// GraduateStudent.cpp : 定义控制台应用程序的入口点。

//

/\* 第三周 第二题

2) 题目：考虑大学生的学生情况，

试利用单一继承来实现学生和毕业生两个类，

设计相关的数据成员及函数，测试继承情况。

提示：作为学生，有学号、姓名、学校名称及入学时间等基本信息，

而毕业生除了这些信息之外，

还应有毕业时间、所获学位等信息，

考虑成员类型、继承方式。

编译环境：Visual Studio 2015

作者：杨政权

\*/

#include "stdafx.h"

#include <iostream>

#include <string>

using namespace std;

class Student //学生类

{

protected:

char StuId[32]; //学号

char StuName[32]; //姓名

char University[32]; //学校

char EntranceTime[32]; //入学时间

public:

Student(){} //无参构造函数

//有参构造函数

Student(char \*Id, char\* Name, char\* university, char\* entranceTime)

{

strcpy(StuId, Id);

strcpy(StuName, Name);

strcpy(University, university);

strcpy(EntranceTime, entranceTime);

}

//设置学号与取得学号

void SetStuId(const char \*aId)

{

strcpy(StuId, aId);

}

char \*GetStuId(void)

{

return StuId;

}

//设置学生姓名与取得学生姓名

void SetStuName(char \*name)

{

strcpy(StuName, name);

}

char \*GetStuName(void)

{

return StuName;

}

//设置学校与得到学校信息

void SetUniversity(char \*university)

{

strcpy(University, university);

}

char \*GetUniversity(void)

{

return University;

}

//设置入学时间与取得入学时间信息

void SetEntranceTime(char \*Time)

{

strcpy(EntranceTime, Time);

}

char \*GetEntranceTime(void)

{

return EntranceTime;

}

virtual void Print(void)

{

cout << "学号是：" << StuId << endl;

cout << "姓名是：" << StuName << endl;

cout << "学校是：" << University << endl;

cout << "入学时间是：" << EntranceTime << endl;

}

};

class GraduateStudent :public Student

{

protected:

char GraduateTime[32]; //毕业时间

char Degree[32]; //学位

public:

//构造函数

GraduateStudent(){}

GraduateStudent(char \*Id, char\* Name, char\* university,

char\* entranceTime,char\* graduatetime,char\* degree)

{

strcpy(StuId, Id);

strcpy(StuName, Name);

strcpy(University, university);

strcpy(EntranceTime, entranceTime);

strcpy(GraduateTime, graduatetime);

strcpy(Degree, degree);

}

//设置毕业时间和取得毕业时间信息

void SetGraduateTime(char \*graduatetime)

{

strcpy(GraduateTime, graduatetime);

}

char \*GetGraduateTime(void)

{

return GraduateTime;

}

//设置学位和取出学位信息

void SetDegree(char \*degree)

{

strcpy(Degree, degree);

}

char \*GetDegree(void)

{

return Degree;

}

void Print(void) //打印信息

{

cout << "学号是：" << StuId << endl;

cout << "姓名是：" << StuName << endl;

cout << "学校是：" << University << endl;

cout << "入学时间是：" << EntranceTime << endl;

cout << "毕业时间是：" << GraduateTime << endl;

cout << "取得学位是：" << Degree << endl;

}

};

void main()

{

//测试代码

Student student("1500860420", "杨政权", "贵州大学", "20150901"); //赋值

student.Print(); //打印

cout << endl<<endl;

