

Demonstrator-Anwendung 2



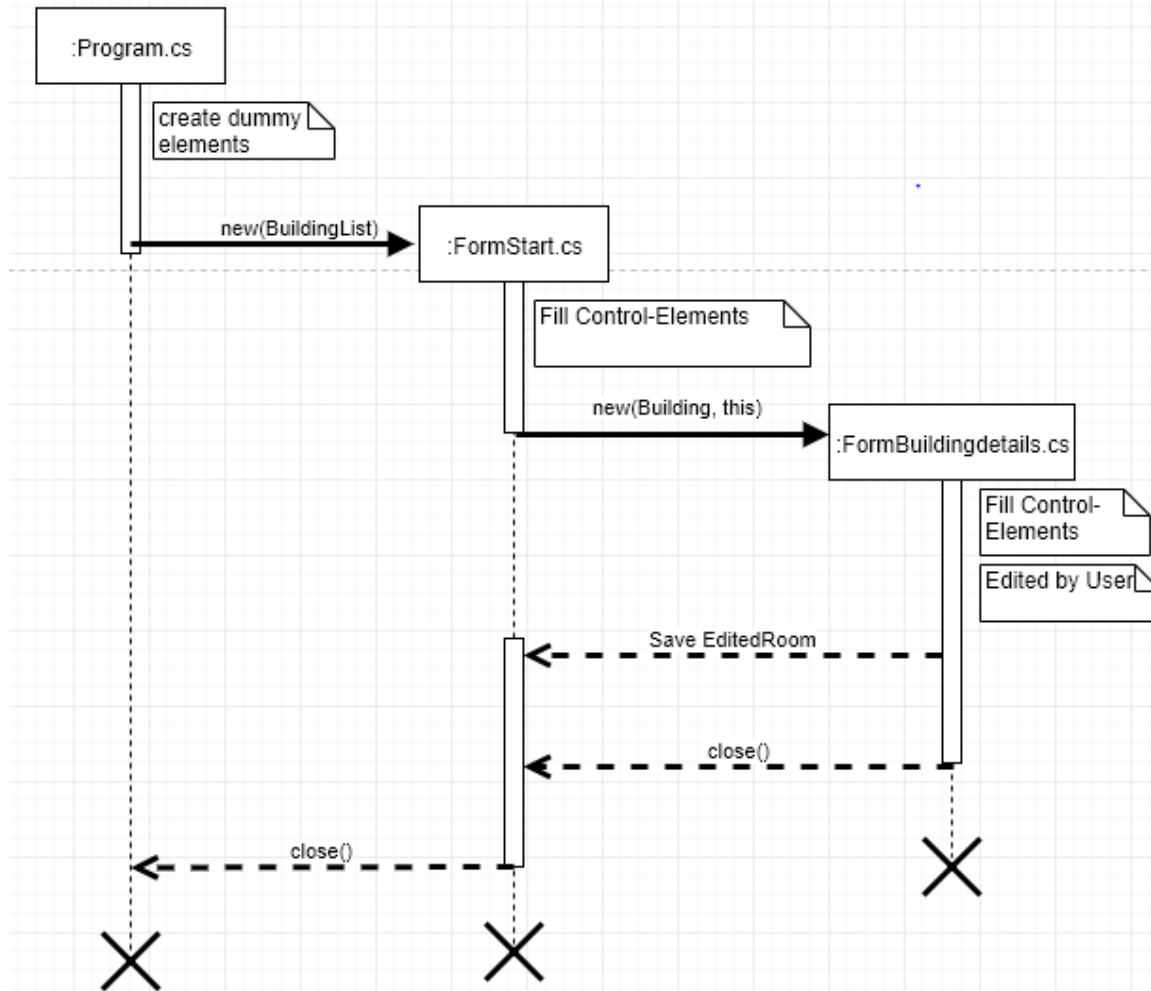
TECHNISCHE
UNIVERSITÄT
DARMSTADT



- Modularer Aufbau
 - Eigene Projekte für Klassen, GUI und Revit-Anbindung
 - höhere Flexibilität bei Änderungen
 - Klassenbibliothek definiert Objekte und deren Funktionen
 - WindowsForms-Anwendung zur Visualisierung der Inhalte
 - Revit AddIn zum Auslesen der Informationen aus Revit und deren Weitergabe an das Form
- ➔ Keine Logik (Berechnung des Verhältnisses zwischen Fenster- und Raumfläche) außerhalb der Klassenbibliothek

- WindowsForms-Anwendung
- 2 Forms
 - Form „FormStart“ – Ausgangspunkt der Anwendung
 - Übergabeparameter: Liste aller zu verwaltenden Gebäude
 - gibt einen Überblick über vorhandene Daten
 - Form „FormBuildingdetails“ – Detailansicht für ein Gebäude
 - Übergabeparameter: betrachtetes Gebäude, Liste aller insg. vorhandenen Stockwerke
 - Anzeigen und (teilw.) Ändern von Detailinformationen des Raums

