

| |
|---|
| 院 |
| |
| 级 |
| |
| 号 |
| |
| 名 |
| |

东 北 大 学 考 试 试 卷 (A 卷)

2003—2004 学年第 学期

课程名称: Java 程序设计

得 分

一. 填空

1. (6 分) 最显著的优点: (); 内存的自行管理: ();

(): 取消了头文件、指针算法、结构、单元、运算符重载、虚基类等概念;

(): 编译器在中间过程生成字节码指令, 可以易于任何机器上的解释; Java 虚拟机: (), (), (), (),

(), () 存储区六部分;

() 表示注释以行; () 表示注释一段;

2. (1 分) 下面变量定义那个错误? ()

(A) byte b1; (B) int tem2p; (C) long 2head; (D) char wm;

3. (3 分) 给出结果 $15/4=()$, $15\%2=()$, $11.0/4=()$;

4. (4 分) 求程序中的 a, b, c, d 运行后的结果?

```
int a = 10;
int b = 10;
int d = 0;
double c = 6.36;
a = a * (b++);
d = (int) (-b * c);
```

5. (2 分) 求程序中的 n, message 运行后的结果?

```
String exp = "Expletive ";
String pg = "deleted";
String message = exp + pg;
```

```
String greeting="Hello";
int n=greeting.length();
```

6. (1 分) 求程序中的 n 运行后的结果?

```
Public static void main(String[] args)
{
    int n=0;
    {
        int k;
        int n=1; //?
    }
}
```

密

封

线

3. 3

| 总分 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | | |
|----|---|---|---|---|---|---|---|---|---|---|--|--|
| | | | | | | | | | | | | |

7. (3分)求下面三个程序中的 i 运行后的结果?

```
int i=0;
while(i!=0) {i++;}
```

```
int i=0;
do{i++;}while{i!=0}
```

```
int i=0;
for(int i=1; i<=10; i++);
```

8. (2分)求下面程序中的 i, j 运行后的结果?

```
public class BreakAndContinue {
    public static void main(String[] args) {
        for(int i = 0; i < 100; i++) {
            if(i == 74) break;
            if(i % 9 != 0) continue;
        }
        int i = 0;
        while(true) {
            i++;
            int j = i * 27;
            if(j == 1269) break;
            if(i % 10 != 0) continue;
        }
    }
}
```

9. (2分) (): 对每个对象的处理需求, 和属性一起封装并被视为一个固有的整体。

(): 对象之间一个较窄的、定义明确的界面。

(): 属性及其服务的封装, 通过问题空间的若干实例对问题空间抽象。

面向对象=对象+分类+()+通过消息的通信;

10. (5分)求下面程序中的 count1~5 运行后的结果?

```
public class Robot {
```

82

```

public Robot(int x, int y) {
    this.x = x;
    this.y = y;
}
public void move(int dx, int dy) {
    x += dx; y += dy;
    instanceMoveCount++;
    classMoveCount++;
}
public int getInstanceMoveCount() {
    return instanceMoveCount;
}
public static int getClassMoveCount() {
    return classMoveCount;
}
public int x, y;
private int instanceMoveCount;
static private int classMoveCount;
public static void main(String[] args) {
    Robot r1 = new Robot(0, 0);
    Robot r2 = new Robot(0, 0);
    r1.move(20, 10);
    r1.move(10, 20);
    r2.move(10, 10);
    int count1 = r1.getInstanceMoveCount();
    int count2 = r2.getInstanceMoveCount();
    int count3 = r1.getClassMoveCount();
    int count4 = r2.getClassMoveCount();
    int count5 = Robot.getClassMoveCount();
}
}

```

11. (2 分)给出下面程序的输出结果:

```

public class Minimum {
    int Mine(int i, int j)
    {
        if (i>j) return j;
        else return i;
    };
    public static void main(String[] args) {
        int a[] = { 75, 34, 80, 11, 95, 34, 53, 81, 33, 13 };
        int min = a[0];
        for (int i = 1; i < a.length; i++) {
            min = Mine(min, a[i]);
        }
    }
}

