{  
 int fiveAmount=3,tenAmount=0,twentyAmount=0;  
 public void run() {  
 if(Thread.*currentThread*().getName().equals("张某")) {  
 saleTicket(20);  
 }else if(Thread.*currentThread*().getName().equals("李某")) {  
 saleTicket(10);  
 }  
 else if(Thread.*currentThread*().getName().equals("赵某")) {  
 saleTicket(5);  
 }  
  
 }  
 private synchronized void saleTicket(int money) {  
 if(money==5) {  
 fiveAmount=fiveAmount+1;  
 System.*out*.println("给"+Thread.*currentThread*().getName()+"入场券"+Thread.*currentThread*().getName()+"的钱正好");  
  
 notifyAll();  
 }else if(money==10) {  
 while(fiveAmount<1) {  
 try {  
 System.*out*.println("\n"+Thread.*currentThread*().getName()+"靠边等...");  
 wait();  
 System.*out*.println("\n"+Thread.*currentThread*().getName()+"继续买票");  
 }catch(InterruptedException e) {}  
 }  
 fiveAmount=fiveAmount-1;  
 tenAmount=tenAmount+1;  
 System.*out*.println("给"+Thread.*currentThread*().getName()+"入场券"+Thread.*currentThread*().getName()+"给10元，找赎5元");  
 notifyAll();  
 }else if(money==20) {  
 while(fiveAmount<3||(tenAmount<1&&fiveAmount<1)) {  
 try {  
 System.*out*.println("\n"+Thread.*currentThread*().getName()+"靠边等...");  
 wait();  
 System.*out*.println("\n"+Thread.*currentThread*().getName()+"继续买票");  
 }catch(InterruptedException e) {}  
 }  
 twentyAmount=twentyAmount+1;  
 if(fiveAmount>=3) {  
 fiveAmount=fiveAmount-3;  
 }else if(tenAmount>=1&&fiveAmount>=1) {  
 fiveAmount=fiveAmount-1;  
 tenAmount=tenAmount-1;  
 }  
 System.*out*.println("给"+Thread.*currentThread*().getName()+"入场券"+Thread.*currentThread*().getName()+"给20元，找赎15元");  
 notifyAll();  
 }  
 }  
 }  
}



12

package java3;  
  
public class Test {  
  
 public static void main(String[] args) {  
 *// TODO Auto-generated method stub* ClassRoom classroom = new ClassRoom();  
 classroom.Teacher.setName("Tony老师");  
 classroom.Student1.setName("甲同学");  
 classroom.Student2.setName("乙同学");  
 classroom.Student1.start();  
 classroom.Student2.start();  
 classroom.Teacher.start();  
 }  
}

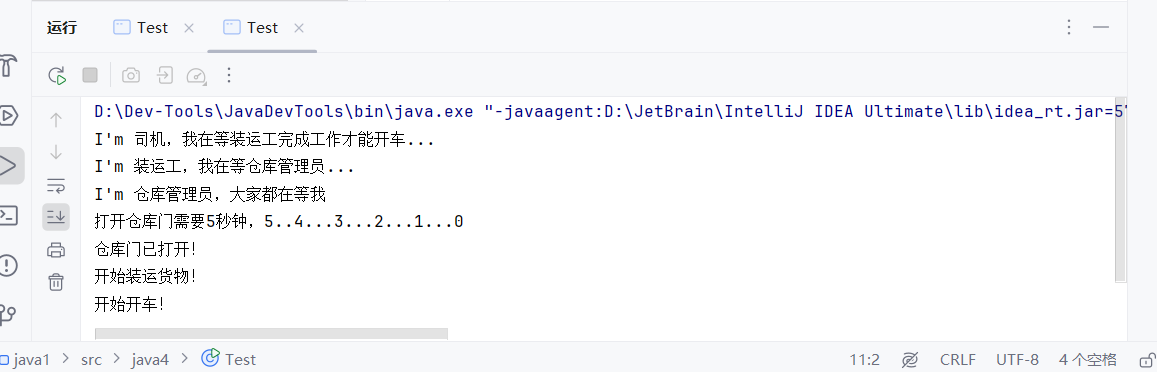
package java3;  
  
public class ClassRoom implements Runnable{  
 Thread Student1,Student2,Teacher;  
 ClassRoom()  
 {  
 Teacher=new Thread(this);  
 Student1=new Thread(this);  
 Student2=new Thread(this);  
 }  
 public void run()  
 {  
 String name=Thread.*currentThread*().getName();  
 if(name.equals("甲同学"))  
 {  
 try  
 {  
 System.*out*.println("我是"+name+"，我在上课,想睡10分钟再开始上课");  
*// System.out.println("想睡10分钟再开始上课");* Thread.*sleep*(1000\*10\*60);  
 }  
 catch(InterruptedException e)  
 {  
 System.*out*.println(name+"被Tony老师叫醒了");  
 }  
 Student2.interrupt();  
 System.*out*.println(name+"继续上课");  
 }  
 else if(name.equals("乙同学"))  
 {  
 try  
 {  
 System.*out*.println("我是"+name+"，我在上课,想睡1小时再开始上课");  
 *//System.out.println("想睡1小时再开始上课");* Thread.*sleep*(1000\*60\*60);  
 }  
 catch(InterruptedException e)  
 {  
 System.*out*.println(name+"被甲同学叫醒了");  
 Student2.interrupt();  
 }  
*// attachThread.interrupt();* System.*out*.println(name+"继续上课");  
 }  
 else if(name.equals("Tony老师"))  
 {  
 for(int i=1;i<=3;i++)  
 {  
 System.*out*.println("上课！");  
 try  
 {  
 Thread.*sleep*(1000);  
 }  
 catch(InterruptedException e) {}  
 }  
*// attachThread.interrupt();* Student1.interrupt();  
 }  
 }  
}



