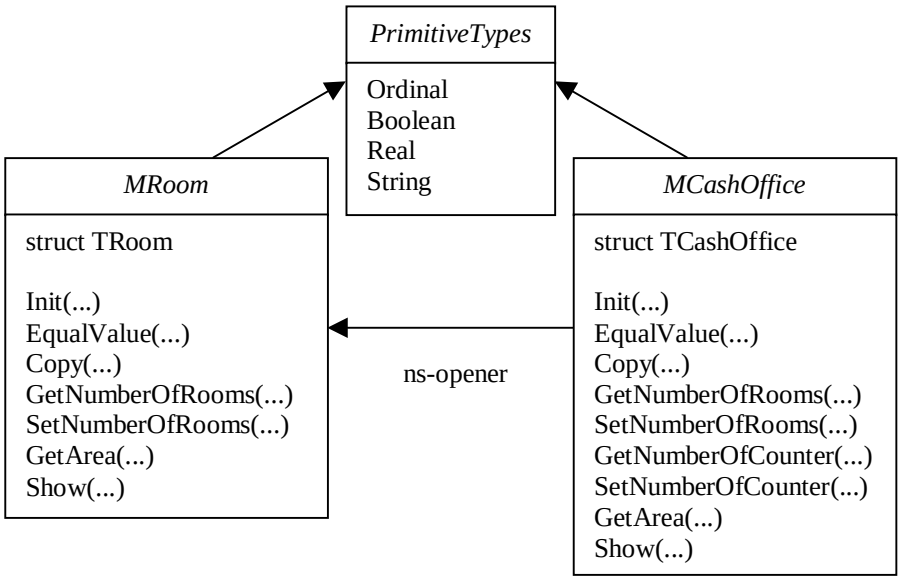
Aufgabe 6.1.

Die Namensraumstruktur der CEDL-Beschreibung lässt sich graphisch veranschaulichen:



Die Basis-Datentypen werden in *PrimitiveTypes.h* definiert. Ich verzichte auf die Verwendung eines eigenen Namensraumes, da ich in diesem Fall den erhöhten Aufwand für nicht gerechtfertigt halte.

PrimitiveTypes.h:

```
////////////////////////////////////////////////////////
// Softwarebauelemente I, Aufgabe M6.1
//
// author:            Stephan Brumme
// last changes:      November 23, 2000

#ifndef __PRIMITIVETYPES_H__
#define __PRIMITIVETYPES_H__

// define the basic types
// names are closely related to their Pascal equivalents

typedef int   Ordinal;
typedef bool  Boolean;
typedef float Real;
typedef char* String;

#endif
```

Der Namensraum *MRoom* baut auf den vorherigen Übungen auf. Die Parameter könnten zwar des öfteren als konstante Referenz übergeben werden, ich verzichte aber darauf, da die Datenstruktur nur unwesentlich größer als ein Zeiger ist.

MRoom.h:

```
////////////////////////////////////////////////////////
// Softwarebauelemente I, Aufgabe M6.1
//
// author:            Stephan Brumme
// last changes:      November 23, 2000

#ifndef __ROOM_H__
#define __ROOM_H__
```

```
#include "PrimitiveTypes.h"


// define the namespace Room
namespace MRoom
{
	// Data structure representing a room unit
	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 M6.1
//
// author:            Stephan Brumme
// last changes:      November 23, 2000


// 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)
{
	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)
{
```

```
        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)
{
        if ((roo1 == NULL) ||
                (EqualValue(*roo1, roo2)))
                return false;

        roo1->Area = roo2.Area;
        roo1->NumberOfRooms = roo2.NumberOfRooms;
        return true;
}


// Returns the NumberOfRooms attribute
Ordinal MRoom::GetNumberOfRooms(TRoom roo)
{
        return roo.NumberOfRooms;
}


// Sets the NumberOfRooms attribute
void MRoom::SetNumberOfRooms(TRoom &roo, Ordinal nor)
{
        roo.NumberOfRooms = nor;
}


// Returns the Area attribute
Ordinal MRoom::GetArea(TRoom roo)
{
        return roo.Area;
}


// Displays the attributes
void MRoom::Show(TRoom roo)
{
        cout<<"Es sind "<<roo.NumberOfRooms<<" Raeume mit einer Flaeche von "<<roo.Area
                <<"."<<endl;
}

// end of namespace Room
```

Der Namensraum *MCashOffice* unterscheidet sich nicht allzu stark von *MRoom*. Da ich den Quellcode recht ausführlich kommentiert habe, verzichte ich hier auf weitere Anmerkungen.

