Stephan Brumme, SST, 3.FS, Matrikelnr. 70 25 44
mail@stephan-brumme.com

Aufgabe M7.1

Ich habe *MCashOffice.** aus Aufgabe 6.1 wiederverwendet. Die Headerdatei *MRoom.h* wurde lediglich hinsichtlich des Datentyps *TRoom* verändert, ansonsten entspricht sie genau der Version aus Aufgabe 6.2.

In *MRoom.cpp* gibt jede Funktion nach einem kurzen Statustext sämtliche Parameter aus, die übergeben werden. Anschließend kann der Benutzer den Rückgabewert beeinflussen. Der nun zurückgegebene Wert wird noch kurz angezeigt, dann folgt auch schon ein abschließender Statustext. Den originalen Code des Moduls *MRoom* ließ ich als Kommentar stehen.

Die alte Struktur *TRoom* wurde durch einen Integerwert ersetzt, der eine Art Objektidentität repräsentiert.

```
typedef int TRoom;
```

Er wird bei allen Operationen ausgegebenen und ermöglicht die Unterscheidung der einzelnen Räume.

Sämtliche Dialogführung erfolgt über die Standard-Aus-/Eingabe und kann daher problemlos auf andere Kanäle, z.B. Dateien umgelenkt, wo dann eine dauerhafte Speicherung des Protokolls möglich ist.

```
// Returns the NumberOfRooms attribute
Ordinal MRoom::GetNumberOfRooms(TRoom roo)
{
	// display in-parameters
	cout<<"MRoom::GetNumberOfRooms entered"<<endl;
	cout<<"	IN:  Room: "<<roo<<endl;

	cout<<"------------------------------------------------------------"<<endl;
	cout<<"How many rooms ? ";
	int nReturn;
	cin>>nReturn;
	// display entered char
	cout<<nReturn<<endl;
	cout<<"------------------------------------------------------------"<<endl;

	// display out-parameters
	cout<<"		return "<<nReturn;
	cout<<"MRoom::GetNumberOfRooms left"<<endl<<endl;

	return nReturn;

//	replaced code:
//	return roo.NumberOfRooms;
```

MRoom.h:

```
////////////////////////////////////////////////////////
// Softwarebauelemente I, Aufgabe M7.1
//
// author:		Stephan Brumme
// last changes:	January 08, 2001

#ifndef __ROOM_H__
#define __ROOM_H__


#include "PrimitiveTypes.h"


// define the namespace Room
namespace MRoom
{
	// Data structure representing a room unit
	typedef int TRoom;

//	struct TRoom
//	{
//		Ordinal NumberOfRooms;
//		Ordinal Area;
```

```
//      };


        // Initializes the TRoom structure
        void Init(TRoom &roo, Ordinal nor, Ordinal ar);

        // Compares two exemplars
        // returns "true" if attributes of both are equal; "false" otherwise
        Boolean EqualValue(TRoom roo1, TRoom roo2);

        // Copies the attributes of roo2
        // returns "true" if successful, "false" if no memory allocated
        Boolean Copy(TRoom& roo1, TRoom roo2);

        // Returns the NumberOfRooms attribute
        Ordinal GetNumberOfRooms(TRoom roo);

        // Sets the NumberOfRooms attribute
        void SetNumberOfRooms(TRoom &roo, Ordinal nor);

        // Returns the Area attribute
        Ordinal GetArea(TRoom roo);

        // Displays the attributes
        void Show(TRoom roo);
}

#endif
```

