2016 秋季 java 课程线下考试答案及评分标准（A 卷）

一．卷选择题答案：

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
B	B	D	B	A	D	D	B	D	B	C	B	A	B	C	D	B	A	D	C

二．判断题答案：

1	2	3	4	5	6	7	8	9	10
√	√	√	√	√	√	√	√	x	x

三．完善程序和程序结果题（每空 2 分，共 20 分）
按照要求将代码中空白处填入正确的代码。

```
abstract class Student{
    final static int CourseNo = 3;     //设置整型静态常量 CourseNo 的初始值
为 3
    String type;
    String name;
    int [] courses;
    String courseGrade;

    public Student(String name){
        this.name = name;
        courses = new int[CourseNo];
        courseGrade = null;  //设置 courseGrade 的初始值为 null
    }

    public abstract void calculateGrade();

    public String getType(){
        return type;  //返回 type
    }

    public void setType(String type){
        this.type = type;  //将参数 type 的值赋给成员变量 type
    }

    public int[] getCourses(){
        return courses;
    }

    public String getName(){
        return name;
    }

    public void setCourses(int[] courses){
        this.courses = courses; //将参数 courses 的值赋给成员变量 courses
    }

    public int getCourseScore(int courseNumber){
        return courses[courseNumber];
    }

    public void setCourseScore(int courseNumber, int courseScore){
        this.courses[courseNumber] = courseScore;
    }

    public String getCourseGrade(){
        return courseGrade;
    }
}

class Undergraduate extends Student{
    public Undergraduate(String name){
        super(name);            //调用父类的构造方法
        type = "本科生";
    }
    public void calculateGrade(){
        double average = 0.0;
        int total = 0;
        for (int i =0; i < CourseNo; i++){
            total += courses[i];
        }
        average = (double)total/CourseNo; //必须转 double     //求平均分
        if(average >= 60)   //如果不低于 60 分
            courseGrade = "通过";
        else
            courseGrade = "未通过";
    }
}

public class School{
    public static void main(String [] args){
        Student[] students = new Student[3]; //创建 students 数组, 共 3 个
元素
        students[0] = new Undergraduate("张席");
        students[1] = new Undergraduate("潘伟科");
        students[2] = new Undergraduate("于仕琪");
        for(int i = 0; i < 3; i++){
            students[i].setCourseScore(0,86); //将 0 号课程分数设置为 86
            students[i].setCourseScore(1,75);
            students[i].setCourseScore(2,91);
        }

        for(int i = 0; i < 3; i++){
            students[i].calculateGrade();
        }
        System.out.println("姓名    成绩有无通过");
        for(int i = 0; i < 3; i++){
            System.out.println(students[i].getName()    +   "        "   +
students[i].getCourseGrade()); //打印姓名和通过情况, 二者之间用空格隔开
        }
    }

}
```

四 、 程序设计题（每小题 10 分，共 30 分）

1、编写一个服务器端程序 ServerDemo.java，它能在 8001 端口响应客户端的请求。如果客户端发来内容是字符串“Hello”，服务器将回复字符串“welcome”给客户端。服务器还需要将所有请求的请求时间和请求内容写入日志文件。客户端会将收到的内容打印到屏幕。

```
import java.io.*;
import java.net.*;
import java.util.Date;
public class ServerDemo{
   public static void main(String args[]){
      ServerSocket server=null;
      Socket sk=null;
      DataOutputStream out=null;
      DataInputStream  in=null;
      FileOutputStream logfile = null;
      try{  server=new ServerSocket(8001);
                                                          ----------------(2 分)
```

```
        }
        catch(IOException e1){
            System.out.println("ERROR:"+e1);
        }
        try{  sk=server.accept();
            in=new DataInputStream(sk.getInputStream());
            out=new DataOutputStream(sk.getOutputStream());
                                                    ----------------(2 分)
             logfile = new FileOutputStream("log.txt");
            while(true){
                String s=in.readUTF();
                if("Hello".equals(s)){
                    out.writeUTF("welcome");
                }
                logfile.write( ((new Date())+ ": " +s).getBytes());
                                                    ----------------(2 分)
            }
        }
        catch(IOException e){
            System.out.println(e);
        }
    }
}


import java.io.*;
import java.net.*;
public class ClientDemo{
   public static void main(String args[]){
       String s=null;
       Socket mysocket;
       DataInputStream in=null;
       DataOutputStream out=null;
       try{  mysocket=new Socket("127.0.0.1",8001);
           in=new DataInputStream(mysocket.getInputStream());
           out=new DataOutputStream(mysocket.getOutputStream());
           out.writeUTF("Hello");
                                                    ----------------(2 分)

             s=in.readUTF();
           System.out.println(s);
                                                    ----------------(2 分)

       }
```

```
        catch(IOException e){
            System.out.println(e);
        }
    }
}
```

