Erweiterung Koordinaten

/\*T. Hofmann

Koordinaten Erweiterung\*/

#include <iostream>

#include <string>

using namespace std;

struct Point

{

    int x;

    int y;

    Point()

    {

        x = 0;

        y = 0;

    }

    Point(int x, int y)

    {

        this->x = x;

        this->y = y;

    }

    string stringPoint()

    {

        string res = "(" + to\_string(x) + "|" + to\_string(y) + ")";

        return res;

    }

};

Point operator+(Point a, Point b)

{

    Point res;

    res.x = a.x + b.x;

    res.y = a.y + b.y;

    return res;

}

Point operator-(Point a, Point b)

{

    Point res;

    res.x = a.x - b.x;

    res.y = a.y - b.y;

    return res;

}

Point operator\*(Point a, Point b)

{

    Point res;

    res.x = a.x \* b.x;

    res.y = a.y \* b.y;

    return res;

}

Point operator/(Point a, Point b)

{

    Point res;

    res.x = a.x / b.x;

    res.y = a.y / b.y;

    return res;

}

ostream& operator<<(ostream& os, Point& a)

{

    os << "(" << a.x << "|" << a.y << ")";

    return os;

}

istream& operator>>(istream& is, Point& a)

{

    cout << "x: ";

    is >> a.x;

    cout << "y: ";

    is >> a.y;

    return is;

}

int main()

{

    bool repeat = true;

    int action = 0;

    Point a = Point();

    Point b = Point();

    int x = 0;

    int y = 0;

    cin >> a;

    cout << endl;

    cout << a << endl;

    while (repeat)

    {

        cout << endl;

        cout << "Please choose your next action:" << endl;

        cout << "1) Add point" << endl;

        cout << "2) Subtract point" << endl;

        cout << "3) Multiply point" << endl;

        cout << "4) Divide point" << endl;

        cout << "5) End the program" << endl;

        cout << "Input: ";

        cin >> action;

        while (action != 1 && action != 2 && action != 3 && action != 4 && action != 5)

        {

            cout << endl;

            cout << "Invalid action! Please try again." << endl;

            cout << "1) Add point" << endl;

            cout << "2) Subtract point" << endl;

            cout << "3) Multiply point" << endl;

            cout << "4) Divide point" << endl;

            cout << "5) End the program" << endl;

            cout << "Input: ";

            cin >> action;

        }

        switch (action)

        {

        case 1:

            cout << endl;

            cout << "Create point B that you want to add to point A:" << endl;

            cin >> b;

            cout << endl;

            cout << a.stringPoint() << " + " << b.stringPoint() << " = ";

            a = a + b;

            cout << a << endl;

            cout << endl;

            break;

        case 2:

            cout << endl;

            cout << "Create point B that you want to subtract from point A:" << endl;

            cin >> b;

            cout << endl;

            cout << a.stringPoint() << " - " << b.stringPoint() << " = ";

            a = a - b;

            cout << a << endl;

            cout << endl;

            break;

        case 3:

            cout << endl;

            cout << "Create point B that you want to multiply with point A:" << endl;

            cin >> b;

            cout << endl;

            cout << a.stringPoint() << " \* " << b.stringPoint() << " = ";

            a = a \* b;

            cout << a << endl;

            cout << endl;

            break;

        case 4:

            cout << endl;

            cout << "Create point B that you want to divide with point A:" << endl;

            cin >> b;

            cout << endl;

            while (b.x == 0 || b.y == 0)

            {

                cout << "Point B can't have any 0 in it. Please redefine the point:" << endl;

                cin >> b;

                cout << endl;

            }

            cout << a.stringPoint() << " / " << b.stringPoint() << " = ";

            a = a / b;

            cout << a << endl;

            cout << endl;

            break;

        case 5:

            cout << endl;

            cout << "Result: ";

            cout << a << endl;

            repeat = false;

            break;

        default:

            break;

        }

    }

    system("pause");

}