**实验1 键盘操作练习**

**实验目的：**掌握使用Thread的子类创建线程

**实验要求**：编写一个java应用程序，在主线程中再创建两个线程，一个线程负责给出键盘上的字母，另一个线程负责让用户在命令行输入所给出的字母。

**程序模板：**

按模板要求，将【代码】替换为java程序代码

**TypeKey.java**

public class TypeKey {

public static void main(String args[]) {

System.out.println("键盘练习(输入#结束程序)");

System.out.printf("输入显示的字母(回车)\n");

Letter letter;

letter = new Letter();

GiveLetterThread giveChar;

InuptLetterThread typeChar;

【代码1】//创建线程giveChar

giveChar.setLetter(letter);

giveChar.setSleepLength(3200);

【代码2】创建线程typeChar

typeChar.setLetter(letter);

giveChar.start();

typeChar.start();

}

}

**Letter.java**

public class Letter {

char c ='\0';

public void setChar(char c) {

this.c = c;

}

public char getChar() {

return c;

}

}

**GiveLetterThread.java**

public class GiveLetterThread extends Thread {

Letter letter;

char startChar ='a',endChar = 'z';

int sleepLength = 5000;

public void setLetter(Letter letter) {

this.letter = letter;

}

public void setSleepLength(int n){

sleepLength = n;

}

public void run() {

char c = startChar;

while(true) {

letter.setChar(c);

System.out.printf("显示的字符:%c\n ",letter.getChar());

try{

【代码3】//调用sleep方法使得线程中断sleepLength豪秒

}

catch(InterruptedException e){}

c = (char)(c+1);

if(c>endChar)

c = startChar;

}

}

}

**InuptLetterThread.java**

import java.awt.\*;

import java.util.Scanner;

public class InuptLetterThread extends Thread {

Scanner reader;

Letter letter;

int score = 0;

InuptLetterThread() {

reader = new Scanner(System.in);

}

public void setLetter(Letter letter) {

this.letter = letter;

}

public void run() {

while(true) {

//System.out.printf("输入显示的字母(回车)\n");

String str = reader.nextLine();

char c = str.charAt(0);

if(c==letter.getChar()) {

score++;

System.out.printf("\t\t输入对了,目前分数%d\n",score);

}

else {

System.out.printf("\t\t输入错了,目前分数%d\n",score);

}

if(c=='#')

System.exit(0);

}

}

}