

PA2 2.test  
progtest

Martin Franc

19. března 2014

**Otázka 1:** Co zobrazí následující program?

```
class A {
    public:
        A ( int x = 21 ) { m_X = x; cout << x; }
    private:
        int m_X;
};
class B{
    public:
        B ( int x ) { m_A = A ( x ); }
    private:
        A m_A;
};
...
B test( 35 );
```

**Program zobrazí 2135.**

---

**Otázka 2:** Co zobrazí následující program?

```
class A{
    public:
        A ( int x = 4 ) { m_X = x; cout << x; }
    private:
        int m_X;
};
class B{
    public:
        B ( int x ) { m_A = new A ( x ); }
    private:
        A * m_A;
};
...
B test( 93 );
```

**Program zobrazí 93.**

---

**Otázka 3:** Co zobrazí následující program?

```
class A{
    public:
        A ( int x ) { m_X = x; cout << x; }
    private:
        int m_X;
};
class B{
    public:
        B ( int x ) : m_A ( x ) { }
    private:
        A m_A;
};
...
B test( 57 );
```

**Program zobrazí 57.**

---

**Otázka 4:** Co zobrazí následující program?

```
class A{
    public:
        A( int x ){ m_X = x; cout << x; }
    private:
        int m_X;
};
class B{
    public:
        B( int x ){ m_A = A(x); }
    private:
        A m_A;
};
...
B test( 28 );
```

**Program nepůjde zkompilevat.**

---

**Otázka 5:** Co zobrazí následující program?

```
class A{
    public:
        A ( int x ) { m_X = x; cout << x; }
    private:
        int m_X;
};
class B{
    public:
        B ( int x ) { m_A = new A ( x ); }
    private:
        A * m_A;
};
...
B test( 15 );
```

**Program zobrazí 15.**

---

```
class A{
    public:
        A ( int x ) { m_X = x; cout << x; }
    private:
        int m_X;
};
class B{
    public:
        B ( int x ) : m_A ( new A ( x ) ) { }
    private:
        A * m_A;
};
...
B test( 31 );
```

**Program zobrazí 31.**

---

**Otázka 7:** Co zobrazí následující program?

```
class CInt {
public:
    CInt ( int x ) : m_X ( new int ( x ) ) {}
    CInt ( const CInt & src ){ m_X = new int ( *src.m_X ); }
    ~CInt(void){ delete m_X; }
    void operator = ( int x ) { *m_X = x; }
    void print ( void ) const { cout << *m_X; }
private:
    int *m_X;
};
int main( void ) {
    CInt a ( 12 ), b ( 42 ), c = a;
    a = 17;
    c . print ();
    return 0;
}
```

Zobrazí se 12.

---

**Otázka 8:** Co zobrazí následující program?

```
class CInt{
public:
    CInt ( int x ) : m_X ( new int ( x ) ) {}
    CInt ( const CInt & src ) { m_X = new int ( *src . m_X ); }
    ~CInt ( void ) { delete m_X; }
    void operator = ( int x ) { *m_X = x; }
    void print ( void ) const { cout << *m_X; }
private:
    int * m_X;
};
int main ( void ){
    CInt a ( 49 ), b ( 6 ), c = a;
    a = b;
    c . print ();
    return 0;
}
```

Program půjde zkompilovat, ale může spadnout/spadne.

---

**Otázka 9:** Co zobrazí následující program?

```
class CInt{
public:
    CInt ( int x ) : m_X ( new int ( x ) ) {}
    CInt ( const CInt & src ) { m_X = new int ( *src . m_X ); }
    ~CInt ( void ) { delete m_X; }
    void operator = ( int x ) { *m_X = x; }
    void print ( void ) const { cout << *m_X; }
private:
    int * m_X;
};
int main ( void ){
    CInt a ( 74 ), b ( 20 ), c = a;
    a = 54;
    c . print ();
    return 0;
}
```

Zobrazí se 74.

---

**Otázka 10:** Co zobrazí následující program?

```
class CInt{
public:
    CInt ( int x ) : m_X ( new int ( x ) ) {}
    CInt ( const CInt & src ) { m_X = src . m_X; }
    ~CInt ( void ) { delete m_X; }
    void operator = ( int x ) { *m_X = x; }
    void print ( void ) const { cout << *m_X; }
private:
    int * m_X;
};
int main ( void ){
    CInt a ( 65 ), b ( 12 ), c = a;
    a = 92;
    c . print ();
    return 0;
}
```

Program půjde zkompilovat, ale může spadnout/spadne.

---

**Otázka 11:** Co zobrazí následující program?

```
class CInt
{
    public:
        CInt ( int x ) : m_X ( new int ( x ) ) {}
        void operator = ( int x ) { *m_X = x; }
        void print ( void ) const { cout << *m_X; }
    private:
        int * m_X;
};

int main ( void ){
    CInt a ( 37 ), b ( 24 ), c = a;
    a = 5;
    c . print ();
    return 0;
}
```

Zobrazí se 5.