```

14. (5分)输出下面程序的结果:

```
class TT{
    {
        System.out.println("~~");
    }
}

public class RandomIntGenerator{
    public RandomIntGenerator(int l, int h){
        low=l;
        high=h;
        System.out.println("#|");
    }
    public int draw(){
        int r=low+(int)((high-low+1)*nextRandom());
        return r;
    }
    public static void main(String[] args){
        TT t1=new TT();
        TT t2=new TT();
        RandomIntGenerator r1=new RandomIntGenerator(1,10);
        RandomIntGenerator r2=new RandomIntGenerator(0,1);
    }
    private static double nextRandom(){
        int pos=(int)(java.lang.Math.random()*BUFFER_SIZE);
        if(pos==BUFFER_SIZE) pos=BUFFER_SIZE-1;
        double r=buffer[pos];
        buffer[pos]=java.lang.Math.random();
        return r;
    }
    private static final int BUFFER_SIZE=11;
    private static double[] buffer=new double[BUFFER_SIZE];
    static
    {
        int i;
        for(i=0; i<BUFFER_SIZE; i++)
            buffer[i]=java.lang.Math.random();
        System.out.print("#|");
    }
    private int low;
    private int high;
}
```

| |
|-----|
| 得 分 |
| |

二. (1 分)使用封装的类有两种方法, 假设有 `Java. www. make` 对象, 请用任何一种方法为这个对象定义一个实例对象。

| |
|-----|
| 得 分 |
| |

三. (5 分)修改上例, 使 A 和 B 都具有参数的构造函数, 为 C 编写一个构造函数 (输入参数为空), 并且为 C 创建一个实例。

| |
|-----|
| 得 分 |
| |

四. (3 分)定义一个“鸟”的类: 简单描述鸟的属性特征 (羽毛, 嘴, 爪), 和它的基本行为 (飞, 跳)

| |
|-----|
| 得 分 |
| |

五. (5 分)
编写一个程序, 编译完成后, 做如下:
`java 程序名 姓名`
程序执行后, 返回“姓名”?

| |
|-----|
| 得 分 |
| |

六. (5 分)将一个集合 [1, 16, 5, 7, 20, 37, 36, 66] 有小到大进行排列?

24

| |
|-----|
| 得 分 |
| |

- 七. (5分)请编写两个带有构造函数(输入参数为空)的类A和B,并且产生类C,使其继承A,并在C中以B为成员,不要为C提供任何构造函数,最后,产生一个C的实例观察结果。

| |
|-----|
| 得 分 |
| |

八. 改错

1. (1分)变量初始化的位置正确么?

```
int test(int testval, int target) {  
    if(testval > target)  
        result = +1;  
    else if(testval < target)  
        result = -1;  
    else  
        result = 0; // Match  
    return result;  
    int result = 0;  
}
```

2. (1分)现有两个类A和B,其中A类继承了B类,即 `class A extends B{}`;问下面的表达示是否正确?

```
A a=new A();  
B b=a;
```

3. (3分)

```
class A{  
    String name;  
    int hirDay;  
    double salary;  
  
    A(String n, double s, int d)  
    {  
        name = n;  
        hireDay = d;  
        double salary = s;  
    }  
}
```

是否可以使用A("James", 100, 3)来初始化类A中的三个变量(name, hirDay, salary).
如果不能如
何改正?

9/5

2003 年研究生 Java 语言试题 (开卷)

姓名_____ 学号_____ 专业_____

注意事项:

1. 第一题答在本题卷上, 其余题在答题本上作答。
2. 写清自己的学号
3. 部分试题需要认真查找相关资料才能完成。

一、名词解释 (每题 1 分, 共 10 分)

1. GUI
2. J2ME
3. final
4. Vector
5. JDBC
6. abstract
7. interface
8. upcasting
9. JVM
10. Override

二、简答题

1. 给出 5 个使用 java 进行数据库编程时常用的类及其功能, 并写出一段简短的数据库操作程序。(15 分)

2. 改错: 在 FindErrors.java 文件的代码如下。请改错, 并将正确的程序写在答题纸上。(10 分)

```
public class Find {
    private int i;
    public void f() {System.out.println("Find.f()");}
}
public class NewFind extends Find {
    public void f() {System.out.println("NewFind.f()");}
    public void f1() {System.out.println("NewFind.f1()");}
    public void f(long i) {this.i=i;}
}
public class FindErrors extends Find {
    public void f2(Find ff);
```



```

        if(f instanceof NewFind){
            f1();
        }
    }
}

public void main(String args) {
    NewFind nf=new NewFind();
    f2(nf);
}
}

