第十三讲单元作业

问题1：

设计一个点类(Point)，具有数据成员x,y（点的坐标），以及设置、输出数据成员及求两点之间距离的功能。再编写主函数对该类进行测试。

答案1：

#include <iostream>

using namespace std;

class Point

{

private:

double x,y;

public:

void set(double x1,double y1)

{

x=x1;

y=y1;

}

void print()

{

cout<<"("<<x<<","<<y<<")"<<endl;

}

double dist(Point p)

{

return sqrt((x-p.x)\*(x-p.x)+(y-p.y)\*(y-p.y));

}

};

int main()

{

Point p1,p2;

double x1,y1,x2,y2;

cin>>x1>>y1;

p1.set(x1,y1);

cout<<"point p1:";

p1.print();

cin>>x2>>y2;

p2.set(x2,y2);

cout<<"point p2:";

p2.print();

cout<<"distance between p1 and p2:"<<p1.dist(p2)<<endl;

system("pause");

return 0;

}

问题2：

设计一个字符串类(Mystring)，除具有一般的输入输出字符串的功能外，还要求具有计算字符串长度、连接两个字符串等功能，其中求字符串长度和连接字符串功能不能直接调用字符串处理函数。再编写主函数对该类进行测试。

答案2：

#include <iostream>

using namespace std;

class Mystring

{

private:

char str[100];

public:

void set(char s[])

{

strcpy(str,s);

}

void print()

{

cout<<str<<endl;

}

int Length()

{

int i=0;

while(str[i]!='\0')

i++;

return i;

}

void strcat1(Mystring m1)

{

int i=0,j;

while(str[i]!='\0')

i++;

j=0;

while(m1.str[j]!='\0')

{

str[i]=m1.str[j];

i++;

j++;

}

str[i]='\0';

}

};

int main()

{

Mystring m1,m2;

char s1[100],s2[100];

gets(s1);

m1.set(s1);

m1.print();

cout<<"length="<<m1.Length()<<endl;

gets(s2);

m2.set(s2);

m2.print();

cout<<"length="<<m2.Length()<<endl;

m1.strcat1(m2);

cout<<"连接后的串:"<<endl;

m1.print();

system("pause");

return 0;

}

问题3：

设计一个分数类Fraction。该类的数据成员包括分子fz和分母fm；类中还包括如下成员函数：

（1）构造函数，用于初始化分子和分母。

（2）成员函数print，将分数以 "fz/fm" 的形式输出。

（3）成员函数Reduction，用于对分数的分子和分母进行约分。

再编写主函数对该类进行测试。

答案3：

#include <iostream>

using namespace std;

class Fraction

{

int fz,fm;

public:

Fraction(int fz1,int fm1)

{

if(fm1!=0)

{

fz=fz1;

fm=fm1;

}

else

{

cout<<"分母不能为0"<<endl;

exit(1);

}

}

void print()

{

cout<<fz<<'/'<<fm<<endl;

}

void Reduction()

{

int gcd,r,fz1=fz,fm1=fm;

while((r=fm1%fz1)!=0)

{

fm1=fz1;

fz1=r;

}

gcd=fz1;

fz=fz/gcd;

fm=fm/gcd;

}

};

int main()

{

int fz,fm;

cin>>fz>>fm;

Fraction f(fz,fm);

f.print ();

f.Reduction();

f.print ();

system("pause");

return 0;

}