

## CS1501 程序设计思想与方法 (C++)

### Homework 11

2024 年 12 月 12 日 至 2024 年 12 月 18 日

#### 内容概要:

- 练习编写老师布置的编程题;
- 练习编写《实验指导》书上实验 14 中指定的编程题;
- 严格遵循运行示例显示程序输出 (包括文字、标点、空格等所有内容)。

#### 具体内容:

1、设计一个细胞群模拟程序 (很假), 部分程序代码已提供, 请将其中注明需填写代码的地方填写完整。注意以下要求:

注 1: 不能在**没有注明“此处需填写代码”**的地方添加代码。

注 2: 用户输入的 `init_size` 是范围在`[10,100]`内的整数。

注 3: `Cell` 类中包含 2 个私有数据成员与 8 个公有成员函数, 没有公有数据成员或私有成员函数。2 个私有数据成员分别为 `size` 与 `resource`, 都是整型数; 8 个成员函数的名称为 `Cell`、`~Cell`、`grow`、`kill`、`get_size`、`get_resource` (虽然有 8 个函数, 但只有 6 个函数名)。要求 **`grow`、`get_size`、`get_resource` 三个函数均没有参数; `kill` 函数不访问 `resource` 且是唯一包含 `delete` 的函数。公有成员函数不能返回私有数据成员的指针或引用。**

注 4: 整个环境中最初的 `resource` 为 500。

注 5: 在 `while` 循环的每个周期中进行以下操作: 初始细胞与每个已经诞生的普通细胞都会成长一次; 一个普通细胞的成长会使其个体的 `size` 加 1, 而整个环境中的 `resource` 减 1; 初始细胞的成长则不会改变其 `size`, 但仍会使环境 `resource` 减 1; 如果一个普通细胞的 `size` 大于 5, 那么它将会创造一个新的普通细胞, 该细胞的 `size` 为原细胞的一半 (取整, 舍去小数), 而环境中的 `resource` 会减少相当于新细胞 `size` 的量。

注 6: 在 **while** 循环的每个周期结束之前, 如果环境中 **size** 最大的一个细胞的 **size** 大于 10, 那么初始细胞会将它杀掉。如果两个普通细胞一样大且都大于 10, 那么只杀掉先出生的那个细胞。当一个细胞被杀掉后, 环境中的 **resource** 增加相当于该细胞的 **size** 的量。

注 7: 当环境中的 **resource** 减少到 0 或负数的时候, 或当第 100 个普通细胞被创造之后, 中断 **while** 循环。

注 8: 每当有细胞成长时, 输出: “Initial Cell grows, remaining resource = XX” 或 “Cell YY grows, remaining resource = XX”, 其中 XX 为当前 **resource** 的值, YY 为相应的普通细胞的编号。每当有细胞被创造时, 输出 “Cell YY is born, remaining resource = XX”。每当有细胞被杀掉时, 输出 “Cell YY is killed, remaining resource = XX”。

```
#include <iostream>
#include <cassert>
using namespace std;

class Cell{
    // 此处需填写代码
};

int main()
{
    int init_size, id=1, max_id=0;
    Cell *cells[100] = {0};

    cout << "Initial Cell Size: ";
    cin >> init_size;
    const Cell *init_cell = new const Cell(init_size);
    cout << "Initial Cell is born, remaining resource = " << Cell::get_resource() << endl;

    cells[0] = new Cell(*init_cell);
    cout << "Cell 0 is born, remaining resource = " << Cell::get_resource() << endl;
    while(true) {
        // 此处需填写代码
        if(init_cell->kill(cells[max_id])){
            assert(cells[max_id] == NULL);
            // 此处需填写代码
        }
    }
    return 0; //不要求在 main 函数结束前释放所有动态分配的内存空间
}
```

// 此处需填写代码 (Cell 类的实现)

要求按以下运行示例来显示程序的运行结果：

```
Initial Cell Size: 10
Initial Cell is born, remaining resource = 490
Cell 0 is born, remaining resource = 485
Initial Cell grows, remaining resource = 484
Cell 0 grows, remaining resource = 483
Cell 1 is born, remaining resource = 480
Initial Cell grows, remaining resource = 479
Cell 0 grows, remaining resource = 478
Cell 2 is born, remaining resource = 475
Cell 1 grows, remaining resource = 474
（省略中间很多行）
Cell 15 grows, remaining resource = 10
Cell 94 is born, remaining resource = 6
Cell 16 grows, remaining resource = 5
Cell 95 is born, remaining resource = 1
Cell 17 grows, remaining resource = 0
```

```
Initial Cell Size: 100
Initial Cell is born, remaining resource = 400
Cell 0 is born, remaining resource = 350
Initial Cell grows, remaining resource = 349
Cell 0 grows, remaining resource = 348
Cell 1 is born, remaining resource = 323
Cell 0 is killed, remaining resource = 374
Initial Cell grows, remaining resource = 373
Cell 1 grows, remaining resource = 372
Cell 2 is born, remaining resource = 359
Cell 1 is killed, remaining resource = 385
（省略中间很多行）
Cell 16 grows, remaining resource = 14
Cell 79 is born, remaining resource = 10
Cell 17 grows, remaining resource = 9
Cell 80 is born, remaining resource = 5
Cell 18 grows, remaining resource = 4
Cell 81 is born, remaining resource = 0
```

