



§ 17. 继承与派生 – 画内存图并分析

要求：分析下面的程序为什么在VS下运行会出错

```
#include <iostream>
using namespace std;
class A {
private:
    char* s;
public:
    int a;
    A() { s = new char[20]; }
    ~A() {
        cout << "A析构" << endl;
        delete s;
    }
};
class B :public A {
public:
    int b;
    ~B() { cout << "B析构" << endl; }
};
int main()
{
    B b;
    b.a = 15;
    A a1(b);    //复制构造
    A a2;
    a2 = b;     //赋值
}
```

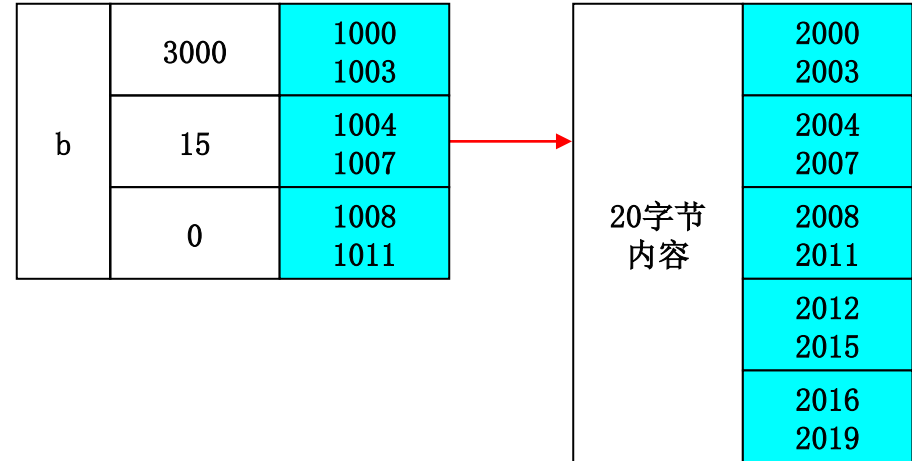
- ★ main函数中红色的三句，每句均要画出当前的内存图（为方便阅读，一句可以多页），分析为什么会出错
- ★ 可参考第13模块 P. 90-92、P. 113-115，第16模块 P. 8-P. 14
- ★ 不允许手写、手写后贴图，转换为pdf后在“文档作业”中提交（2024. 12. 19前）



§ 17. 继承与派生 – 画内存图并分析

要求：分析下面的程序为什么在VS下运行会出错

```
#include <iostream>
using namespace std;
class A {
private:
    char* s;
public:
    int a;
    A() { s = new char[20]; }
    ~A() {
        cout << "A析构" << endl;
        delete s;
    }
};
class B :public A {
public:
    int b;
    ~B() { cout << "B析构" << endl; }
};
int main()
{
    B b;
    b.a = 15;
    A a1(b);    //复制构造
    A a2;
    a2 = b;     //赋值
}
```