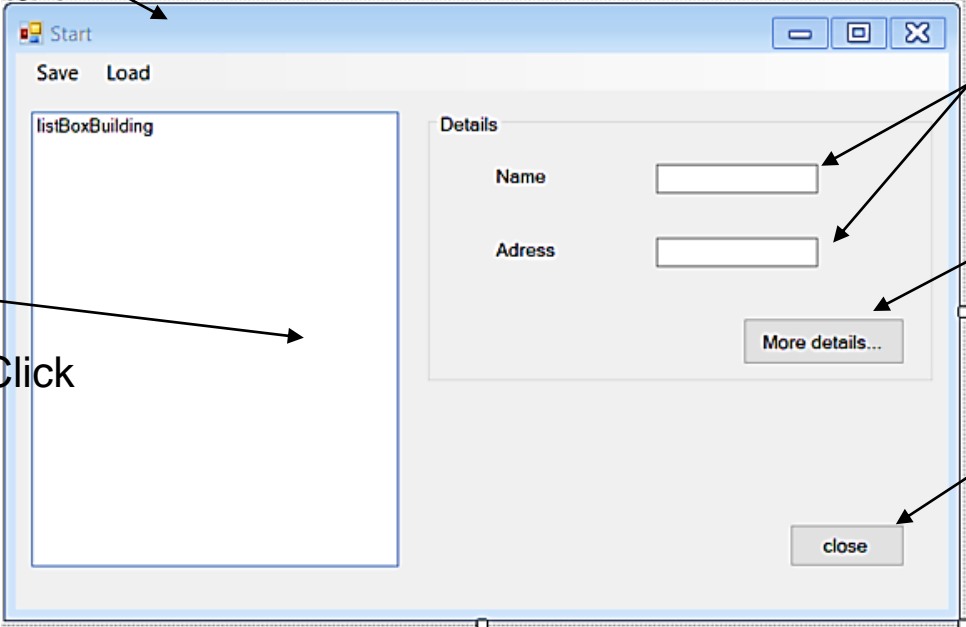
Workflow (1)



Hands On !