2、编写一个程序，统计给定文本文件中的单词出现频率，最后按照字典顺序将统计结果打印出来。注：文件中仅有英文单词，单词之间用空格隔开。

评分说明：此题目有多种实现方法，使用 TreeMap<K,V>实现最为简洁。使用其他方式也可以实现，请按照完成的功能给分。

```
import java.io.*;
import java.util.*;
public class WordCount{
    public static void main(String[] args) {
        try{
            FileReader fr = new FileReader("readme.txt");
            BufferedReader br = new BufferedReader(fr);
                                                    ----------------(2 分)
            //使用 TreeMap
            TreeMap<String,  Integer>  counts  =  new  TreeMap<String,
Integer>();
                                                    ----------------(2 分)
            String line = null;

            while( (line=br.readLine()) != null ){
                                                    ----------------(2 分)
                String [] words = line.split(" ");
                for(String word:words){
                    if( counts.containsKey(word) ){
                        counts.put(word, (counts.get(word)+1));
                                                    ----------------(2 分)
                    }
                    else{
                        counts.put(word,1);
                    }
                }
            }

            fr.close();
            br.close();
           //打印结果
            System.out.println(counts);
                                                      -----------------(2 分)
        }

        catch(IOException e){
              System.err.println(e);
        }
    }
}
```

3、编写一个多线程程序，创建两个线程对象，分别在屏幕上打印 1-100 之间的奇数和偶数。

```
public class ThreadDemo{
   public static void main(String args[ ]){
       SubThread t1 = new SubThread(1,100);
       SubThread t2 = new SubThread(2,100);
                                                      -----------------(2 分)
       t1.start();
       t2.start();
                                                      -----------------(2 分)
   }
}
class SubThread extends Thread{
    int start = 0;
    int end = 0;
    SubThread(int start, int end){
                                                      -----------------(2 分)
       this.start = start;
        this.end = end;
                                                      -----------------(2 分)
    }
    public void run(){
       for(int i=start;i<=end;i+=2){
          System.out.println(i);
       }
                                                      -----------------(2 分)
    }
}
```