★ main函数中红色的三句，每句均要画出当前的内存图（为方便阅读，一句可以多页），分析为什么会出错

★ 可参考第13模块 P. 90-92、P. 113-115，第16模块 P. 8-P. 14

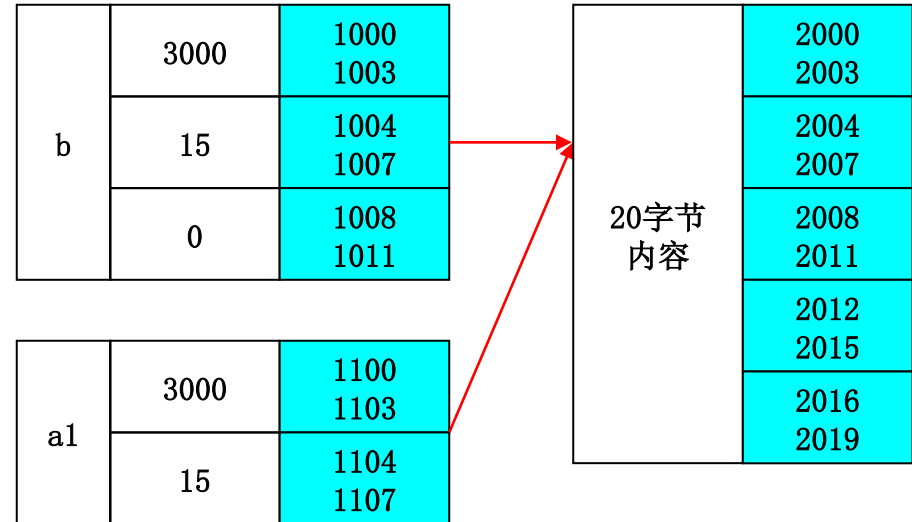
★ 不允许手写、手写后贴图，转换为pdf后在“文档作业”中提交（2024. 12. 19前）



§ 17. 继承与派生 – 画内存图并分析

要求：分析下面的程序为什么在VS下运行会出错

```
#include <iostream>
using namespace std;
class A {
private:
    char* s;
public:
    int a;
    A() { s = new char[20]; }
    ~A() {
        cout << "A析构" << endl;
        delete s;
    }
};
class B :public A {
public:
    int b;
    ~B() { cout << "B析构" << endl; }
};
int main()
{
    B b;
    b.a = 15;
    A a1(b);    //复制构造
    A a2;
    a2 = b;     //赋值
}
```



★ main函数中红色的三句，每句均要画出当前的内存图（为方便阅读，一句可以多页），分析为什么会出错

★ 可参考第13模块 P. 90-92、P. 113-115，第16模块 P. 8-P. 14

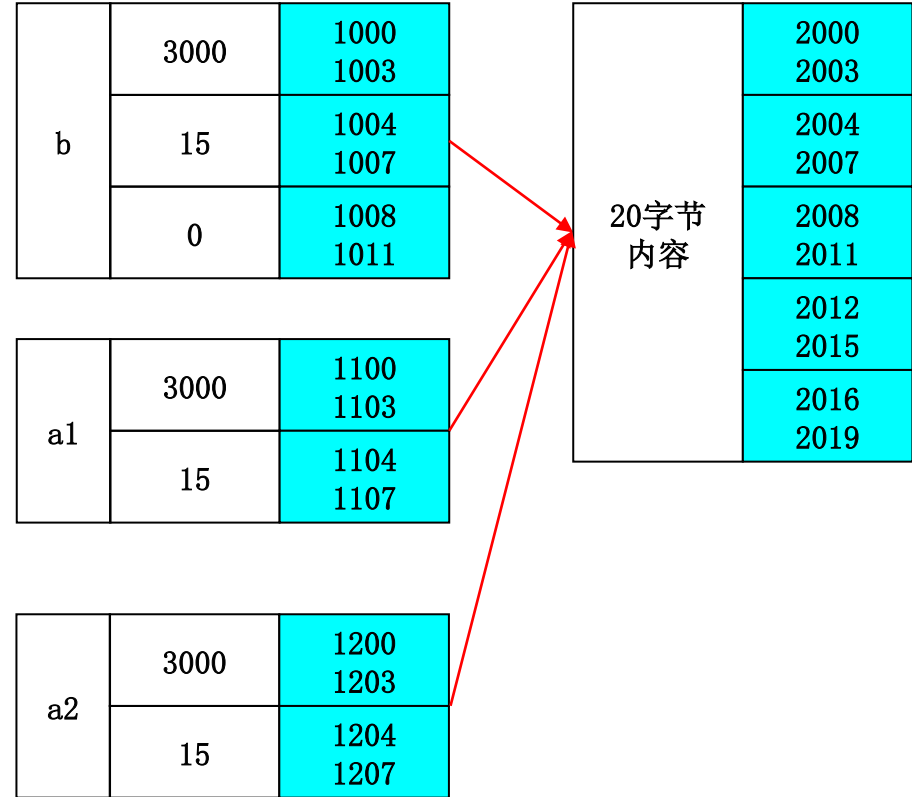
★ 不允许手写、手写后贴图，转换为pdf后在“文档作业”中提交（2024. 12. 19前）



