Sobrecarga: operador producto (*)

```
//main.cpp
...
int main()
{
   Fraction f(1,2), g(3,4), h;
   h = f * g;
   f*=g*=h;
}
```

```
//main.cpp
...
int main()
{
   Fraction f(1,2), g(3, h;
   h = f * g;
   f*=g*=h;
}
```

```
//main.cpp
...
int main()
{
    Fraction f(1,2), g(3,  h;
    h = f * g;
    f*=g*=h;
}

// fraction.cpp
Fraction::Fraction() : m_numerator(0), m_denominator(1)
{
}
```

h

m_numerator: ?
m_denominator: ?

h

m_numerator: 0
m_denominator: ?

```
//main.cpp
...
int main()
{
    Fraction f(1,2), g(3, h;
    h = f * g;
    f*=g*=h;
}

// fraction.cpp
Fraction::Fraction() : m_numerator m_denominator(1)

m_numerator m_denominator(1)

m_numerator m_denominator(1)
```

m_numerator: 0 m denominator: 1

```
//main.cpp
...
int main()
{
    Fraction f(1,2), g(3,  h;
    h = f * g;
    f*=g*=h;
}

// fraction.cpp
Fraction::Fraction() : m_numerator(0), m_denominator(1)
{
}
```

h

m_numerator: 0 m denominator: 1

```
//main.cpp
...
int main()
{
    Fraction f(1, g(3,4), h;
    h = f * g;
    f*=g*=h;
}
```

h

m_numerator: 0 m_denominator: 1

```
//main.cpp
int main()
  Fraction f(1, \square)g(3,4), h;
 h = f * q;
                                                               m numerator: ?
  f*=q*=h;
                                                              m denominator:?
                                                              h
// fraction.cpp
Fraction::Fraction(const int num, const int den) :
                                                               m numerator: 0
    m numerator(num), m denominator(den)
                                                              m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
int main()
                                                              g
  Fraction f(1, \square)g(3,4), h;
 h = f * g;
                                                               m numerator: 3
  f*=q*=h;
                                                              m denominator:?
                                                              h
// fraction.cpp
Fraction::Fraction(const int num, const int den) :
                                                               m numerator: 0
    m_numerator(num), m_denominator(den)
                                                              m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
int main()
                                                              g
  Fraction f(1, \square)g(3,4), h;
 h = f * g;
                                                                m numerator: 3
  f*=q*=h;
                                                               m denominator: 4
                                                              h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                                m numerator: 0
    m_numerator(nu ____) m_denominator(den)
                                                               m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
int main()
  Fraction f(1, \square)g(3,4), h;
 h = f * q;
                                                               m numerator: 3
  f*=q*=h;
                                                              m denominator: 4
                                                              h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                               m numerator: 0
    m_numerator(num), m_denominator(den)
                                                              m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
int main()
  Fraction f(1, \square)g(3,4), h;
 h = f * g;
                                                               m numerator: 3
  f*=q*=h;
                                                              m denominator: 4
                                                              h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                               m numerator: 0
    m_numerator(num), m_denominator(den)
                                                              m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
int main()
  Fraction f(1, \square)g(3,4), h;
 h = f * g;
                                                               m numerator: 3
  f*=q*=h;
                                                              m denominator: 4
                                                              h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                               m numerator: 0
    m_numerator(num), m_denominator(den)
                                                              m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
                                                             m numerator: ?
                                                            m denominator:?
int main()
                                                            g
 Fract f(1,2), g(3,4), h;
 h = f * q;
                                                             m numerator: 3
 f*=q*=h;
                                                            m denominator: 4
                                                            h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                             m numerator: 0
    m_numerator(num), m_denominator(den)
                                                            m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
                                                             m numerator: 1
                                                            m denominator:?
int main()
                                                            g
  Fract f(1,2), g(3,4), h;
 h = f * q;
                                                             m numerator: 3
  f*=q*=h;
                                                            m denominator: 4
                                                            h
// fraction.cpp
Fraction::Fraction(const int num, const int den) :
                                                             m numerator: 0
    m_numerator(num), m_denominator(den)
                                                            m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
                                                              m numerator: 1
                                                             m denominator: 2
int main()
                                                            g
 Fract f(1,2), g(3,4), h;
 h = f * q;
                                                              m numerator: 3
 f*=q*=h;
                                                             m denominator: 4
                                                            h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                              m numerator: 0
    m_numerator(nu ____) m_denominator(den)
                                                             m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
                                                             m numerator: 1
                                                            m denominator: 2
int main()
  Fract f(1,2), g(3,4), h;
 h = f * q;
                                                             m numerator: 3
  f*=q*=h;
                                                            m denominator: 4
                                                            h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                             m numerator: 0
    m_numerator(num), m_denominator(den)
                                                            m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