五 、 附加题（30 分）

使用 Socket 编程，实现一个具有图形界面的聊天软件。详细要求如下：

1. 界面至少包括发送消息的文本框和按钮，接收消息的带滚动条的文本框。
2. 程序既有服务器端又是客户端。即程序能够监听端口 5555 端口，接收连接请求；也能够主动连接其他 IP 的 5555 端口。
3. 一方通过客户端功能向另一方的服务器端发起连接请求，被请求方收到请求后提示用户是否同意连接。如同意，则建立连接进行双向通讯，互发文字信息。
4. 利用该程序可以与多个人聊天，请采用多线程实现多人聊天，并保证多线程传输和显示数据时，不发生数据冲突。
5. 如一方退出聊天会话，另一方会收到提醒并中止会话。

```java
/**
 *服务端
 */

import java.io.*;
import java.net.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

import javax.swing.*;

public class Server extends JFrame implements Runnable {
 private ServerSocket server;
 private Socket connection;
 private OutputStream output;
 private InputStream input;
 private Thread outThread;
 private JTextArea display;
 private JTextField text1;
 private JButton startButton;

 public static void main(String args[]) {
  Server s = new Server();
  s.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }

 public Server() {
  super("Server");
  startButton = new JButton("Start the server");
  text1 = new JTextField(20);
  display = new JTextArea(7, 30);
  display.setEditable(false);
 Container container = getContentPane();
  container.setLayout(new BorderLayout());
  container.add(startButton,BorderLayout.NORTH);
  container.add(new JScrollPane(display),BorderLayout.CENTER);
  container.add(text1,BorderLayout.SOUTH);
                                                                         ----------------(5 分)

 /**//*
   * 给开始按钮添加监听器。
   */
  startButton.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e) {
    display.setText("启动服务器 ");
    startButton.setEnabled(false);
    try {
     //端口设为 5000，最大连接请求为 100 个
     server = new ServerSocket(5555, 100);
     connection = server.accept();
                                                                         ----------------(5 分)
     output = connection.getOutputStream();
     input = connection.getInputStream();
     output.write("连接成功! ".getBytes());
     outThread = new Thread(Server.this);
     outThread.start();
     } catch (IOException ee) {
    }
   }
  });

  /**//*
  /*给文本域添加键盘监听器，按回车发送信息。
       */

  text1.addKeyListener(new KeyAdapter(){
   public void keyPressed(KeyEvent ke) {
    if(ke.getKeyCode() == KeyEvent.VK_ENTER){
     byte writeBytes[] = new byte[50];
     String s = "Server: " + text1.getText() + "";
     text1.setText("");
     writeBytes = s.getBytes();
     display.append(s+" ");
     try {
     output.write(writeBytes);
     } catch (IOException ee) {
     }
     if (s.trim().equals("Server: exit")) {
     outThread.stop();
     quit();
     }
    }
   }
  });
  setSize(300, 400);
  setResizable(false);
  setVisible(true);
 }

 public void run() {
  while (true) {
   byte readBytes[] = new byte[50];
   try {
    input.read(readBytes);//读去对方发送的消息
   } catch (IOException e) {
                                               -----------------(5 分)
   }
   String s = new String(readBytes);
   display.append(s+" ");
   if (s.trim().equals("Client: exit"))
    break;
  }
  quit();
 }

 public void quit() {
  try {
   output.close();
   input.close();
   connection.close();
  } catch (IOException e) {
  }
  startButton.setEnabled(true);
 }
 }
 /**//*
 *客户端
 */
 package edu.jlu.fuliang;
```

```
import java.io.*;
import java.net.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.*;

public class Client extends JFrame implements Runnable {
 private Socket client;
 private OutputStream output;
 private InputStream input;
 private Thread outThread;
 private JTextArea display;
 private JTextField text1;
 private JButton startButton;
 private JMenu loginMenu = new JMenu("登录");
 private JMenuItem register = new JMenuItem("注册");
 private JMenuItem login = new JMenuItem("登录");
 private JMenuBar bar = new JMenuBar();
 private Register registerDlg ;
 private Login loginDlg;
 private RandomAccessFile file;

 public static void main(String args[]) {
  Client c = new Client();
  c.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 }

 public Client() {
  super("Client");
  startButton = new JButton("Connect to server");
  text1 = new JTextField(20);
  display = new JTextArea(7, 30);
  display.setEditable(false);
   loginMenu.add(register);
  loginMenu.add(login);
  bar.add(loginMenu);
  setJMenuBar(bar);
  Container container = getContentPane();
  container.setLayout(new BorderLayout());
  container.add(startButton,BorderLayout.NORTH);
  container.add(new JScrollPane(display),BorderLayout.CENTER);
  container.add(text1,BorderLayout.SOUTH);

  try {
     file = new RandomAccessFile(new File("login.txt"),"rw");
  } catch (IOException e1) {
   e1.printStackTrace();
  }
  registerDlg = new Register(this,file);
  loginDlg = new Login(this,file);
  startButton.addActionListener(new ActionListener(){
   public void actionPerformed(ActionEvent e) {
    display.setText("连接服务器");