2、完成以下程序代码（不能修改原有代码，只能在写有注释“填写代码”的位置按注释要求添加代码），使该程序具备以下功能：

病毒 **v** 每个循环周期吃掉细胞 **c** 的一部分，该部分的大小为 **amount**（**amount** 由用户输入，但当 **amount** 大于细胞 **c** 当前的 **size** 时，**amount** 改为细胞 **c** 当前的 **size**），使细胞 **c** 的 **size** 减少 **amount**、病毒 **v** 的 **life** 增加 **amount**。

```
#include <iostream>
using namespace std;

//填写代码，只有一行
class Virus{
    int life;
public:
    Virus(int x):life(x){}
    //填写代码，只有一行
    bool time_pass(){life --; return (life?true:false);}
};

class Cell{
    //填写代码，只有一行
private:
    int size;
public:
    Cell(int x):size(x){}
    void time_pass(){if(size) size ++;}
};

int main() {
    int cell, virus, amount, cnt;
    cout << "Input cell, virus, amount: "; cin >> cell >> virus >> amount;
    Cell c(cell); Virus v(virus);
    for(cnt=0; ; cnt++){
        v.eat(c, amount);
        if( ! v.time_pass() ) break;
        c.time_pass();
    }
    cout << "Output: Time = " << cnt << endl;
    return 0;
}

//填写代码，定义一个成员函数
```

要求按以下运行示例来显示程序的运行结果：

```
Input cell, virus, amount: 10 2 3
Output: Time = 15
```

```
Input cell, virus, amount: 100 1 4
Output: Time = 132
```

3、完成《实验指导》上实验 14 的编程题（1）。要求其 main 函数如下：

```
int main() {
    Point p1, p2;
    cin >> p1;
    p2 = -p1;
    cout << p2;
    return 0;
}
```

要求按《实验指导》书上的运行示例来显示程序的运行结果。

4、在 Homework10 第 4 题的 SkillList 类的基础上，添加一个下标运算符重载函数，使该类对象的下标运算用于查找一个技能是否已被学会。

要求其 main 函数如下：

```
int main() {
    // 直到 while 循环为止的部分和 Homework10 第 4 题一致
    int select;
    cout << "Select 1-Show learned skills, 2-Search a target skill: ";
    cin >> select; cin.get();
    if (select == 1) {
        cout << "Learned skills: ";
        pMember->display();
    } else if (select == 2) {
        char skillTarget[20];
        cout << "Target skill: "; cin.getline(skillTarget, 20);
        cout << "Search result: ";
        cout << ((*pMember)[skillTarget] ? "Yes" : "No") << endl;
    }
    delete pMember;
    return 0;
}
```

要求按以下运行示例来显示程序的运行结果：

```
Max number of skills: 2
Construct SkillList: None
Learn a new skill: Heal
Learn a new skill: Cure
Learn a new skill: Fire Ball
Select 1-Show learned skills, 2-Search a target skill: 1
Learned skills: Cure, Heal
Destruct SkillList: Heal, Cure
```

```
Max number of skills: 3
Construct SkillList: None
Learn a new skill: Heal
Learn a new skill: Cure
Learn a new skill: Fire Ball
Learn a new skill: Ice Arrow
Select 1-Show learned skills, 2-Search a target skill: 2
Target skill: Fire Ball
Search result: Yes
Destruct SkillList: Heal, Fire Ball, Cure
```

```
Max number of skills: 1
Construct SkillList: None
Learn a new skill: Cure
Learn a new skill: Heal
Select 1-Show learned skills, 2-Search a target skill: 0
Destruct SkillList: Cure
```

## 提交作业的方式与要求：

本次作业共有 4 题：

1. 本文档中的第一道编程题（模拟细胞分裂）；
2. 本文档中的第二道编程题（细胞与病毒）；
3. 《实验指导》书上实验 14 的编程题（1）；
4. 本文档中的第三道编程题（培养战斗角色的游戏之技能列表之下标运算）。

请将每道编程题的程序代码所在的 `main.cpp` 文件重新命名，命名规则为

“HW11-姓名-题号.cpp”。

例如张三同学编写的第 3 题（即《实验指导》书上实验 14 的编程题（1））的 `main.cpp` 文件应重命名为“HW11-张三-3.cpp”。

**不按要求给 `main.cpp` 重命名的作业得零分。**

将重命名后的.cpp 文件在以下网页上提交：

<https://jbox.sjtu.edu.cn//B1scw8>

注意！请勿上传除了.cpp 文件之外的任何其它文件。

**本次作业提交截止至 12 月 18 日晚 23:59，过期未交的作业得零分。**