```
//main.cpp
                                                             m numerator: 1
                                                            m denominator: 2
int main()
  Fract f(1,2), g(3,4), h;
 h = f * q;
                                                             m numerator: 3
  f*=q*=h;
                                                            m denominator: 4
                                                            h
// fraction.cpp
Fraction::Fraction(const int num, const int den):
                                                             m numerator: 0
    m_numerator(num), m_denominator(den)
                                                            m denominator: 1
    if(m_denominator == 0)
        cout << "error: 0 passed in as denominator" << endl;</pre>
        exit(1);
```

m_numerator: 1 m_denominator: 2

g

m_numerator: 3 m_denominator: 4

h

m_numerator: 0 m_denominator: 1

```
f en main
                                     g en main
                Ihs en operator*
                                     rhs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                      m numerator: 3
                                                            m numerator: 0
                m denominator: 2
                                     m denominator: 4
                                                           m_denominator: 1
int main()
  Fraction f(1,2), g(3,4), h;
 f * g;
  f*=q*=h;
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    return (result*=rhs);
```

```
f en main
                                     g en main
                Ihs en operator*
                                     rhs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                      m numerator: 3
                                                            m numerator: 0
                m denominator: 2
                                     m_denominator: 4
                                                           m_denominator: 1
int main()
  Fraction f(1,2), g(3,4), h;
 f * g;
  f*=q*=h;
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    return (result*=rhs);
```

```
f en main
                                     g en main
               Ihs en operator*
                                     rhs en operator*
                                                          h en main
//main.cpp
                 m numerator: 1
                                      m numerator: 3
                                                            m numerator: 0
                m denominator: 2
                                     m denominator: 4
                                                           m denominator: 1
int main()
                                                          result en operator*
  Fraction f(1,2), g(3,4), h;
 f * g;
                                                            m numerator: ?
  f*=q*=h;
                                                           m denominator:?
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    return (result*=rhs);
Fraction::Fraction(const Fraction & source) :
    m_numerator(source.m_numerator), m_denominator(source.m_denominator) {}
```

```
f en main
                                     g en main
               Ihs en operator*
                                     rhs en operator*
                                                          h en main
//main.cpp
                 m numerator: 1
                                      m numerator: 3
                                                            m numerator: 0
                m denominator: 2
                                     m denominator: 4
                                                          m denominator: 1
int main()
                                                          result en operator*
  Fraction f(1,2), g(3,4), h;
 f * g;
                                                            m numerator: 1
  f*=q*=h;
                                                           m denominator:?
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    return (result*=rhs);
Fraction::Fraction(const Fraction & source):
    m_numerator(source.m_numerator), m_denominator(source.m_denominator) {}
```

```
f en main
                                     g en main
                Ihs en operator*
                                     rhs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                      m numerator: 3
                                                            m numerator: 0
                m denominator: 2
                                     m denominator: 4
                                                           m denominator: 1
int main()
                                                           result en operator*
  Fraction f(1,2), g(3,4), h;
 f * g;
                                                            m numerator: 1
  f*=q*=h;
                                                           m denominator: 2
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    return (result*=rhs);
Fraction::Fraction(const Fraction & source) :
    m_numerator(source.m_numerato___) m_denominator(source.m_denominator) {}
```