MRoom.cpp:

```
////////////////////////////////////////////////////////
// Softwarebauelemente I, Aufgabe M7.1
//
// author:            Stephan Brumme
// last changes:      January 08, 2001


// import cout to display some data
#include <iostream>
#include "MRoom.h"

// open std namespace
using namespace std;



// define the namespace Room

// Initializes the TRoom structure
void MRoom::Init(TRoom &roo, Ordinal nor, Ordinal ar)
{
        // display in-parameters
        cout<<"MRoom::Init entered"<<endl;
        cout<<"       IN:  NumberOfRooms: "<<nor<<endl;
        cout<<"            Area:          "<<ar<<endl;

        // global counter
        static int nCount = 0;
        roo = ++nCount;

        // display out-parameters
        cout<<"       OUT: TRoom:         "<<roo<<endl;
        cout<<"MRoom::Init left"<<endl<<endl;

//      replaced code:
//      roo.NumberOfRooms = nor;
//      roo.Area = ar;
}


// Compares two exemplars
// returns "true" if attributes of both are equal; "false" otherwise
```

```
Boolean MRoom::EqualValue(TRoom roo1, TRoom roo2)
{
	// display in-parameters
	cout<<"MRoom::EqualValue entered"<<endl;
	cout<<"	IN:  Room1: "<<roo1<<endl;
	cout<<"	     Room2: "<<roo2<<endl;

	// ask user
	cout<<"------------------------------------------------------------"<<endl;
	cout<<"Are those rooms equal (Y=yes, else false) ? ";
	char cInput;
	cin>>cInput;
	// display entered char
	cout<<cInput<<endl;
	cout<<"------------------------------------------------------------"<<endl;

	// convert to Boolean
	Boolean bReturn = (cInput=='Y')||(cInput=='y');

	// display out-parameters
	cout<<"	OUT: return "<<bReturn;
	cout<<"MRoom::EqualValue left"<<endl<<endl;

	return bReturn;

//	replaced code:
//	return ((roo1.Area == roo2.Area) &&
//		(roo1.NumberOfRooms = roo2.NumberOfRooms));
}


// Copies the attributes of roo2
// returns "true" if successful, "false" if no memory allocated
Boolean MRoom::Copy(TRoom& roo1, TRoom roo2)
{
	// display in-parameters
	cout<<"MRoom::Copy entered"<<endl;
	cout<<"	IN:  Room1: "<<roo1<<endl;
	cout<<"	     Room2: "<<roo2<<endl;

	// ask user
	cout<<"------------------------------------------------------------"<<endl;
	cout<<"Should Room2 be copied to Room1 (Y=yes, else false) ? ";
	char cInput;
	cin>>cInput;
	// display entered char
	cout<<cInput<<endl;
	cout<<"------------------------------------------------------------"<<endl;

	// convert to Boolean
	Boolean bReturn = (cInput=='Y')||(cInput=='y');
	if (bReturn)
		roo1 = roo2;

	// display out-parameters
	cout<<"	OUT: Room1: "<<roo1<<endl;
	cout<<"	     return "<<bReturn;
	cout<<"MRoom::Copy left"<<endl<<endl;

	return bReturn;

//	replaced code:
//	if (EqualValue(roo1, roo2))
//		return false;
//
//	roo1.Area = roo2.Area;
//	roo1.NumberOfRooms = roo2.NumberOfRooms;
//	return true;
}


// Returns the NumberOfRooms attribute
Ordinal MRoom::GetNumberOfRooms(TRoom roo)
{
	// display in-parameters
```

```
        cout<<"MRoom::GetNumberOfRooms entered"<<endl;
        cout<<"       IN:  Room: "<<roo<<endl;

        cout<<"------------------------------------------------------------"<<endl;
        cout<<"How many rooms ? ";
        int nReturn;
        cin>>nReturn;
        // display entered char
        cout<<nReturn<<endl;
        cout<<"------------------------------------------------------------"<<endl;

        // display out-parameters
        cout<<"            return "<<nReturn;
        cout<<"MRoom::GetNumberOfRooms left"<<endl<<endl;

        return nReturn;

//      replaced code:
//      return roo.NumberOfRooms;
}


// Sets the NumberOfRooms attribute
void MRoom::SetNumberOfRooms(TRoom &roo, Ordinal nor)
{
        // display in-parameters
        cout<<"MRoom::SetNumberOfRooms entered - not affecting anything !"<<endl;
        cout<<"       IN:  Room:          "<<roo<<endl;
        cout<<"            NumberOfRooms: "<<nor<<endl;

        // display out-parameters
        cout<<"MRoom::SetNumberOfRooms left"<<endl<<endl;

//      replaced code:
//      roo.NumberOfRooms = nor;
}


// Returns the Area attribute
Ordinal MRoom::GetArea(TRoom roo)
{
        // display in-parameters
        cout<<"MRoom::GetArea entered"<<endl;
        cout<<"       IN:  Room: "<<roo<<endl;

        cout<<"------------------------------------------------------------"<<endl;
        cout<<"How many square feet ? ";
        int nReturn;
        cin>>nReturn;
        // display entered char
        cout<<nReturn<<endl;
        cout<<"------------------------------------------------------------"<<endl;

        // display out-parameters
        cout<<"            return "<<nReturn;

        // display out-parameters
        cout<<"MRoom::GetArea left"<<endl<<endl;

        return nReturn;

//      replaced code:
//      return roo.Area;
}


// Displays the attributes
void MRoom::Show(TRoom roo)
{
        // display in-parameters
        cout<<"MRoom::Show entered - not affecting anything !"<<endl;
        cout<<"       IN:  Room: "<<roo<<endl;

        // display out-parameters
        cout<<"MRoom::Show left"<<endl<<endl;
```

```
//      replaced code:
//      cout<<"Es sind "<<roo.NumberOfRooms<<" Raeume mit einer Flaeche von "<<roo.Area
//              <<"."<<endl;
}

// end of namespace Room
```