13

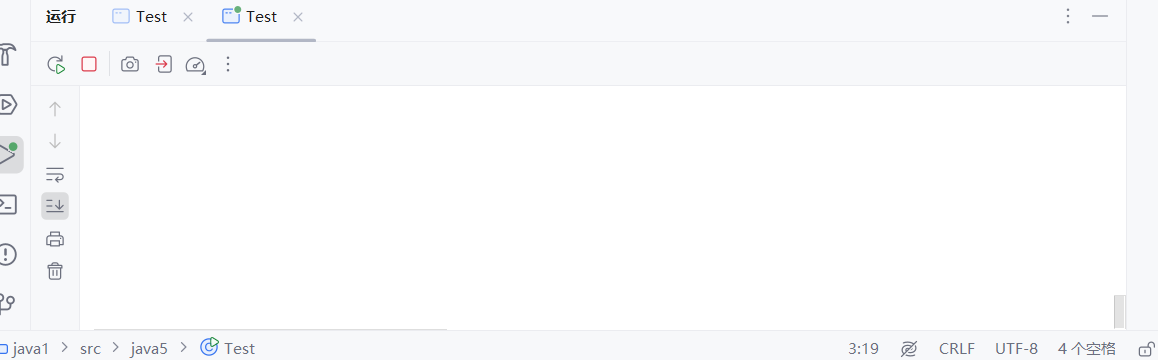
package java4;  
  
public class Test {  
  
 public static void main(String[] args) {  
 Squard squard = new Squard();  
 squard.driver.start();  
  
 }  
}

package java4;  
  
public class Squard implements Runnable{  
 Thread shipper;*//装运工* Thread driver;*//司机* Thread stockman;*//仓库管理员* Squard()  
 {  
 shipper=new Thread(this);  
 driver=new Thread(this);  
 stockman=new Thread(this);  
 shipper.setName("装运工");  
 driver.setName("司机");  
 stockman.setName("仓库管理员");  
 }  
 public void run()  
 {  
 if(Thread.*currentThread*().getName().equals("司机"))  
 {  
 System.*out*.println("I'm "+Thread.*currentThread*().getName()+"，我在等装运工完成工作才能开车...");  
 try  
 {  
 shipper.start();*//线程开始* shipper.join();*//线程联合* }  
 catch(InterruptedException e)  
 {  
 e.printStackTrace();  
 }  
 System.*out*.println("开始开车！");  
 }  
 else if(Thread.*currentThread*().getName().equals("装运工"))  
 {  
 System.*out*.println("I'm "+Thread.*currentThread*().getName()+"，我在等仓库管理员...");  
 try  
 {  
 stockman.start();*//线程开始* stockman.join();*//线程联合* }  
 catch(InterruptedException e)  
 {  
 e.printStackTrace();  
 }  
 System.*out*.println("开始装运货物！");  
 }  
 else if(Thread.*currentThread*().getName().equals("仓库管理员"))  
 {  
 System.*out*.println("I'm "+Thread.*currentThread*().getName()+"，大家都在等我");  
 System.*out*.println("打开仓库门需要5秒钟，5..4...3...2...1...0");  
 try  
 {  
 stockman.*sleep*(5000);  
 }  
 catch(InterruptedException e)  
 {  
 e.printStackTrace();  
 }  
 System.*out*.println("仓库门已打开！");  
 }  
 }  
}



14

package java5;  
  
import java.awt.\*;  
import java.awt.event.\*;  
public class Test implements Runnable  
{  
 StringBuffer bf = new StringBuffer();  
 Thread t1,t2;  
 Test(){  
 t1=new Thread(this);  
 t2=new Thread(this);  
 }  
 public synchronized void addChar(char c)  
 {  
 if(Thread.*currentThread*()==t1){  
 while (bf.length()==0){  
 try {  
 wait();  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 bf.append(c);  
 }  
 if (Thread.*currentThread*()==t2){  
 bf.append(c);  
 notifyAll();  
 }  
 }  
 }  
 public static void main(String[] args){  
 Test t=new Test();  
 t.t1.start();  
 t.t2.start();  
 while (t.t1.isAlive()||t.t2.isAlive()){  
 System.*out*.println(t.bf);  
 }  
  
 }  
 public void run(){  
 if(Thread.*currentThread*()==t1){  
 addChar('A');  
 }  
 if (Thread.*currentThread*()==t2){  
 addChar('B');  
 }  
 }  
}



无限循环空格