```
f en main
                                     g en main
                Ihs en operator*
                                      rhs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                       m numerator: 3
                                                            m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                           m denominator: 1
int main()
                                                           result en operator*
  Fraction f(1,2), g(3,4), h;
 f * g;
                                                            m numerator: 1
  f*=q*=h;
                                                           m denominator: 2
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    retu (result*=rhs);
```

```
g en main
                                      rhs en operator*
                f en main
                                      rhs en operator*=
                Ihs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                       m numerator: 3
                                                             m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 1
int main()
                                                           result en operator*
                                                           *this en operator*=
  Fraction f(1,2), g(3,4), h;
 f * q;
                                                             m numerator: 1
  f*=q*=h;
                                                            m denominator: 2
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    retu (result*=rhs);
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m denominator*=rhs.m denominator;
    return (*this);
```

```
g en main
                                      rhs en operator*
                f en main
                                      rhs en operator*=
                Ihs en operator*
                                                           h en main
//main.cpp
                 m_numerator: 1
                                       m numerator: 3
                                                             m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 1
int main()
                                                           result en operator*
                                                           *this en operator*=
  Fraction f(1,2), g(3,4), h;
 f * g;
                                                             m numerator: 1
  f*=q*=h;
                                                            m denominator: 2
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    retu (result*=rhs);
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m denominator*=rhs.m denominator;
    return (*this);
```

```
g en main
                                      rhs en operator*
                f en main
                                      rhs en operator*=
                Ihs en operator*
                                                           h en main
//main.cpp
                 m_numerator: 1
                                       m numerator: 3
                                                             m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 1
int main()
                                                           result en operator*
                                                           *this en operator*=
  Fraction f(1,2), g(3,4), h;
 f * g;
                                                             m numerator: 3
  f*=q*=h;
                                                            m denominator: 2
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    retu (result*=rhs);
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m denominator*=rhs.m denominator;
    return (*this);
```

```
g en main
                                      rhs en operator*
                f en main
                                      rhs en operator*=
                Ihs en operator*
                                                           h en main
//main.cpp
                 m_numerator: 1
                                       m numerator: 3
                                                             m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 1
int main()
                                                            result en operator*
                                                            *this en operator*=
  Fraction f(1,2), g(3,4), h;
  f * q;
                                                             m numerator: 3
  f*=q*=h;
                                                            m denominator: 8
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    retu (result*=rhs);
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m denominator*=rhs.m denominator;
    return (*this);
```

```
g en main
                                      rhs en operator*
                f en main
                                      rhs en operator*=
                Ihs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                       m numerator: 3
                                                             m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 1
int main()
                                                            result en operator*
                                                            *this en operator*=
  Fraction f(1,2), g(3,4), h;
  f * q;
                                                             m numerator: 3
  f*=q*=h;
                                                            m denominator: 8
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    retu (result*=rhs);
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
f en main
                                     g en main
                Ihs en operator*
                                     rhs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                       m numerator: 3
                                                            m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                           m denominator: 1
int main()
                                                           result en operator*
  Fraction f(1,2), g(3,4), h;
 f * g;
                                                            m numerator: 3
  f*=q*=h;
                                                           m denominator: 8
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    retu (result*=rhs);
```

```
f en main
                                     g en main
                Ihs en operator*
                                      rhs en operator*
                                                           h en main
//main.cpp
                 m numerator: 1
                                       m numerator: 3
                                                             m numerator: 0
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 1
int main()
                                                           result en operator*
  Fraction f(1,2), g(3,4), h;
  f * q;
                                                             m numerator: 3
  f*=q*=h;
                                                            m denominator: 8
Fraction operator* (const Fraction & lhs, const Fraction & rhs)
    Fraction result(lhs);
    return (result*=rhs);
```

f en main

m_numerator: 1 m_denominator: 2

g en main

m_numerator: 3 m_denominator: 4

h en main

m_numerator: 0 m_denominator: 1

result from operator*

m_numerator: 3 m_denominator: 8

//main.cpp

int main()

f en main

g en main

m_numerator: 3 m_denominator: 4

h en main

m_numerator: 3 m_denominator: 8

result from operator*

m_numerator: 3 m_denominator: 8

f en main

g en main

m_numerator: 3m_denominator: 4

h en main

m_numerator: 3m_denominator: 8