§ 17. 继承与派生 – 画内存图并分析

要求：分析下面的程序为什么在VS下运行会出错

```
#include <iostream>
using namespace std;
class A {
private:
    char* s;
public:
    int a;
    A() { s = new char[20]; }
    ~A() {
        cout << "A析构" << endl;
        delete s;
    }
};
class B :public A {
public:
    int b;
    ~B() { cout << "B析构" << endl; }
};
int main()
{
    B b;
    b.a = 15;
    A a1(b);    //复制构造
    A a2;
    a2 = b;     //赋值
}
```



★ main函数中红色的三句，每句均要画出当前的内存图（为方便阅读，一句可以多页），分析为什么会出错

★ 可参考第13模块 P. 90-92、P. 113-115，第16模块 P. 8-P. 14

★ 不允许手写、手写后贴图，转换为pdf后在“文档作业”中提交（2024. 12. 19前）

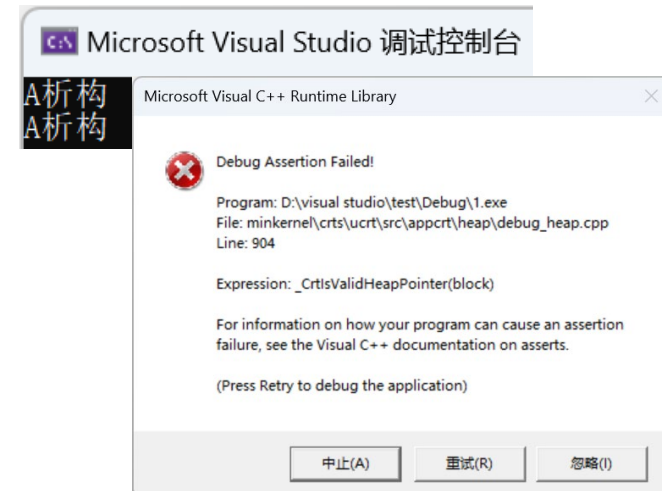


§ 17. 继承与派生 – 画内存图并分析

要求：分析下面的程序为什么在VS下运行会出错

```
#include <iostream>
using namespace std;
class A {
private:
    char* s;
public:
    int a;
    A() { s = new char[20]; }
    ~A() {
        cout << "A析构" << endl;
        delete s;
    }
};
class B : public A {
public:
    int b;
    ~B() { cout << "B析构" << endl; }
};
int main()
{
    B b;
    b.a = 15;
    A a1(b);    //复制构造
    A a2;
    a2 = b;     //赋值
}
```

有一个潜在的问题是在执行 `A a1(b);` 时调用了 `A` 类的复制构造函数，但由于 `A` 类没有提供复制构造函数，它会使用编译器默认生成的复制构造函数，这会导致浅拷贝问题。对于语句 `A a1(b);` 和 `a2 = b;` 的对象赋值操作，只能简单的把指针 `b.s` 的值浅拷贝给 `a1.s` 和 `a2.s` 导致三者指向同一内存空间。调用析构函数时，造成对同一内存空间的多次释放，从而引起弹窗报错。



为避免浅拷贝，手动实现了赋值操作符重载，确保在赋值时正确地释放旧的内存并分配新的内存。

- ★ `main`函数中红色的三句，每句均要画出当前的内存图（为方便阅读，一句可以多页），分析为什么会出错
- ★ 可参考第13模块 P. 90–92、P. 113–115，第16模块 P. 8–P. 14
- ★ 不允许手写、手写后贴图，转换为pdf后在“文档作业”中提交（2024. 12. 19前）