    startButton.setEnabled(false);
    try {
     client = new Socket("127.0.0.1", 5555);
     output = client.getOutputStream();
     input = client.getInputStream();
     outThread = new Thread(Client.this);
     outThread.start();
                                                      ----------------(5 分)
    } catch (IOException ee) {
    }
   }
  });
  text1.addKeyListener(new KeyAdapter(){
   public void keyPressed(KeyEvent ke) {
   if(ke.getKeyCode() == KeyEvent.VK_ENTER){
    byte writeBytes[] = new byte[50];
    String s = loginDlg.getLoginName()+": " + text1.getText() + "";
    text1.setText("");
    writeBytes = s.getBytes();
    display.append(s+" ");
    try {
     output.write(writeBytes);
    } catch (IOException ee) {
    }
    if (s.trim().equals(loginDlg.getLoginName()+": exit")) {
     outThread.stop();
     quit();
    }
   }
   }
  });
  register.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
     registerDlg.setVisible(true);
    }
   });
   login.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
     loginDlg.setVisible(true);
    }
   });
   setSize(300, 400);
   setResizable(false);
   setVisible(true);
  }

  public void run() {
   while (true) {
    byte readBytes[] = new byte[1024];
    try {
     input.read(readBytes);
    } catch (IOException e) {
    }
    String s = new String(readBytes);
    display.append(s+" ");
    if (s.trim().equals("Server: exit"))
     break;
   }
                                                        -----------------(5 分)
   quit();
  }

  public void quit() {
   try {
    output.close();
    input.close();
    client.close();
   } catch (IOException e) {
   }
   startButton.setEnabled(true);
  }
 }

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;


public class Login extends JDialog{
  private JTextField textField;
  private JButton loginButton;
  private RandomAccessFile file;//保存注册信息的文件
  private String loginName = "guest";//保存登录者的名字，为登陆为 guest;

  public Login(JFrame f,RandomAccessFile file){
   super(f,"登陆",false);
   this.file = file;
   JPanel panel = new JPanel();
   panel.add(new JLabel("昵称:"));
   textField = new JTextField(10);
   panel.add(textField);
   Container container = getContentPane();
   container.setLayout(new BorderLayout());
   container.add(panel,BorderLayout.NORTH);
   loginButton = new JButton("登陆");
   container.add(loginButton,BorderLayout.SOUTH);
   setVisible(false);
   setBounds(100,200,200,200);
   loginButton.addActionListener(new LoginListener());
  }

  public String getLoginName(){
   return loginName;
  }
  /**//*
   * 登录监听器，当单击登陆按钮时，触发该事件
   * 从文件中读取并查找是否注册过，如果没有找
   * 到则弹出未注册警告。否则弹出欢迎对话框表
   * 示欢迎
   */
  private class LoginListener implements ActionListener{
  public void actionPerformed(ActionEvent e) {
   boolean flag = false;
   try {
    String name = textField.getText().trim();
    textField.setText("");
    file.seek(0);
    while(file.getFilePointer() < file.length()){
     String nik = file.readUTF();
     if(nik.equals(name)){
      flag = true;
      loginName = name;
      break;
     }
    }
    if(!flag){
     String warning="没有找到你的账号请先注册!";
              JOptionPane.showMessageDialog(Login.this,warning,"   警   告
",JOptionPane.WARNING_MESSAGE);
    }else{
     String welcome="欢迎来聊天!";
              JOptionPane.showMessageDialog(Login.this,welcome,"   欢   迎
",JOptionPane.WARNING_MESSAGE);
    }
    Login.this.setVisible(false);
                                                  ----------------(5 分)
   } catch (IOException e1) {
    e1.printStackTrace();
   }
  }
  }
}


import java.awt.*;
import java.awt.event.*;
import java.io.*;
import javax.swing.*;


public class Register extends JDialog{
  private String nickName;
  private JTextField textField;
  private JButton registerButton;
  private RandomAccessFile file;////保存注册信息的文件

  public Register(JFrame f,RandomAccessFile file){
   super(f,"注册",false);
   this.file = file;
   JPanel panel = new JPanel();
   panel.add(new JLabel("昵称:"));
   textField = new JTextField(10);
   panel.add(textField);
   Container container = getContentPane();
   container.setLayout(new BorderLayout());
   container.add(panel,BorderLayout.NORTH);
   registerButton = new JButton("注册");
   container.add(registerButton,BorderLayout.SOUTH);
   setVisible(false);
   setBounds(100,200,200,200);
   registerButton.addActionListener(new RegisterListener());
  }
  /**//*
   * 注册监听器，当单击登注册按钮时，触发该事件
   * 并向文件中写入注册信息。
   */
  private class RegisterListener implements ActionListener{
  public void actionPerformed(ActionEvent e) {
   try {
    file.seek(file.length());
    String  str = textField.getText();
    textField.setText("");
    file.writeUTF(str);
    Register.this.setVisible(false);
   } catch (IOException e1) {
    e1.printStackTrace();
   }
  }
  }
}
```