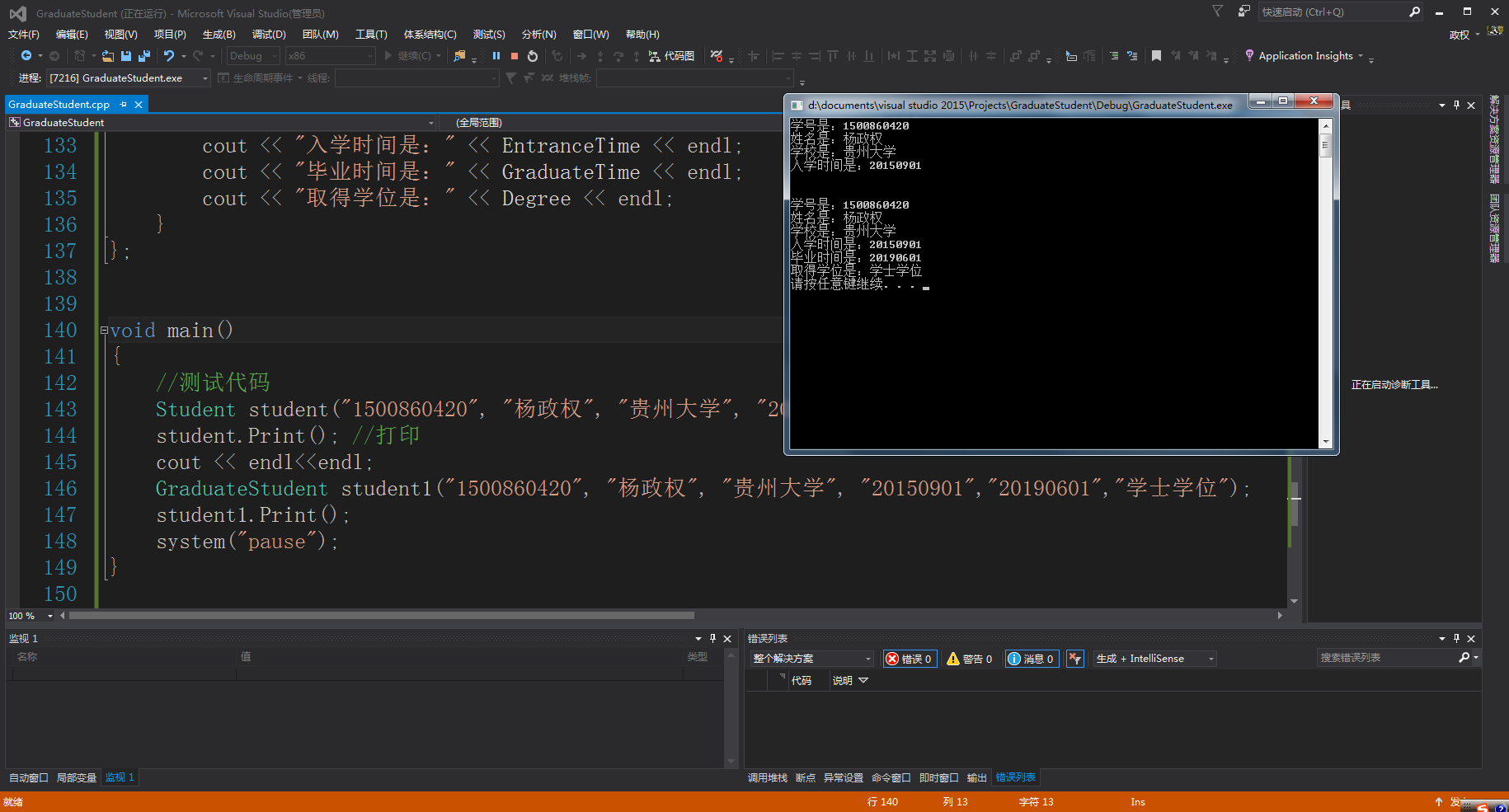
GraduateStudent student1("1500860420", "杨政权", "贵州大学", "20150901","20190601","学士学位");

student1.Print();

system("pause");

}

//运行截图



//实验总结：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\* 这个实验和上一个实验差不多

\*\*\* 没什么难度

\*\*\* 由于时间关系，我也没做什么扩展，还是没优化

\*\*\* 还是将就着看吧

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/