---

**Otázka 12:** Co zobrazí následující program?

```
class CInt{
    public:
        CInt ( int x ) : m_X ( new int ( x ) ) {}
        CInt ( const CInt & src ) { m_X = src . m_X; }
        void operator = ( int x ) { *m_X = x; }
        void print ( void ) const { cout << *m_X; }
    private:
        int * m_X;
};

int main ( void ){
    CInt a ( 83 ), b ( 12 ), c = a;
    a = 95;
    c . print ();
    return 0;
}
```

Zobrazí se 95.

---

**Otázka 13:** Co zobrazí následující kód?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum operator + ( CNum & a, CNum & b ) { return a + b; }
    void print ( void ) const { cout << m_X; }
private:
    int m_X;
};
int main ( void ){
    CNum a (17), b(30);
    CNum c = a + b;
    c . print ();
    return 0;
}
```

**Program nepůjde zkompilevat.**

---

**Otázka 14:** Co zobrazí následující kód?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    friend CNum operator + ( CNum & a, CNum & b ) { return a + b; }
    void print ( void ) const { cout << m_X; }
private:
    int m_X;
};
int main ( void ){
    CNum a (94);
    CNum c = a + 78;
    c . print ();
    return 0;
}
```

**Program nepůjde zkompilevat.**

---



**Otázka 15:** Co zobrazí následující kód?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum operator + ( const CNum & a ) { m_X += a . m_X; return *this; }
    void print ( void ) const { cout << m_X; }
private:
    int m_X;
};
int main ( void ){
    CNum a (98);
    CNum c = a + 43;
    c . print ();
    return 0;
}
```

Program funguje správně, ale přetěžuje operátory neobvyklým způsobem, bude pro ostatní programátory nepochopitelný.

---

**Otázka 16:** Co zobrazí následující kód?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum operator + ( const CNum & a ) const {
        CNum res ( m_X + a . m_X ); return CNum ( res );
    }
    void print ( void ) const { cout << m_X; }
private:
    int m_X;
};
int main ( void ){
    CNum a (39);
    CNum c = a + 16;
    c . print ();
    return 0;
}
```

Zobrazí se 55.

---

**Otázka 17:** Co zobrazí následující kód?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum operator + ( const CNum & a ) const { return a + m_X; }
    void print ( void ) const { cout << m_X; }
private:
    int m_X;
};
int main ( void ){
    CNum a (64);
    CNum c = a + 27;
    c . print ();
    return 0;
}
```

Program půjde zkompilevat, ale může spadnout/spadne.

---

**Otázka 18:** Co zobrazí následující program?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum ( const CNum & src ) { m_X = src . m_X; cout << m_X; }
    void debug ( void ) const { cout << *this; }
    friend ostream & operator << ( ostream & os, const CNum & x );
private:
    int m_X;
};
ostream & operator << ( ostream & os, const CNum & x ){
    os << x . m_X;
    return os;
}
...
CNum a ( 21 ), b = a;
b . debug();
```

Zobrazí se 2121.

---

**Otázka 19:** Co zobrazí následující program?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum ( const CNum & src ) { m_X = src . m_X; cout << m_X; }
    friend ostream & operator << ( ostream & os, const CNum x );
private:
    int m_X;
};
ostream & operator << ( ostream & os, const CNum x ){
    os << x . m_X;
    return os;
}
...
CNum a ( 88 ), b = a;
cout << a;
```

**Zobrazí se 888888.**

---

**Otázka 20:** Co zobrazí následující program?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum ( const CNum & src ) { m_X = src . m_X; cout << m_X; }
    friend ostream & operator << ( ostream & os, const CNum & x );
private:
    int m_X;
};
ostream & operator << ( ostream & os, const CNum & x ){
    os << x . m_X;
    return os;
}
...
CNum a ( 70 ), b = a;
cout << a;
```

**Zobrazí se 7070.**

---

**Otázka 21:** Co zobrazí následující program?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum ( const CNum & src ) { cout << m_X; m_X = src . m_X; }
    friend ostream & operator << ( ostream & os, const CNum & x );
private:
    int m_X;
};
ostream & operator << ( ostream & os, const CNum & x ){
    os << x . m_X;
    return os;
}
...
CNum a ( 55 ), b = a;
cout << a;
```

**Program nespadne, ale výstup je nedefinovaný.**

---

**Otázka 22:** Co zobrazí následující program?

