

Task 2 使用说明

1. 作业提交说明

大家好，这是我们的第二次离线作业了，希望大家对整个框架、流程以及评分规则有了一定的熟悉。

离线方式下的作业完成和提交流程：

Step1. 仔细阅读任务说明中的每一小节。按照说明中的内容，在充分理解和掌握课上授课老师所讲的知识点的基础上，阅读任务说明。

Step2. 每一小节任务说明都包含有必要的背景信息、技术信息、代码信息等多项内容。请按照要求，打开提供的框架程序，在指定的地方书写代码，完成任务要求。

Step3. 每完成一小节任务，请在本机的（自己的）IDE（编译器）上完成编译流程，并可执行检查结果是否正确。

Step4. 在提供的框架程序中，寻找 submit.exe 文件，点击 submit 文件，即可完成离线的提交。按照任务说明，可以每完成一小节任务提交一次（强烈推荐）；也可以完成所有任务后提交一次（强烈不推荐）。

Step5. 完成所有任务后，请在指定的 deadline 之前，将规定的文件发送、拷贝、提交到指定的地点（详细内容请见任务说明）。

该流程如有问题，可以在上课的时候向授课老师反馈。

注意事项：

1. 采用离线形式对大家的程序进行检查的目的是为了节约更多的课上时间，回答大家的问题。所以请仔细阅读以下注意事项，并严格遵守。
2. **请不要随意删除、修改系统提供的程序框架里的代码**，否则可能会造成提交困难或者影响最终成绩。
3. 在完成任务的过程中，允许和鼓励大家互相交流，相互讨论完成。我们并没有严格

要求必须独立完成。但是，严禁抄袭。但凡有抄袭现象，将会给予严厉的处罚（包括抄袭者和被抄袭者）。

4. 提交过程可以多次执行，我们只记最后一次的成绩。鼓励大家在执行过程中多次通过 submit.exe 程序验证自己的代码的正确性。
5. 我们本学期统一采用 vs2013 作为开发平台，请大家自行安装。

推荐查询如下网址：

http://zhidao.baidu.com/link?url=WdrqT5zk6zYtCg_gstQPeRklVecvEAXAj0vS59zk1Dx_yLpk5R4WopWgionMEc02oRlaesbyRKmRbl4fxuWi6cvUcnK07h1rw0Is-kmLNfS

仅仅是推荐，一切后果，本课程概不负责。

6. 如果在使用过程中，有任何问题，可以与我以及课程助理联系：

岳朋飞（计工本、计信本）18363010229@163.com

赵彦会（卓计本）18366138945@163.com

我们只负责解决框架问题、知识点问题；技术问题会在上机课时给大家解决。

2. 任务说明

第二次任务的目的是让大家学会创建类，以及通过创建类，联系对于对象的创建和调用。

2.1 前言

我曾养了一只可爱的猫（Cat），她又能吃（eat）又能喝（drink）不爱运动（sport），却特能睡（sleep）。吃饱之后就睡觉，当然会长胖。虽然适当运动有助于减肥，但舒适的生活使她常常忘记了锻炼身体。我整天提心吊胆地喂她，吃得太多或喝的太多会撑死；当然了，不吃不喝就会饿死……这娇气的猫。

这只猫多多少少会给人一些启发。我准备编个程序纪念她，没事儿的时候拿出来运行一下，就可以让她吃饭、喝水、运动、睡觉……有意思吧！

等一下！要编程了。

下面我们就从测试程序开始，考虑一下，我们怎样“对待”一只猫。

“可是，我们的猫在哪里呢？”

别着急，这叫测试驱动开发（Test Driven Development, TDD）。

```
void CastTest()

{

    Cat cat("MaoMi");    //那只可爱的猫叫 MaoMi

}
```

我把 cat 看成一个对象，这意味着要编写一个 class Cat，还有一个构造函数可以给它取个名字。这里特别提醒一下，很多初学者可能会忽视了字母的大小写。注意，这里的 Cat 是类的名字，而 cat 是对象的名字。如果敲成相同的词，你就会得到一大堆恼人的编译错误和警告。不过，你可以试一试。

```
class Cat

{

public:

    Cat(const string& name); //为了记下猫的名字，我需要一个记录名字的成员。

private:

    String name;

};
```

写出这个构造函数，先编写测试，然后编写程序使测试运行。这正是 TDD 的一般过程。

```
Cat::Cat(const string& name) /*能区分出成员和参数吗? */

{ }
```

这些虽然通过编译了，但不足以让我们感兴趣。尝试着增加更多测试。

Question1:

我们想和 MaoMi 打个招呼，看看她的情况，可能会这样：

```
void CatTest()

{

    Cat cat("MaoMi");

    cat.talk();

}
```

我们希望得到的情况是：当执行 `cat.talk()` 时，能够回答出：

“Miao~wu, I'm MaoMi” // MaoMi 是猫的名字，因为我的猫就叫做 MaoMi，
请不要随便改名哟。

在 CCat 类中的相应地方添加代码，然后就可以提交 question 1 了。

Question 2:

先讨论吃的问题。“猫以食为天”啊。

```
cat.eat(10);          //吃 10 点食物
```

看出来，需要一个 public member function; `Cat::eat(int food)`。

在 class Cat 中增加：

```
public:

    void eat(int f);          //吃食物 f 点（1 单位食物为 1 点）
```

为了知道猫总共吃了多少食物，再增加一个 data member; `food`。

```
private:

    int food; //已经吃下的食物
```

