#include <iostream>  
  
using namespace std;  
  
#pragma region 정적할당과 동적할당의 차이점  
/\*  
 1. 정적할당과 동적할당의 차이점  
  
할당 시점: 정적할당은 컴파일 시점에, 동적할당은 실행 시점에 메모리가 할당된다.  
메모리 위치: 정적할당은 데이터 영역 또는 스택 영역에, 동적할당은 힙 영역에 메모리가 할당된다.  
유연성: 동적할당은 메모리 크기를 동적으로 조정할 수 있어 유연성이 높지만  
 정해진 메모리의 크기의 프로그램이라면 정적할당을 통해 안전성을 높힐 수 있다.  
  
  
\*/  
#pragma endregion  
  
#pragma region 클래스란 무엇인가  
/\*  
 2. 클래스란 무엇인가  
  
 클래스는 구조체와 같은 역할을 하며  
 멤버 변수와 멤버 함수를 지니고 있다  
  
 접근지정자를 통해 멤버 변수나 함수를 접근 가능하거나 불가능하게 할 수 있다  
 public: 모든 외부 코드에서 접근 가능.  
 private: 클래스 내부에서만 접근 가능.  
 protected: 클래스와 그 하위 클래스에서만 접근 가능.  
  
 생성자를 통해 객체를 초기화 할 수 있다  
  
 캡슐화 (Encapsulation):  
 클래스는 데이터와 함수를 하나의 단위로 묶어, 외부에서 접근을 제한할 수 있다.  
 이를 통해 데이터의 무결성을 유지할 수 있다.  
 상속 (Inheritance):  
 클래스는 다른 클래스의 특성과 기능을 상속받아 재사용할 수 있다.  
 이를 통해 코드의 재사용성을 높일 수 있다.  
 다형성 (Polymorphism):  
 동일한 인터페이스를 통해 서로 다른 동작을 수행할 수 있다.  
 주로 가상 함수와 함께 사용된다.  
\*/  
#pragma endregion  
  
/\*  
 3. 구조체를 활용한 학생 정보 출력하기  
 학생 정보를 출력하는 함수  
 학생 정보를 추가하는 함수  
 참조방식으로 하시오  
\*/  
  
struct Student  
{  
 string name;  
 int gender;  
 int age;  
 int height;  
 int weight;  
  
 Student()   
 {  
 cout << " 학생이 생성되었습니다 " << endl;  
 }  
 ~Student()  
 {  
 cout << " 학생이 소멸되었습니다 " << endl;  
 }  
  
 void SetName(const string& \_name)  
 {  
 name = \_name;  
 }  
 void SetGender(const int& \_gender)  
 {  
 gender = \_gender;  
 }  
 void SetAge(const int& \_age)  
 {  
 age = \_age;  
 }  
 void SetHeight(const int& \_height)  
 {  
 height = \_height;  
 }  
 void SetWeight(const int& \_weight)  
 {  
 weight = \_weight;  
 }  
   
 string GetName()  
 {  
 return name;  
 }  
 int GetGender()  
 {  
 return gender;  
 }  
 int GetAge()  
 {  
 return age;  
 }  
 int GetHeight()  
 {  
 return height;  
 }  
 int GetWeight()  
 {  
 return weight;  
 }  
  
 void SetStudent()  
 {  
 string s\_input;  
 int i\_input;  
 while (true)  
 {  
 cout << endl;  
 cout << "학생 이름을 입력하시오" << endl;  
 cin >> s\_input;  
  
 if (!s\_input.empty())  
 {  
 SetName(s\_input);  
 break;  
 }  
  
 cout << "잘못 입력하였습니다" << endl;  
 }  
  
 while (true)  
 {  
 cout << endl;  
 cout << "학생 성별을 입력하시오" << endl;  
 cout << "1 : 남자 2 : 여자" << endl;  
 cin >> i\_input;  
  
 if (i\_input >= 1 && i\_input <= 2)  
 {  
 SetGender(i\_input);  
 break;  
 }  
  
 cout << "잘못 입력하였습니다" << endl;  
 }  
  
 while (true)  
 {  
 cout << endl;  
 cout << "학생 나이를 입력하시오" << endl;  
 cin >> i\_input;  
  
 if (i\_input >= 8 && i\_input < 20)  
 {  
 SetAge(i\_input);  
 break;  
 }  
  
 cout << "잘못 입력하였습니다" << endl;  
 }  
  
 while (true)  
 {  
 cout << endl;  
 cout << "학생 키를 입력하시오" << endl;  
 cin >> i\_input;  
  
 if (i\_input > 0 && i\_input < 200)  
 {  
 SetHeight(i\_input);  
 break;  
 }  
  
 cout << "잘못 입력하였습니다" << endl;  
 }  
  
 while (true)  
 {  
 cout << endl;  
 cout << "학생 몸무게를 입력하시오" << endl;  
 cin >> i\_input;  
  
 if (i\_input > 0 && i\_input < 200)  
 {  
 SetWeight(i\_input);  
 break;  
 }  
  
 cout << "잘못 입력하였습니다" << endl;  
 }  
 }  
 void GetStudent()  
 {  
 cout << "학생의 이름은 " << GetName() << "입니다" << endl;  
 cout << "학생의 성별은 ";  
 if(GetGender() == 1)  
 cout << "남자 입니다" << endl;  
 else  
 cout << "여자 입니다" << endl;  
 cout << "학생의 나이는 " << GetAge() << "입니다" << endl;  
 cout << "학생의 키는 " << GetHeight() << "입니다" << endl;  
 cout << "학생의 몸무게는 " << GetWeight() << "입니다" << endl;  
 }  
  
  
};  
  
int main()  
{  
 int b = 0;  
  
 Student \*A = new Student;  
 A->SetStudent();  
 A->GetStudent();  
  
 delete A;  
}