```

三、论述题

1. 查找资料并说明，什么是 java Bean，什么是 EJB，两者的功能有什么差别？
(10 分)
2. 查找资料并说明，J2EE 由哪些部分组成？其框架是什么样的？应该如何架设？
(15 分)

四、编程题

1. 用程序实现仿记事本的程序。要求：能使用菜单读取用户选定的文本文件并显示到文本区域，支持文本文件的存储和另存(save as)。在程序中使用 javadoc 标签进行注释。
(20 分)
2. 使用 socket 编写网络应用程序，要求实现：
 - (1) 在客户与服务端间传输一个对象。
 - (2) 服务器采用多线程结构，至少支持 2 个客户。(20 分)

26

2006 年研究生 java 语言试题 (开卷) 试题及答案

一、名词解释 (每题 2 分, 共 20 分)

1. J2EE: Java2 Platform, Enterprise Edition. Java2 平台企业版。
2. Wrapper class: 把基本数据类型升级为引用数据类型的类。
3. Delegation: 委托, 一种引用类型。组件将事件的处理委托给特定对象, 当该组件发生指定事件的时候, 由委托的对象接受并处理该事件。
4. Layout Manager: 布局管理器。
功能包括: 1、计算容器的最佳和最小尺寸
2、将容器中的组件大小及位置作合适的安排。
5. Swing: Java API (java . swing) 中的包, 包括与图形用户接口相关的类。
6. Decorator: 装饰器: 利用层次化对象动态和透明地添加单个对象的能力的做法。
7. Serialize: 在外部文件中存储和获取对象的过程。
8. Downward cast: 父类到子类的强制转换。
9. Singleton: 单子, 它能提供对象的一个 (而且只有一个) 实例。创建单子的关键就是防止客户程序员采用除由我们提供的之外的任何一种方式来创建一个对象。
10. Overload: 方法的重载, 方法名相同, 参数个数或者类型不同。

二、简答

1 解释下列关键字的含义, 并说明其在编程中的特点 (10 分)

(1) .static: 用 static 修饰符修饰的属性是紧属于类的静态属性, 相应的成员变量也称为类变量或者静态变量。它保存在类的内存区域的公共存储单元。用 static 修饰符修饰的方法成为静态方法, 也叫类方法, 在调用时, 一般适用类名做前缀。当然, 也可以通过对象来调用, 但是必须清楚的是它不依赖任何对象。在 static 方法中只能处理类变量, 也可以访问其他 static 方法, 但绝不能访问任何归属对象空间的变量或方法。

(2) .protected: 类中定义为 protected 的成员可以被这个类本身、它的子类 (包括同一个包以及不同包中的子类) 以及同一个包中所有的其他类所访问。

(3) .abstract: 抽象类和抽象方法。拥有抽象方法的类一定是抽象类, 抽象类被继承后, 必须将所有的抽象方法实现, 否则子类仍然是抽象类。抽象类表示的是一个抽象的概念, 不能被实例化。

(4) .transient: 临时。控制序列化过程时, 可能有一个特定的子对象不愿让 Java 的序列化机制自动保存与恢复。一般地, 若那个子对象包含了不想序列化的敏感信息 (如密码), 就会面临这种情况。即使那种信息在对象中具有 “private” (私有) 属性, 但一旦经序列化处理, 人们就可以通过读取一个文件, 或者拦截网络传输得到它。为防止对象的敏感部分被序列化, 一个办法是将自己的类实现为 Externalizable, 就象前面展示的那样。这样一来, 没有任何东西可以自动序列化, 只能在 writeExternal() 明确序列化那些需要的部分。然而, 若操作的是一个 Serializable 对象, 所有序列化操作都会自动进行。为解决这个问题, 可以用 transient (临时) 逐个字段地关闭序列化, 它的意思是 “不要麻烦你 (指自动机制) 保存或恢复它了——我会自己处理的”。

(5) .final: 如果一个类被 final 修饰符所修饰和限定, 说明这个类成为最终类, 它不可能有子类, 有子类就意味着可以定义新成员。用 final 修饰符修饰的类的方法, 是功能和内部语句不能被更改的最终方法, 即是不能被当前类的子类重新定义的方法。所有已经被 private 修饰符限定为私有的方法, 以及所有包含在 final 类中的方法, 都被缺省地认为是 final 的。由 final 标记的变量即为常量。

