Kỹ Thuật Lập Trình

Khóa Cuối kì

KTLT-CK-HK232

Thảo luận BTL môn KTLT, DSA, NMLT, PPL ....
https://www.facebook.com/groups/211867931379013

Tp. Hồ Chí Minh, Tháng 5/2024

Mục lục

1 Lập trình hướng đối tượng

1. Kể 4 tính chất của OOP
2. Tìm các thuộc tính và static có thể truy cập của class A, trong các *SCOPE*

```
class A{
    static int a0;
    int a1;
protected:
    int a2;
public:
    int a3;
    static int a4;

    int test1(const A& others){ /* SCOPE 1 */}
};

class B : public A{
    /* SCOPE 3 */
};

class C : public B{
    /* SCOPE 4 */
};

class D : A{
    /* SCOPE 5 */
};

class E : protected D{
    /* SCOPE 6 */
};

int test2(const A& others){ /* SCOPE 7 */}

int main(int argc, char **argv)
{
    A a; /* SCOPE 8 */
    B b; /* SCOPE 9 */
    C c; /* SCOPE 10 */
    D d; /* SCOPE 11 */
    E e; /* SCOPE 12 */
    return 1;
}
```

3. Tìm các thuộc tính có thể truy cập trong các prama của hàm

```
class B;
class C;

class A{
private:
    int a1;
protected:
```

```
    int a2;
public:
    int a3;

    friend void test1(const A& a, const B& b, const C& c);

    void test2(const B& b){ /* SCOPE 1 */}
    friend class B;
};

void test1(const A& a, const B& b, const C& c){ /* SCOPE 2 */}

class B{
private:
    int b1;
protected:
    int b2;
public:
    int b3;

    void test3(const A& a, const C& c){ /* SCOPE 3 */}
};

class C : public A{
private:
    int c1;
protected:
    int c2;
public:
    int c3;
    void test4(const A& a, const B& b){ /* SCOPE 4 */}
};
```

4. kết quả khi khai báo và rời khỏi tầm vực của khai báo đó

```
A a(1), a;
B b(1), b;
C c(1), c;

A* b = new B(1);
delete b;

B* c = new C;
delete c;
```

```
class A{
public:
    A(int i){cout << i;}
    A() = default;
    ~ A(){cout << "A";}
};

class B : public A{
public:
    B(int i):A(i+1){cout << i;}
```

```
    B(){cout << "B";}
    ~ B(){cout << "B";}
};

class C : public B{
public:
    C(int i):B(i+1){cout << i;}
    C() = default;
    ~C() = default;
};
```

5. Hãy chỉnh sửa các hàm print có thể thêm virtual, =0 hoặc xóa nó sao cho thõa các điều kiện sau

```
class A{
public:
    void print();
};

class B : public A{
public:
    void print();
};

class C : public B{
public:
    void print();
};
```

- A a; không khai báo được
- B b; không khai báo được
- C c; không khai báo được
- A b = new B(); gọi hàm print của class A
- A b = new B(); gọi hàm print của class B
- B b = new B(); gọi hàm print của class A
- B b = new B(); gọi hàm print của class B
- A c = new C(); gọi hàm print của class A
- A c = new C(); gọi hàm print của class B
- A c = new C(); gọi hàm print của class C
- B c = new C(); gọi hàm print của class A
- B c = new C(); gọi hàm print của class B
- B c = new C(); gọi hàm print của class C
- C c = new C(); gọi hàm print của class A
- C c = new C(); gọi hàm print của class B
- C c = new C(); gọi hàm print của class C

6. hãy hiện thực hàm print của C sao cho gọi được đến A (gợi ý gọi trong hàm C)

```
class A{
public:
    virtual void print(){cout << "A";}
};

class B : public A{
public:
    virtual void print() = 0;
};

class C : public B{
public:
    //TODO: implement print
};


int main(int argc, char **argv){
    A* c = new C();
    c->print();
}
```

7. kết quả sau cho hệ thống 64-bit

```
class A{
public:
    virtual void print() = 0;
    void print(int i){cout << "C";}
};

class B : public A{
public:
    void print(int i){cout << "B";}
    void print(){cout << "A";}
};


int main(int argc, char **argv){
    A* b = new B();
    b->print(1);
    b->print();
    dynamic_cast<B*>(b)->print(1);
    dynamic_cast<B*>(b)->print();
}
```

8. kết quả sau

```
class A{
public:
    long long a;
    char b;
    int c;
```

```
    static int s;
    virtual void print() = 0;
};

class B : public A{
public:
    int d;
    long long e;
    void print(){cout << "A";}
};

int main(int argc, char **argv){
    cout << sizeof(void*) << " " << sizeof(A) << " " << sizeof(B) << endl;
}

```

9. kết quả

```
class A{
public:
    static int s;
};

class B : public A{
public:
    B(){s++;}
    ~B(){A::s--;}
};

int A::s = 1;
int main(int argc, char **argv){
    B b[10]; cout << A::s << endl;
    A* c = new A[20]; cout << B::s << endl;
    delete[] c;
    B* d = new B[20]; cout << B::s << endl;
    delete[] d;
    cout << b[0].s << endl;
}
```

10. Phân biệt stuct với class

Thảo luận BTL môn KTLT, DSA, NMLT, PPL ....
https://www.facebook.com/groups/211867931379013

- Lớp BTL1 + GK + LAB + Lý thuyết + Harmony của môn DSA HK232
- Lớp BTL2 + CK + LAB + Lý thuyết + Harmony của môn DSA HK232
- Lớp BTL1 + Lý thuyết + Harmony của môn KTLT HK232
- Lớp BTL2 + Lý thuyết + Harmony của môn KTLT HK232
- Lớp CK + LAB + Harmony của môn KTLT HK232
- Lớp BTL1 + BTL2 + GK + Harmony của môn PPL HK232
- Lớp BTL3 + BTL4 + CK + Harmony của môn PPL HK232

CHÚC CÁC EM HỌC TỐT