好了，想好怎么实现 `Cat::eat()` 了么？当猫吃下 `f` 点食物之后，肚子里食物相应增加了。

```
void Cat::eat(int f)
```

```
{  
  
    food += f;  
  
}
```

当我们执行下列语句时，结果会是什么呢？

```
void CatTest()  
  
{  
  
    Cat cat("MaoMi");  
  
    cat.talk();  
  
    cat.eatd(10);                //吃 10 点食物  
  
    eat.talk();  
  
}
```

现在可以仿照 eat（）很容易写出 drink（）。然后，请运行测试程序：

```
void CatTest()  
  
{  
  
    Cat cat("MiaoMi");  
  
    cat.talk();  
  
    cat.eat(10);                //吃 10 点食物  
  
    cat.talk();  
  
    cat.drink(5);                //喝 5 点水  
  
    cat.talk();  
  
}
```

写完 eat 函数、drink 函数，就可以 submit 你的 question 2 了。

Question 3:

想一下，吃饱了就睡，肯定会长胖的。长胖是什么意思？长胖就是长肉，其实质是吃进去的食物转化成了自己的体重。

这就来了问题，我们需要一个数据成员表示体重（weight）。至于其初值，可以设定为一个合理的正值。（本程序都用 80 作为初始值）

```
class Cat

{

    ... ..

private:

    int weight;

};

Cat::Cat(const string& name)

    :name(name), food(0), water(0), weight(80)

{}

void Cat::talk()

{

    ... ..

    cout << "weight=" << weight << endl;

}
```

然后，我们假设在猫睡觉的时候，每过一段时间，如 1 小时，都会有 1 点食物和 1 点水转化成 1 点体重。

这样一来，猫睡觉的样子像这样：

```
class Cat
```

```

{

public:

    ... ..

    int sleep(int h);        //睡 h 小时

};

void Cat::sleep(int h)

{

    food -= 1 * h;          //消耗食物

    water -= 1 * h;         //消耗水

    weight += 1 * h;        //体重增加了

    //仅此而已么?

}

```

上面的程序好像完整了，又好像缺点什么。还是让我们先运行测试吧。

```

void CatTest()

{

    Cat cat("MaoMi");

    cat.talk();

    cat.eat(10);             //吃 10 点食物

    cat.talk();

    cat.drink(5);           //喝 5 点水

    cat.talk();

    cat.sleep(4);           //睡 4 小时
}

```

```
        cat.talk();

        cat.sleep(2);

        cat.talk();

    }
```

结果会如何呢？此处也许你可以 submit 一下了，如果成功的话，那么恭喜你，你可以跳过下面的内容，如果没成功，那么，你就需要思考下面的问题了。

water=-1, 这是不可能的，想一下，在猫睡着的时候，胃里的水都消耗尽了会怎样？

她长时间缺水 and 食物，只是蒙头大睡，肯定不会再长胖了，应该会变得消瘦些。

于是，我可以简单地从体重中扣除一部分，用来抵消成为负数的水。

那么程序应该如何改呢？额……，我想这就是你的工作了。

解决完这个问题后，就可以通过 submit，提交你的 question3 了。

Question 4:

不过要想迅速减肥，还是要做一些健康的运动。我们假设，每运动 1 个小时，会额外消耗 2 点食物和水。也就是说，运动过程中每 1 小时消耗 3 点食物和 3 点水，但只增加 1 点体重。当食物和水不足的时候，体重可能会加速下降。

运动的程序类似睡觉，请完成运动的程序。

运行完整的测试：

```
void CatTest()

{

    Cat cat("MaoMi");

    cat.talk();

    cat.eat(10);    //吃 10 点食物

    cat.talk();
```



```

        cat.drink(5);    // 喝 5 点水

        cat.talk();

        cat.sleep(4);    //睡 4 小时

        cat.talk();

        cat.sport(1);    //运动 1 小时

        cat.talk();

    }

```

好了完成 sport 函数后，可以通过 submit 提交你的 question4 了。

Question 5:

随着时间的流逝，小猫会一个小时一个小时地长大。我们能否记录她的年龄呢（这样会增加一些趣味性，比如两人可以比赛谁喂的猫更加健康长寿。）

应该很简单。增加数据成员 age 记录她的年龄（为简化，按小时计算），并增加初始化，在 sleep（）和 sport（）时，同时修改 age。

重写 talk（）如下：

```

void Cat::talk()

{

    cout << name << "(age=" << age << "weight=" << weight << "food="

        << food << "water=" << water << ")" << endl;

}

```

接下来你可以自己测试一下你的年龄情况了：

这个并不计入成绩，所以请大家随便了。

最后留一个思考题：

思考题

最后，留下一个思考题。

小猫每天作息时间表如下：

6: 00am 起床，晨练 1 小时

7: 00am 自助早餐（食物 4 点，水 6 点），睡觉 4 小时

11: 00am 午餐（食物 6 点，水 4 点），活动 2 小时

1: 00pm 睡觉 4 小时

5: 00pm 跑步 1 小时

6: 00pm 晚餐（食物 10 点，水 10 点），活动 1 小时

7: 00pm 睡觉 6 小时

1: 00am 活动 1 小时，捉住一只老鼠（食物 6 点，水 6 点）吃掉

2: 00am 睡觉 4 小时

问：小猫在一个月（30 天）以后的状况如何？

请大家通过程序完成这个思考题，然后通过 submit 提交 Question 5.