Designt das Hauptfenster für die Anwendung

Form „FormStart“



ListBox „listBoxBuilding“

- Ereignis bei Click

TextBox „textBoxName“ & „textBoxAdress“

Button „buttonMoredetails“

- Ereignis bei Click

Button „buttonSchliessen“

- Ereignis bei Click

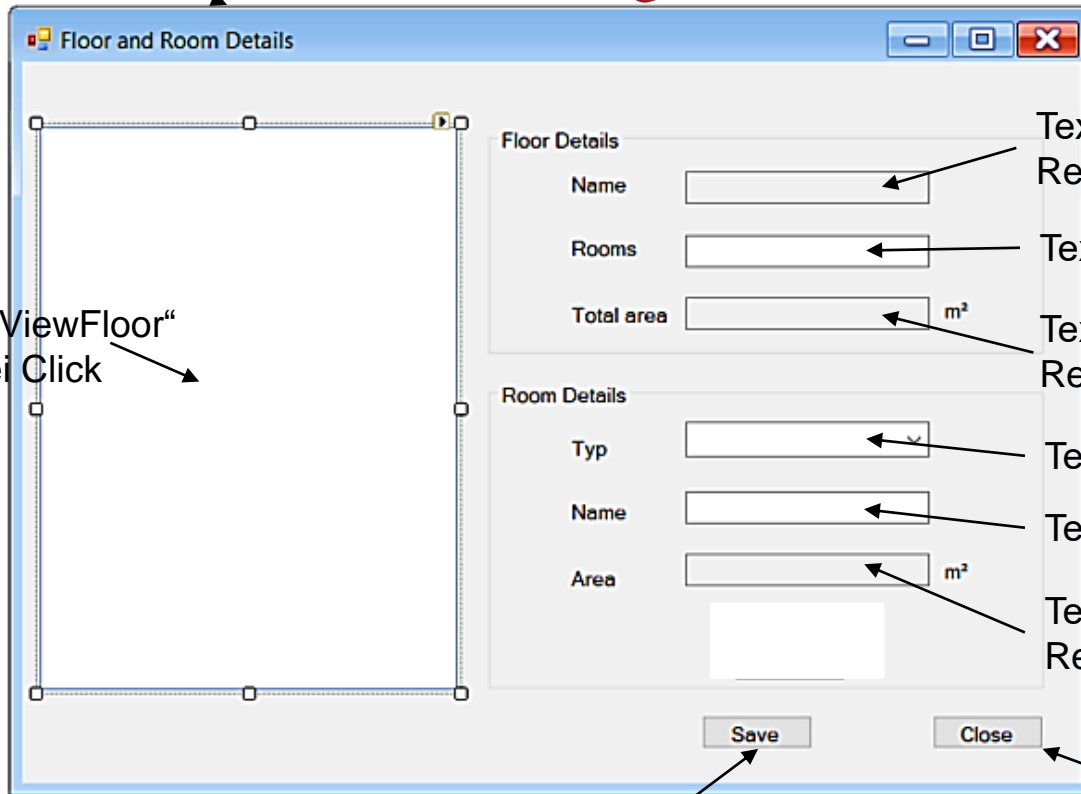
Dauer: 10 Minuten

Hands On ! - Designt das Detailform für Räume

Dauer: 10 Minuten

Form „FormBuildingdetails“

TreeView „treeViewFloor“
▪ Ereignis bei Click



TextBox „textBoxFloorName“
ReadOnly = „True“

TextBox „textBoxRoomcount“

TextBox „textBoxFloorArea“
ReadOnly = „True“

TextBox „comboBoxRoomTyp“

TextBox „textBoxRoomname“

TextBox „textBoxRoomarea“
ReadOnly = „True“

Button „buttonSave“
▪ Ereignis bei Click

Button „buttonClose“
▪ Ereignis bei Click

Hands On !



TECHNISCHE
UNIVERSITÄT
DARMSTADT

**Importiert die Klassenbibliothek „IIB1_Demonstrator“
in die GUI-Anwendung „IIB1_Demonstrator_GUI“**

- Klassenbibliothek „builden“ (Rechtsklick)
- Erstell einer GUI Anwendung (Windows Forms App)
- Verweis der GUI Anwendung hinzufügen
- Browse nach Debug-Ordner der Klassenbibliothek
- .dll Datei auswählen

Dauer: 5 Minuten

Hands On !



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Schreibt eine Methode, die Beispiel-Objekte automatisch generiert

- Beispiele Gebäude-Objekte instanzieren mit random Name
- Stockwerke-Objekte instanzieren durch Zufallszahl und sie in die Attribute “Stockwerk-List” des Gebäudes hinzufügen (for/ foreach, List.Add())
- Büro und Aufenthaltsräume Objekte generieren und sie in Stockwerk Attribute “Räume-List” hinzufügen (for/ foreach, List.Add())

Dauer: 15 Minuten

Methode: expGenerate()



```
private static void expGenerate()
{
    Random rd = new Random();

    for (int i = 0; i<3; i++)
    {
        string name = "Building " + rd.Next(1, 4).ToString() + "a";
        string adress = rd.Next(1, 75).ToString() + "Street";
        int levelnumber = rd.Next(1, 5);
        BindingList<Floor> floors = new BindingList<Floor>();
        for (int j = 0; j < levelnumber; j++)
        {
            string vname = j.ToString();
            string levelname = "Floor" + vname;
            int officenumber = rd.Next(1, 4);
            int livenumber = rd.Next(1, 4);

            BindingList<Room> rooms = new BindingList<Room>();
```

Methode: expGenerate()



```
for (int k= 0; k < officenumber; k++)
{
    string offname = "Office " + vname + k.ToString();
    double area = Math.Round(rd.NextDouble() * rd.Next(50, 100),1);
    rooms.Add(new Office(offname,area));
}

for (int k = 0; k < livenumber; k++)
{
    string livename = "Livingroom " + vname + k.ToString();
    double area = Math.Round(rd.NextDouble() * rd.Next(50, 100),1);
    rooms.Add(new Livingroom(livename, area));
}
floors.Add(new Floor(levelname, rooms));
}
buildings.Add(new Building(name,address,floors));
}
}
```

Hands On !



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Benutzerdefinierter Konstruktor und Füll-Methode für Steuerelemente im „FormStart“

- Erstellt einen Konstruktor für „FormStart“ mit Übergabeparameter `BindingList<Building>`
- Übernommene Gebäudeliste zu globale Attribute zuweisen
- Schreibt eine Methode, die die Steuerelemente des „FormStart“ mit Gebäudeliste ausfüllt

Dauer: 10 Minuten

```
public FormStart(BindingList<Building> buildings)
{
    InitializeComponent();
    this._buildings = buildings;
    filllist();
}

private void filllist()
{
    foreach (Building b in _buildings)
    {
        listBoxBuilding.DataSource = _buildings;
        listBoxBuilding.DisplayMember = "Name";
    }
}
```

Hands On!



