2.2 类的设计_物联网 / 嵌入式工程师 - 慕课网

> 慕课网慕课教程 2.2 类的设计涵盖海量编程基础技术教程，以图文图表的形式，把晦涩难懂的编程专业用语，以通俗易懂的方式呈现给用户。

2. 类的设计

面向对象的特点：

```
#include <stdio.h>
#include <string.h>

typedef struct student{
     char name[20];
     int  age;
     int score;
}student_t;

void studentPlayGame(student_t *stu,const char *game)
{
     printf("Name : %s\n",stu->name);
     printf("Age  : %d\n",stu->age);
     printf("Score: %d\n",stu->score);
     printf("Game : %s\n",game);

     return;
}

int main(void)
{
    student_t stu;

    strcpy(stu.name,"xiaoming");
    stu.age   = 10;
    stu.score = 100;
    studentPlayGame(&stu,"King");

    return 0;
 }
```

分析不足点:

- name 是一个数组，大小固定，无法动态扩展

- studentPlayGame 是学生的一个行为，应该包含在 student 结构体中，但是 C 语言的结构体中是无法包含函数实现

在 C++ 中可以很好的解决以上两个问题:

```
#include <iostream>


using namespace std;


struct Student
{
    string name;
    int  age;
    int  score;
    void playGame(const string &game);
};


void Student::playGame(const string &game)
{
     cout << "Name : " << name  << endl;
     cout << "Score: " << score << endl;
     cout << "Age  : " << age   << endl;
     cout << "Game : " << game  << endl;


     return;
}


int main(void)
{
     Student stu;


     stu.name  = "xiaoming";
     stu.score = 100;
     stu.age   = 10;


     stu.playGame("king");


     return 0;
}
```

可以看到 C++ 对 struct 关键字的功能是有增强的，体现如下

- 结构体里面可以直接包含函数，作为结构体的成员存在
- 结构体名可以直接用来定义变量
- 在结构体成员函数里面可以直接访问结构体成员变量

```
#include <iostream>


using namespace std;


class Student
{
        string name;
        int  age;
        int  score;
        void playGame(const string &game);
};


void Student::playGame(const string &game)
{
        cout << "Name : " << name  << endl;
        cout << "Score: " << score << endl;
        cout << "Age  : " << age   << endl;
        cout << "Game : " << game  << endl;


        return;
}


int main(void)
{
     Student stu;


     stu.name  = "xiaoming";
     stu.score = 100;
     stu.age   = 10;


     stu.playGame("king");


     return 0;
}
```

编译代码有如下错误:

```
student.cpp: In function ‘int main()’:
student.cpp:6:9: error: ‘std::string Student::name’ is private
  string name;
         ^
student.cpp:26:6: error: within this context
  stu.name  = "xiaoming";
      ^
```

```
student.cpp:8:7: error: ‘int Student::score’ is private
  int  score;
       ^
student.cpp:27:6: error: within this context
  stu.score = 100;
      ^
student.cpp:7:7: error: ‘int Student::age’ is private
  int  age;
       ^
student.cpp:28:6: error: within this context
  stu.age   = 10;
      ^
student.cpp:12:6: error: ‘void Student::playGame(const string&)’ is private
 void Student::playGame(const string &game)
      ^
student.cpp:30:21: error: within this context
  stu.playGame("king");
```

```
class 类名称
{
    public:
         公有成员(外部接口)

    private:
         私有成员

    protected:
         保护成员
};
```

用来描述一个对象的属性信息, 与一般的变量声明相同, 但需要将它存放在类的声明体中

- 用来描述一个对象的行为动作，与一般的函数声明相同，但需要将它存放在类的声明体中
- 在类中说明函数原型，在类外给出函数体实现，并在函数名前使用 ** 类名::** 加以限定
- 也可以直接在类中给出函数体，形成内联成员函数
- 允许声明重载函数和带默认值参数的函数
- public(公有成员), 类的成员函数以及该类定义的对象都能够访问这些公有成员
- private(私有成员)****, 只能被类的成员函数直接访问，类外定义的对象不可以直接访问
- protected(保护成员), 只能被该类的成员函数和其派生类的成员函数访问 (继承的时候讲解)

C++ 对 struct 关键字扩展了其功能, 和 class 的功能几乎等价, 区别主要体现在默认访问权限上, struct 的成员默认访问权限是 public, class 的成员默认访问权限是 private

编写一个类描述时间对象，可以设置小时、分钟、秒，也可以输出设置的时间

全文完