```
g en main
                                                           h en main
                                      *this en operator*=
                f en main
                                                           rhs en operator*=
//main.cpp
                 m numerator: 1
                                       m numerator: 3
                                                             m numerator: 3
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
  h = f * q;
    a*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
g en main
                                                           h en main
                                      *this en operator*=
                f en main
                                                           rhs en operator*=
//main.cpp
                 m numerator: 1
                                       m numerator: 3
                                                             m numerator: 3
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
  h = f * q;
    a*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
g en main
                                                           h en main
                                      *this en operator*=
                f en main
                                                           rhs en operator*=
//main.cpp
                 m numerator: 1
                                       m numerator: 9
                                                             m numerator: 3
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
  h = f * q;
    a*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
g en main
                                                           h en main
                                      *this en operator*=
                f en main
                                                           rhs en operator*=
//main.cpp
                 m numerator: 1
                                       m numerator: 9
                                                             m numerator: 3
                m denominator: 2
                                      m denominator: 4
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
  h = f * q;
    q*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
g en main
                                                           h en main
                                      *this en operator*=
                f en main
                                                            rhs en operator*=
//main.cpp
                 m numerator: 1
                                                             m numerator: 3
                                       m numerator: 9
                                      m_denominator: 32
                m denominator: 2
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
 h = f * q;
    a*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
g en main
                                                           h en main
                                      *this en operator*=
                f en main
                                                            rhs en operator*=
//main.cpp
                 m numerator: 1
                                                             m numerator: 3
                                       m numerator: 9
                                      m_denominator: 32
                m denominator: 2
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
 h = f * q;
    a*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

f en main

g en main

m_numerator: 9 m_denominator: 32

h en main

m_numerator: 3m_denominator: 8

fen main //main.cpp m nu

Fraction f(1,2), g(3,4), h;

int main()

h = f * g;

f*=g*=h;

m_numerator: 1 m_denominator: 2

g en main

m_numerator: 9m_denominator: 32

h en main

m_numerator: 3m_denominator: 8

```
f en main
                                      g en main
                *this en operator*=
                                      rhs en operator*=
                                                            h en main
//main.cpp
                 m numerator: 1
                                                             m numerator: 3
                                       m numerator: 9
                m denominator: 2
                                       m denominator: 32
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
  h = f * q;
  f*=q*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
f en main
                                      g en main
                *this en operator*=
                                      rhs en operator*=
                                                            h en main
//main.cpp
                 m numerator: 1
                                                             m numerator: 3
                                       m numerator: 9
                                       m_denominator: 32
                m denominator: 2
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
  h = f * q;
  f*=q*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
f en main
                                      g en main
                *this en operator*=
                                      rhs en operator*=
                                                            h en main
//main.cpp
                 m numerator: 9
                                                             m numerator: 3
                                       m numerator: 9
                                       m_denominator: 32
                m denominator: 2
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
  h = f * q;
  f*=q*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
f en main
                                      g en main
                *this en operator*=
                                      rhs en operator*=
                                                            h en main
//main.cpp
                                                             m numerator: 3
                 m numerator: 9
                                       m numerator: 9
                 m denominator: 64
                                       m denominator: 32
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
 h = f * q;
  f*=q*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

```
f en main
                                      g en main
                *this en operator*=
                                      rhs en operator*=
                                                            h en main
//main.cpp
                                                             m numerator: 3
                 m numerator: 9
                                       m numerator: 9
                 m denominator: 64
                                       m denominator: 32
                                                            m denominator: 8
int main()
  Fraction f(1,2), g(3,4), h;
 h = f * q;
  f*=q*=h;
Fraction& Fraction::operator*= (const Fraction & rhs)
    m_numerator*=rhs.m_numerator;
    m_denominator*=rhs.m_denominator;
    return (*this);
```

f en main

Fraction f(1,2), g(3,4), h;

//main.cpp

int main()

h = f * g;

f*=g*=h;

m_numerator: 9 m_denominator: 64

g en main

m_numerator: 9 m_denominator: 32

h en main

m_numerator: 3m_denominator: 8