2. 有一段代码, 如下所示。程序运行的结果为什么为 false? 如何通过 override 修改 Value

类, 使结果为 true?(15 分)

```
//Equals2.java
class Value{int i;}
public class Equals2{
    public static void main(String[] args){
        Value n1=new Value();
        Value n2=new Value();
        n1.i=100;
        n2.i=100;
        System.out.println(n1.equals(n2));
    }
}
```

答: 因为 equals()方法是进行的地址的比较。虽然 n1.i 和 n2.i 的值相等, 但是他们的地址不同, 所以运行结果为 false.

重写 Value 后的代码:

```
//Equals2.java
class Value{
    int i;
    boolean equals(Value n)//重写了 equals()方法
    {
        if(this.i==n.i)return true;
        else return false;
    }
}

public class Equals2{
    public static void main(String[] args){
        Value n1=new Value();
        Value n2=new Value();
        n1.i=100;
        n2.i=100;
        System.out.println(n1.equals(n2));
    }
}
```

3. 改错: 在 FindErrors.java 文件的代码如下。请找到错误并改错。(8 分)

```
public class Find{//去掉
    private int i;//改为 public
    public void f(){System.out.println("Find.f()");};//分号放外边
}

public class NewFind extends Find{//去掉
    public void f(){System.out.println("NewFind.f()");}
    public void f1(){System.out.println("NewFind.f1()");}
    public void f(long i){this.i=i;}//改为 int
}
```

```

public class FindErrors extends Find{
    public void f2(Find ff){
        if(ff instanceof NewFind){
            NewFind mf=(NewFind)ff;
            mf.f1();//父类不能调用子类方法去掉需强制类型转换
            ff.f();
        }
    }

    public void static main(String []args){//缺少 static 和 []
        NewFind nf=new NewFind();
        FindErrors fe=new FindErrors();
        fe.f2(nf);//通过对象调用方法
    }
}

```

4. 请详细说明，在 jdk1.5 中，枚举类型是如何保证类型安全的？（10 分）

答：public class Season{

```

    private final String name;
    private Season(String name){
        this.name=name;
    }

    public static final Season SPRING=new Season("spring");
    public static final Season SUMMER=new Season("summer");
    public static final Season AUTUMN=new Season("autumn");
    public static final Season WINTER=new Season("winter");
}

```

如上代码，这里构造函数不能为 public 或者 protected，从而保证构造函数只能在内部使用，客户代码不能 new 一个枚举值的实例出来。因为枚举值是 public static final 的常量。

5. 某人在 DataAccess.java 和 ComputeData.java 两个文件中分别定义了 DataAccess 和 ComputeData 两个类。这两个类都属于名称为 cn.edu.neu.ise 的 package。应该如何做才能在 Company 类中引用 Employee 类？（7 分）（原题好像出错了，应该是如何做才能在 DataAccess 类中引用 ComputeData 类吧？不太明白）

答：按照修改后的做，这道题 7 分，应该不难的。

// ComputeData.java 文件一

```

package cn.edu.neu.ise;

class ComputeData{
    int salary;
    int age;
}

```

// DataAccess.java 文件二

```

package cn.edu.neu.ise;
import cn.edu.neu.ise.ComputeData;

public class DataAccess{
    public static void main(String []args)

```



```

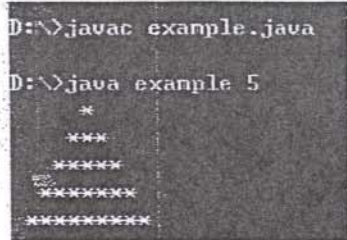
    {
        Employee sb=new Employee();
    }
}

```

三、编程题

1. 用程序实现三角形的输出。要求：(1) 能从命令行参数读取三角形的高度。
(2) 实现至少一处自定义例外。(3) 进行例外控制 (15 分)

例如: java example 5



```

D:\>javac example.java
D:\>java example 5
*
**
***
****
*****