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Click-Events

- Beim Klicken der Elemente in Listbox die Gebäudeinformation zur globalen Attribute “**Building** _build” zuweisen und Gebäudename und -adress in Textbox anzeigen
- Erstell einen Konstruktor für „FormBuildingdetailst (**Building** b, **FormStart** _parent)
- Beim Klicken Button “More details...” neue Fenster “FormBuildingdetails” aufrufen

Dauer: 15 Minuten

In FormStart

```
private void listBoxBuilding_SelectedIndexChanged(object  
    sender, EventArgs e)  
{  
    _build = (Building)listBoxBuilding.SelectedItem;  
    textBoxName.Text = _build.Name;  
    textBoxAdress.Text = _build.Adress;  
}
```

In FormStart

```
private void buttonMoredetails_Click(object sender, EventArgs e)
{
    (new FormBuilderdetails(_build, this)).ShowDialog();
}
```

In FormBuilderdetails

```
public FormBuilderdetails(Building b, FormStart _parent)
{
    InitializeComponent();
    _build = b;
    this.Owner = _parent;
    filltreeview();
}
```

Instanz als Owner von FormBuilderdetails

Nächste Aufgabe

Hands On!



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Schreibt eine Methode, die die Steuerelemente des „FormBuildingdezails“ füllt

- Parent-Nodes, Child-Nodes
- Foreach
- `TreeView.Nodes.Add()`

Dauer: 15 Minuten


```
private void loadNodes()
{
    foreach (Floor f in _build.Floors)
    {
        TreeNode tn = treeViewFloor.Nodes.Add(f.Name);
        tn.Tag = f;
        addChildNode(tn);
    }
}
private void addChildNode(TreeNode tn)
{
    foreach (Room r in ((Floor)tn.Tag).Rooms)
    {
        TreeNode tnChild = tn.Nodes.Add(r.Name);
        tnChild.Tag = r;
    }
}
```

Dynamisches Anzeigen TreeView-Nodes



```
private void filltreeview()  
{  
    treeViewFloor.Nodes.Clear();  
    treeViewFloor.BeginUpdate();  
    loadNodes();  
    treeViewFloor.EndUpdate();  
}
```

Hands On !



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Fügt Methoden zum Serialisieren und Deserialisieren der Raumliste dem FormMain hinzu

- Events „ladenToolStripMenuItem_Click(...)“ und „speichernToolStripMenuItem_Click(...)“
- Aufrufen eines SafeFileDialogs bzw. OpenFileDialogs

Dauer: 10 Minuten

Event speichernToolStripMenuItem_Click(...)



```
private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    saveToFile();
}

public void saveToFile()
{
    SaveFileDialog sfd = new SaveFileDialog();
    if (sfd.ShowDialog() == DialogResult.OK)
    {
        FileStream fs = new FileStream(sfd.FileName,
        FileMode.Create);
        BinaryFormatter bf = new BinaryFormatter();
        bf.Serialize(fs, _buildings);
        fs.Close();
    }
}
```

Event ladenToolStripMenuItem_Click(...)

```
private void loadToolStripMenuItem_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    if (ofd.ShowDialog() == DialogResult.OK)
    {
        FileStream fs = new FileStream(ofd.FileName,
        FileMode.Open);

        BinaryFormatter bf = new BinaryFormatter();

        _buildings =
        (BindingList<Building>)bf.Deserialize(fs);
        fs.Close();
        filllist();
    }
}
```

Hands On !



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Übergabe des veränderten Raumes aus dem FormBuildingdetails in das FormStart

- Erstellen Methode in FormBuildingdetails
 - Löscht in TreeView ausgewählten Raum aus Raumliste
 - Fügt veränderten Raum der Raumliste zu
 - Wählt veränderten Raum in der ListBox aus
- Aufrufen der erstellten Methode aus der FormStart
 - Nach Änderung des Raums (buttonSave_Click(...))

Dauer: 10 Minuten



```
private void buttonSave_Click(object sender, EventArgs e)
{
    if (_room == null)
    {
        MessageBox.Show("No changes have been saved. Be sure the Room belongs to the Floor!","Attention!");
    }
    else
    {
        Room r = treeViewFloor.SelectedNode.Tag as Room;
        r.Name = textBoxRoomname.Text;
        r.Area = Double.Parse(textBoxRoomarea.Text);

        int i = _floor.Rooms.IndexOf(_room);
        if (comboBoxRoomTyp.SelectedIndex == 0)
        {
            _room = new Livingroom(r);
        }
        else
        {
            _room = new Office(r);
        }
        _floor.Rooms[i] = _room;
        treeViewFloor.SelectedNode.Tag = _room;
        treeViewFloor.SelectedNode.Text = _room.Name;
        treeViewFloor.Refresh();

        // Allow a Dialog Form "Buildingdetails" to Retrieve Information from its Calling Form "Start"
        ((FormStart)this.Owner).saveToFile();
    }
}
```

Neu bearbeitete Rauminstanz zuweisen
und BindingList aktualisieren

Aktualisierung Steuerelemente (Werte für
Raum verändert)