

```

/* binom_modul.h */

// berechnet Wert aus dem Pascal'schen Dreieck
// ( Zeilenindex, Spaltenindex)
unsigned int pascal(unsigned int, unsigned int);

// gibt das Pascal'sche Dreieck auf Konsole aus
// ( Tiefe=4, Abstand=4)
void show_pascal(unsigned int tiefe=4, unsigned int abstand=4);

// berechnet eine Binomial-Potenz (a+b)^exp
// ( ErsterWert, ZweiterWert, Exponent)
long double pot_binom(float, float, unsigned int);

// zeigt die Berechnungsvorschrift fuer gegebene
// Binomial-Potenz an
// (ErsterWert, ZweiterWert, Exponent)
void show_binom_calc(float, float, unsigned int);

```

```

/* binom_modul.cpp */

#include "binom_modul.h"
#include <iostream>
#include <iomanip>
using namespace std;

// Prototyp -> statische Funktionen werden
// nicht nach aussen bekannt gemacht, also auch nicht deren
// Prototypen (Funktionsdeklarationen)
static long double power(float, int);

unsigned int pascal(unsigned int zeile, unsigned int spalte)
{
    if (spalte==0 || spalte==zeile)
        return 1;
    // nicht zulaessig, da negative Fakultaet nicht existiert
    if (spalte > zeile)
        return 0;
    return pascal(zeile-1, spalte-1)+
           pascal(zeile-1, spalte);
}

void show_pascal(unsigned int tiefe, unsigned int abstand)
{
    for(int i=0; i<tiefe; i++)
    {
        for(int leer=tiefe; leer>i; leer--)
            cout << setw(abstand)<< " ";
        for(int j=0; j<=i; j++)
        {
            cout << setw(2*abstand) << pascal(i,j);
        }
        cout << endl;
    }
}

```

```

long double pot_binom(float a, float b, unsigned int exp)
{
    long double ergebnis = 0;
    for(int i=0; i <= exp; i++)
    {
        ergebnis += pascal(exp,i) *
                    power(a,exp-i) *
                    power(b,i);
    }
    return ergebnis;
}

void show_binom_calc(float a, float b, unsigned int exp)
{
    cout << "\n( " << a << " + " << b << " ) ^ " << exp << " = ";
    for(int i=0; i <= exp; i++)
    {
        if (i > 0)
            cout << " + ";
        cout << pascal(exp,i) << "*"
            << "a^" << exp-i << "*"
            << "b^" << i;
    }
    cout << endl;
}

static long double power(float x, int y)
{
    if(x == 0)
        return 0;
    if(x == 1 || y == 0)
        return 1;
    if(y < 0)
        return 1/power(x, -y);
    if(y == 1)
        return x;
    return x * power(x, y-1);
}

```

```

/* binom.cpp */

#include "binom_modul.h"
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    unsigned int tiefe;
    long double ergebnis;
    float a,b;
    unsigned int n,k;
    unsigned int exp;

    cout << "Pascal'sche Binomial-Potenzen" << endl;
    cout << "=====\n" << endl;
    cout << "Berechnung von (a+b)^exp:\n" << endl;
    cout << "> Wert von a: ";
    cin >> a;
    cout << "> Wert von b: ";
    cin >> b;
    cout << "> Wert von exp: ";
    cin >> exp;

    show_pascal(exp+1);
    show_binom_calc(a,b,exp);
    cout << "\n( " << a << " + " << b << " ) ^ " << exp << " = "
        << fixed << pot_binom(a,b,exp) << endl << endl << endl;

    cout << "Pascal'sche Binomial-Koeffizienten" << endl;
    cout << "=====\n" << endl;
    cout << "Berechnung der Kombinationen von n "
        << "Elementen der k-ten Klasse:\n" << endl;
    cout << "> Wert von n: ";
    cin >> n;
    cout << "> Wert von k: ";
    cin >> k;
    cout << endl;
    cout << "( " << setw(3) << n << " ) " << endl
        << "( " << setw(3) << " " << " ) = " << pascal(n,k) << endl
        << "( " << setw(3) << k << " ) " << endl;
    return 0;
}

```

## Programmerstellung

```

g++ binom.cpp binom_modul.cpp -o binom
./binom

```