```
class CNum{
public:
    CNum ( int x ) { m_X = x; }
    CNum ( const CNum & src ) { m_X = src . m_X; cout << m_X; }
    void debug ( void ) const { cout << *this; }
    friend ostream & operator << ( ostream & os, CNum & x );
private:
    int m_X;
};
ostream & operator << ( ostream & os, CNum & x ){
    os << x . m_X;
    return os;
}
...
CNum a ( 69 ), b = a;
b . debug();
```

**Program nepůjde zkompilovat.**

---

**Otázka 23:** Co zobrazí následující kód?

```
class CArray{
public:
    CArray ( int len ){
        m_D = new int[ m_L = len ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = 0;
    }
    CArray & operator = ( const CArray & src ){
        delete [] m_D;
        m_D = new int[ m_L = src . m_L ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = src . m_D[i];
        return *this;
    }
    int & operator [] ( int i ) { return m_D[i]; }
private:
    int * m_D;
    int m_L;
};
int main ( void ){
    CArray a(11), b = a;
    a[2] = 95;
    cout << b[2];
    return 0;
}
```

Program nespadne, ale neuvolňuje alokovanou paměť.

---

**Otázka 24:** Co zobrazí následující kód?

```
class CArray{
public:
    CArray ( int len ){
        m_D = new int[ m_L = len ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = 0;
    }
    CArray & operator = ( const CArray & src ){
        delete [] m_D;
        m_D = new int[ m_L = src . m_L ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = src . m_D[i];
        return *this;
    }
    int & operator [] ( int i ) { return m_D[i]; }
private:
    int * m_D;
    int m_L;
};
int main ( void ){
    CArray a(23), b = a;
    a[14] = 6;
    cout << b[14];
    return 0;
}
```

Program nespadne, ale neuvolňuje alokovanou paměť.

---

**Otázka 25:** Co zobrazí následující kód?

```
class CArray{
public:
    CArray( int len ){
        m_D = new int[ m_L = len ];
        for( int i = 0; i < m_L; i++ ) m_D[i] = 0;
    }
    CArray & operator = ( const CArray & src ){
        delete [] m_D;
        m_D = new int[ m_L = src . m_L ];
        for( int i = 0; i < m_L; i ++ ) m_D[i] = src . m_D[i];
        return *this;
    }
    ~CArray(void){ delete [] m_D; }
    int & operator [] ( int i ){ return m_D[i]; }
private:
    int * m_D;
    int m_L;
};

int main ( void ){
    CArray a(69), b = a;
    a[19] = 1;
    cout << b[19];
    return 0;
}
```

Program půjde zkompilovat, ale může spadnout/spadne.

---

**Otázka 26:** Co zobrazí následující kód?

```
class CArray{
public:
    CArray ( int len ){
        m_D = new int[ m_L = len ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = 0;
    }
    CArray & operator = ( const CArray & src ){
        delete [] m_D;
        m_D = new int[ m_L = src . m_L ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = src . m_D[i];
        return *this;
    }
    ~CArray ( void ) { delete [] m_D; }
    int & operator [] ( int i ) { return m_D[i]; }
private:
    int * m_D;
    int m_L;
};

int main ( void ){
    CArray a(96), b(89);
    b = a;
    a[89] = 72;
    cout << b[89];
    return 0;
}
```

**Program nespadne a zobrazí výsledek 0.**

---



**Otázka 27:** Co zobrazí následující kód?

```
class CArray{
public:
    CArray ( int len ){
        m_D = new int[ m_L = len ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = 0;
    }
    CArray & operator = ( const CArray & src ){
        delete [] m_D;
        m_D = new int[ m_L = src . m_L ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = src . m_D[i];
        return *this;
    }
    ~CArray ( void ) { delete [] m_D; }
    int & operator [] ( int i ) { return m_D[i]; }
private:
    int * m_D;
    int m_L;
};

int main ( void ){
    CArray a(89), b(87);
    a[87] = 64;
    b = a;
    cout << b[87];
    return 0;
}
```

**Program nespadne a zobrazí výsledek 64.**

---

**Otázka 28:** Co zobrazí následující kód?

```
class CArray{
public:
    CArray ( int len ){
        m_D = new int[ m_L = len ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = 0;
    }
    CArray & operator = ( const CArray src ){
        delete [] m_D;
        m_D = new int[ m_L = src . m_L ];
        for ( int i = 0; i < m_L; i ++ ) m_D[i] = src . m_D[i];
        return *this;
    }
    ~CArray ( void ) { delete [] m_D; }
    int & operator [] ( int i ) { return m_D[i]; }
private:
    int * m_D;
    int m_L;
};

int main ( void ){
    CArray a(64), b(57);
    b = a;
    a[57] = 42;
    cout << b[57];
    return 0;
}
```

Program půjde zkompilovat, ale může spadnout/spadne.

---

zdroj: 94.229.33.246/~kempa/progtest\_2test.tex