1 Organisatorisches

1.1 Team

- Reinhard Penn, s1110306019
- Bernhard Selymes, s1110306024

1.2 Aufteilung

- Reinhard Penn
  - Planung
  - Klassendiagramm
  - Implementierung der Klassen Object, Roomlayout, Room, Side, Wall, Door
  - Testen aller Klassen
- Bernhard Selymes
  - Planung
  - Klassendiagramm
  - Implementierung der Klassen Object, Roomlayout, Room, Side, Wall, Door
  - Dokumentation

1.3 Zeitaufwand

- geschätzte Mh: 7h
- tatsächlich: Reinhard (10h), Bernhard (10h)

2 Systemspezifikation

Eine Software für einen Raumplan soll entwickelt werden. Ein Raumplan enthält mehrere Räume, jeder Raum hat 4 Seiten, die wiederum eine Wand oder ein Durchgang sein können. Die Räume sind alle gleich groß und liegen alle untereinander und nicht nebeneinander. Eine Wand hat eine Farbe, ein Durchgang ist offen oder geschlossen und verbindet einen oder zwei Räume - es gibt auch Durchgänge die nach draußen führen. Wenn zwei Durchgänge aufeinander treffen wird nur einer ausgegeben. Bei zwei Wänden werden beide ausgegeben. In der gesamten Datenstruktur können nur Objekte hinzugefügt, aber nicht gelöscht werden. Der gesamte Raumplan kann ausgedruckt werden.

3 Systementwurf

3.1 Klassendiagramm

3.2 Komponentenübersicht

- Klasse ”Object”:
  Basis aller Basisklassen.
- Klasse ”RoomLayout”:
  Beinhaltet die Räume und kann diese mithilfe einer Printfunktion ausgeben.
- Klasse ”Room”:
  Beinhaltet vier Seiten und kann mithilfe einer Printfunktion ausgegeben werden.
- Klasse ”Side”:
  Abstrakte Klasse, von der Wall und Door abgeleitet werden. Speichert die Himmelsrichtung der Seite.
- Klasse ”Wall”:
  Hat einen Member der die Farbe der Wand speichert.
- Klasse ”Door”:
  Speichert ob Door offen oder geschlossen ist und welche Räume von Door verbunden werden.
- Enumeration ”Direction”:
  Hat vier Zustände: North, West, East, South

4 Komponentenentwurf

4.1 Klasse ”Object”

Abstrakte Basisklasse aller Klassen. Von ihr werden alle anderen Klassen abgeleitet. Beinhaltet einen virtuellen Destruktor.

4.2 Klasse ”RoomLayout”

Besitzt eine Liste die Zeiger auf die Klasse Room beinhaltet, einen Zeiger, der auf den vorherigen Raum zeigt und einen Wahrheitswert, der angibt ob im letzten Raum im Süden ein Durchgang war.

Methode ”Print”: Gibt die Räume die in der Liste gespeichert sind der Reihe nach aus und merkt sich immer ob im vorherigen Raum eine Durchgang im Süden war. Wenn ein Durchgang war wird der Durchgang nur einmal ausgegeben.

Methode ”AddRoom”: Fügt Räume zum Raumplan hinzu. Ein Raum wird nur dann hinzugefügt wenn die südliche Seite der vorherigen Raumes mit der nördlichen Seite des Raumes, der eingefügt werden soll, übereinstimmt oder der gesamte Raumplan noch leer ist. Wenn ein Durchgang war: Im vorherigen Raum wird bei Durchgang der aktuelle Raum hinzugefügt und beim Durchgang vom aktuellen Raum wird der vorherigen Raum gespeichert. Weiters wird gespeichert ob der aktuelle Raum einen Durchgang im Süden hat und der Zeiger, der auf den vorherigen Raum gezeigt hat, zeigt jetzt auf den aktuellen Raum.

4.3 Klasse ”Room”

Beinhaltet einen Vektor der Zeiger auf die Klasse Side hat.

Methode ”Print”: Gibt den Raum aus. Durchgang wird im Norden nicht ausgegeben, wenn im letzten Raum im Süden ein Durchgang war. Die restlichen Seiten werden ganz normal ausgegeben, eine Wand im Norden oder Süden wird durch 9 Sterne gekennzeichnet, im Westen oder Osten durch 5. Ein Durchgang wird durch ein ”D” gekennzeichnet.

Methode ”AddSide”: Fügt Seiten zu einem Raum hinzu. Eine Seite wird nur dann hinzugefügt, wenn noch keine Seite für diese Himmelsrichtung existiert. Der Vektor mit den Seiten wird nach dem Einfügen nach der Definiton der Enumeration sortiert: North, West, East, South. Wenn eine Seite ein Durchgang ist, wird im Durchgang der aktuelle Raum hinzugfügt.

4.4 Klasse ”Side”

Abstrakte Klasse, die eine Seite eines Raumes repräsentiert. Enthält einen Member der die ”Direction” der Seite speichert und einen der angibt ob es ein Durchgang oder eine Wand ist.

4.5 Klasse ”Wall”

Repräsentiert eine Wand. Hat einen Konstruktor dem die Farbe und die Richtung der Wand übergeben werden.

4.6 Klasse ”Door”

Repräsentiert einen Durchgang. Hat einen Member der speichert ob die Tür offen oder geschlossen ist und einen Vektor der die Räume speichert, die der Durchgang verbindet.

Methode ”AddRoom”: Fügt zum Durchgang einen Raum hinzu, wenn noch nicht die maximale Anzahl der Räume die ein Durchgang verbinden kann erreicht ist.

4.7 Enumeration ”Direction”

- North
- West
- East
- South

5 Testausgaben

```
Testcase0: Empty roomlayout


Testcase1: Roomlayout with empty rooms
Error occured in RoomLayout::AddRoom: Room is not full
Error occured in RoomLayout::AddRoom: Room is not full


Testcase2: Roomlayout with wrong rooms
*********
*       *
*       D
*       *
****D****


Testcase3: Roomlayout with correct rooms
*********
*       *
*       D
*       *
****D****
*       *
*       D
*       *
****D****


Testcase4: Roomlayout with correct rooms.
Trying to add a fifth side to one room
Error occured in Room::AddSide: Max amount of walls already reached
*********
*       *
*       D
*       *
*********
*********
*       *
*       D
*       *
****D****
```