MCashOffice.h:

```
////////////////////////////////////////////////////////
// Softwarebauelemente I, Aufgabe M6.1
//
// author:            Stephan Brumme
// last changes:      November 23, 2000


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

namespace MCashOffice
{
        // Import the Room namespace
        using namespace MRoom;


        // Data structure representing a cash office unit
```

```
        struct TCashOffice
        {
                TRoom Room;
                Ordinal NumberOfCounter;
        };


        // Initializes the TCashOffice structure
        void Init(TCashOffice &cof, Ordinal nor, Ordinal ar, Ordinal noc);

        // Compares two exemplars
        // returns "true" if attributes of both are equal; "false" otherwise
        Boolean EqualValue(TCashOffice cof1, TCashOffice cof2);

        // Copies the attributes of roo2
        // returns "true" if successful, "false" if no memory allocated
        Boolean Copy(TCashOffice* cof1, TCashOffice cof2);

        // Returns the NumberOfRooms attribute
        Ordinal GetNumberOfRooms(TCashOffice cof);

        // Sets the NumberOfRooms attribute
        void SetNumberOfRooms(TCashOffice &cof, Ordinal nor);

        // Returns the Area attribute
        Ordinal GetArea(TCashOffice cof);

        // Returns the NumberOfCounter attribute
        Ordinal GetNumberOfCounter(TCashOffice cof);

        // Sets the NumberOfCounter attribute
        void SetNumberOfCounter(TCashOffice &cof, Ordinal nor);

        // Displays the attributes
        void Show(TCashOffice cof);

        // end of namespace CashOffice
}
```

MCashOffice.cpp:

```
////////////////////////////////////////////////////////
// Softwarebauelemente I, Aufgabe M6.1
//
// author:            Stephan Brumme
// last changes:      November 23, 2000


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


// open std namespace
using namespace std;


void MCashOffice::Init(TCashOffice &cof, Ordinal nor, Ordinal ar, Ordinal noc)
{
        cof.NumberOfCounter = noc;
        cof.Room.NumberOfRooms = nor;
        cof.Room.Area = ar;
}


// Compares two exemplars
// returns "true" if attributes of both are equal; "false" otherwise
Boolean MCashOffice::EqualValue(TCashOffice cof1, TCashOffice cof2)
{
        return ((cof1.NumberOfCounter == cof2.NumberOfCounter) &&
                        (MRoom::EqualValue(cof1.Room, cof2.Room)));
}
```

```
// Copies the attributes of roo2
// returns "true" if successful, "false" if no memory allocated
Boolean MCashOffice::Copy(TCashOffice* cof1, TCashOffice cof2)
{
	if ((cof1 == NULL) ||
		(EqualValue(*cof1, cof2)))
		return false;

	cof1->NumberOfCounter = cof2.NumberOfCounter;
	MRoom::Copy(&(cof1->Room), cof2.Room);
	return true;
}


// Returns the NumberOfRooms attribute
Ordinal MCashOffice::GetNumberOfRooms(TCashOffice cof)
{
	return cof.Room.NumberOfRooms;
}


// Sets the NumberOfRooms attribute
void MCashOffice::SetNumberOfRooms(TCashOffice &cof, Ordinal nor)
{
	cof.Room.NumberOfRooms = nor;
}


// Returns the Area attribute
Ordinal MCashOffice::GetArea(TCashOffice cof)
{
	return cof.Room.Area;
}


// Returns the NumberOfCounter attribute
Ordinal MCashOffice::GetNumberOfCounter(TCashOffice cof)
{
	return cof.NumberOfCounter;
}


// Sets the NumberOfCounter attribute
void MCashOffice::SetNumberOfCounter(TCashOffice &cof, Ordinal nor)
{
	cof.NumberOfCounter = nor;
}


// Displays the attributes
void MCashOffice::Show(TCashOffice cof)
{
	cout<<"Es sind "<<cof.Room.NumberOfRooms
		<<" Raeume mit einer Flaeche von "<<cof.Room.Area
		<<" an Kasse "<<cof.NumberOfCounter
		<<"."<<endl;
}

// end of namespace CashOffice
```

Zum Testen verwendete ich die Datei *M06_1.cpp*, die ich jetzt aus Platzgründen in einer Minimalfassung wiedergebe:

M06_1.cpp:

```
////////////////////////////////////////////////////////
// Softwarebauelemente I, Aufgabe M6.1
//
// author:            Stephan Brumme
// last changes:      November 23, 2000
```

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

// open std namespace
using namespace std;


void main()
{
	// ... use the namespaces in some way ...
}
```