```

```

class ParamNotFoundException extends Exception
{
    public ParamNotFoundException(String message)
    { super(message);}
}

class example
{
    public static int getHeight(String args[]) throws ParamNotFoundException
    {
        int m;
        if(args.length==0)
            throw new ParamNotFoundException("没有命令行参数异常发生!");
        m=Integer.parseInt(args[0]);
        return m;
    }

    public static void main(String args[])
    {
        int H=0;//返回三角形高度
        try{
            H=getHeight(args);
        }
        catch(ParamNotFoundException e)
        {
            System.out.println("发生异常, 请再次运行, 并输入命令行参数!");
        }
        catch(Exception e)
        {
            System.out.println("发生异常, 请再次运行, 并输入命令行参数!");
        }
    }
}

```

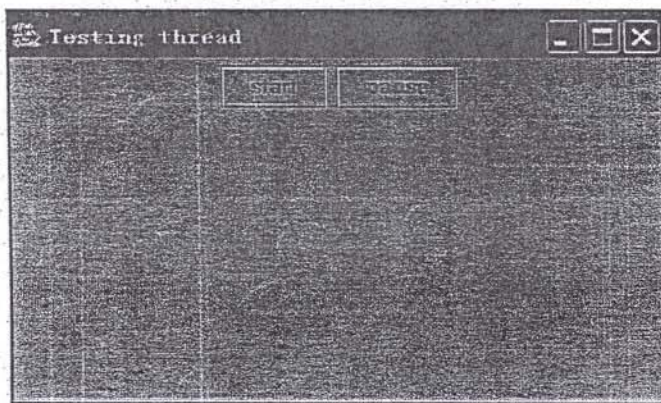
```

    }
    int i, j, k;
    for(j=0; j<H; j++)//控制打印 m 行
    {
        for(i=j; i<H; i++)//控制每行前面的空格数
        { System.out.print(" ");}
        for(k=0; k<=j*2; k++)//控制每一行的星号个数
        { System.out.print("*"); }
        System.out.println(" "); //回车换行
    }
}
}

```

2. 创建一个 JFrame 如图所示。要实现的功能:

- (1) 点击 start 按钮后, 每间隔 100 毫秒在该 Frame 上随机位置画一个红色的点;
- (2) 点击 pause 按钮后, 绘图暂停;
- (3) 再次点击 pause 则继续绘图。(15 分)



```

import java.awt.*;
import javax.swing.*;
import java.awt.Color;
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.event.*;
import java.util.Random;

public class jframe extends JFrame{
    JButton jStart, jPause;
    JPanel framepanel;
    int x;
    int y;
    boolean start = false;
    boolean startRun = false;
    static Random r = new Random();
}

```

```

Thread pThread = new Thread(new PaintThread());

public JFrame(String s) { //构造函数, 界面初始化
    super(s);
    //初始化面板和按钮
    framepanel = new JPanel();
    jStart = new JButton("start");
    jPause = new JButton("pause");
    framepanel.setLayout(new FlowLayout());
    framepanel.add(jStart);
    framepanel.add(jPause);
    //设置布局类型
    Container c = getContentPane();
    c.setLayout(new BorderLayout()); //设置为 BorderLayout
    c.add(framepanel, BorderLayout.NORTH);
    this.setLocation(120, 120);
    jStart.addActionListener(new sMonitor()); //为 start 按钮添加事件触发器 sMonitor
    jPause.addActionListener(new pMonitor()); //为 start 按钮添加事件触发器 pMonitor
    setSize(600, 600);
    setVisible(true);
    this.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            if (start) { start = false; }
            System.exit(0);
        }
    }); //关闭事件
    pThread.start(); //画图开始
    this.setVisible(true);
}

public void paint(Graphics g) { //重写 paint() 方法
    super.paint(g);
    if (start) draw(g);
}

public void draw(Graphics g) { //重写 draw() 方法
    g.setColor(Color.RED);
    g.fillOval(x, y, 15, 15);
}

private class PaintThread implements Runnable { //画点线程
    public void run() { //重写 run() 方法
        while (true) {
            while (start) { //线程开始标志为真
                repaint();
            }
            try {

```

```

        x = r.nextInt(600);
        y = r.nextInt(600);
        Thread.sleep(100);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
}
}

class sMonitor implements ActionListener { //sMonitor 事件
    public void actionPerformed(ActionEvent e) {
        if(!start) {
            startRun = true;
            start = true;
        }
    }
}

class pMonitor implements ActionListener { //pMonitor 事件
    public void actionPerformed(ActionEvent e) {
        if(start) {
            start = false;
        } else if(startRun) {
            start = true;
        }
    }
}

public static void main(String[] args) {
    new JFrame("Testing thread");
}
}

```

说明：此答案由嵌入式实验室无限传感器网络小组的研一成员们一块整理编写，时间仓促，难免有不太正确的地方，望谅解。谨以此献给选 java 课的同学，祝大家考试顺利！

2002 年研究生 Java 语言试题 (开卷)

姓名 _____ 学号 _____ 专业 _____

注意事项:

1. 第一题、第二题的 3 (改错题) 答在本题卷上, 其余题答在答题本上。
2. 写清自己的学号。

一、名词解释 (每题 2 分, 共 20 分)

1. GUI *Graphic User Interface* - 图形用户界面, 以图形的操作界面为操作, 具有良好的人机交互界面, 如微软的 Windows 操作系统。
2. URL *Universal Resource Location* - 统一资源定位或全球资源地址, 唯一定位到所需资源。
3. final 最终的含义, 在 Java 中用来表示其修饰的类、变量、方法是最终类、类或变量所定义, 覆盖。
4. OOP *Object Orient Programming* 面向对象程序设计
5. http *Hyper Text Transport Protocol* 超文本传输协议, 一种网络通信
6. abstract 抽象, Java 语言中用于修饰一个无法实例化的类或方法, 该类或方法
7. interface 接口或界面, 在 Java 中使用接口是为了解决 Java 不支持多继承, 采用接口来弥补多继承的功能。
8. upcasting
9. JVM

10. Override 覆盖 Java 中子类对父类的方法重新定义, 使之具有新功能。同义、同参数、同返回值, 这种情况称为方法覆盖。

二、简答题

1. 什么是内存垃圾收集机制, 在同名现象中内存垃圾收集机制是如何参与的? (10 分)
2. 有一段代码, 如下所示。程序运行的结果为什么为 false? 如何通过 **override** 修改 Value 类, 使结果为 true? (15 分)

```
//Equals2.java
class Value {
    int i;
}

public class Equals2 {
    public static void main(String[] args) {
        Value n1=new Value();
        Value n2=new Value();
        n1.i=100;
        n2.i=100;
        System.out.println(n1.equals(n2));
    }
}
```

```
class Value {
    int i;
    public Boolean equals (val
    {
        if (this.i == n.i)
            return True;
        else return False;
    }
}
```

3. 改错：在 FindErrors.java 文件的代码如下。请找到语法错误并改错。（10 分）

```
public class Find {
    private int i;
    public void f() { System.out.println("Find.f()"); }
```

行
命令或字

VI 的一种类型代表

通信网络

数据再生

设

级时能继续

学问题

```
    }
    class NewFind extends Find {
        public void f() { System.out.println("NewFind.f()"); }
        public void f1() { System.out.println("NewFind.f1()"); }
        public void f(long i) { this.i=i; }
```

```
    }
    public class FindErrors extends Find {
        public void f2(Find ff) {
            if (ff instanceof NewFind) {
                ff.f1(); ff.f();
            }
        }
```

Find ff = new Find();

```
        public void main(String args) {
            NewFind nf = new NewFind(); f2(nf);
        }
    }
}
```

4. 判断下面程序的运行结果，说明其原因并进行改进。（10 分）

```
public class Outer4 {
    private int size = 5;
    public Object makeTheInner(int localVar) {
        final int finalLocalVar = 6;
        class Inner {
            public String toString() { return ("localVar=" + localVar + " "); }
        }
        return new Inner();
    }
    public static void main(String[] args) {
        Outer4 outer = new Outer4(); Object obj = outer.makeTheInner(47);
        System.out.println("The object is " + obj);
    }
}
```

改变的方法

四、编程题

1. 用程序实现三角形的输出。要求：(1) 能从命令行参数读取三角形的高度。（15 分）
例子：java example 5 (2) 能进行差错控制与例外控制。

```
*
***
*****
*****
*****
```

2. 用程序实现仿记事本的程序。要求：能使用菜单读取用户选定的文本文件并显示文本区域，支持文本文件的存储和另存(save as)。（20 分）