

1 Organisatorisches

1.1 Team

- Reinhard Penn, s1110306019
- Bernhard Selymes, s1110306024

1.2 Aufteilung

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

1.3 Zeitaufwand

- geschätzte Mh: 7h
- tatsächlich: Reinhard (?h), Bernhard (6h)

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 Roomplan: Verwaltet mehrere Rooms und kann diese mit Print() ausgeben.
- Klasse Room: Enthält Zeiger auf vier Sides , die jeweils Wall oder Entry sein können
- Klasse Side: Abstrakte Basisklasse für Wall und Entry
- Klasse Wall: Enthält einen Member der die Farbe speichert
- Klasse Entry: Enthält Zustand (offen, geschlossen) und Id der angrenzenden Räume

4 Komponentenentwurf

4.1 Klasse Objekt

Basis aller Basisklassen. Enthält einen virtuellen Destruktor und die enum Typen:

- Typ - Entweder WALL oder ENTRY
- Colour - Verschiedene Farben: WHITE, GREEN, BLUE, RED
- Dir - Enthält Himmelsrichtungen: NORTH, SOUTH, EAST, WEST

4.2 Klasse Roomplan

Klasse zu Verwaltung aller Räume. Enthält einen Vector der alle Räume speichert und einen Zähler der mit zählt wie viele Räume schon erstellt wurden. IDs der Räume beginnen bei 0.

Methoden:

void Print() const

Diese Funktion gibt die enthaltenen Räume untereinander aus. Für Wände werden Sternchen ausgegeben und für Durchgänge Sternchen mit DD dazwischen. Wenn zwei Durchgänge aneinanderschließen, wird nur einer ausgegeben.

void Roomplan::Add(Typ n, Typ s, Typ e, Typ w, Colour const & col, bool open)

Diese Funktion bekommt die Typen der vier Wände übergeben. Es wird überprüft, ob der Raum angefügt werden kann. Wenn dies möglich ist, wird der Raum mittels push_back in den vector eingefügt, sonst wird ein Fehler auf die Konsole ausgegeben. Bei dieser Funktion haben alle Wände die gleiche Farbe und die Durchgänge sind alle offen oder geschlossen.

void Roomplan::Add(Side* NS, Side* SS, Side* ES, Side* WS)

Diese Funktion bekommt die fertigen Side* Objekte übergeben, die entweder Wall oder Entry sein können. Es wird auch wieder überprüft, ob der Raum zum darüberliegenden passt.

int GetCounter () const

Diese Funktion gibt den derzeitigen Raumzähler zurück. Der Raumzähler zählt mit wieviele Räume schon erstellt wurden.

4.3 Klasse Room

Klasse die vier Zeiger auf Side (bzw. Wall oder Entry) hat, die die Seiten des Raumes darstellen. Hat auch einen Member mId welcher die Id des Raums angibt.

Konstruktoren:

Room(Typ n, Typ s, Typ e, Typ w, size_t id, Colour const &col, bool const &open)

Dieser Konstruktor bekommt die Typen der vier Wände, die Id, die Farbe für alle Wände und den Zustand der Durchgänge. Je nach dem ob die Seite eine Wand oder ein Durchgang ist, wird die entsprechende mit new erstellt.

Room::Room(Side* NS, Side* SS, Side* ES, Side* WS, size_t id)

Hier werden die Seiten als Zeiger übergeben. Hier müssen die Objekte, die hinter den Zeigern stehen, kopiert werden. Das wird mithilfe der Funktion Clone() erreicht, welche bei der Klasse

Side, Wall und Entry noch genauer erklärt wird. Wenn man die Zeiger einfach normal zuweisen würde, wäre es eine shallow-copy.

Copy Konstruktor:

Room::Room (Room const& room)

Hier werden die Zeiger auf die Side* Objekte auch mittels Clone kopiert.

Destruktor:

Löscht die Zeiger auf die Seiten und setzt sie zu 0.

Get Methoden:

Die Get...Wall() Methoden geben den Typ der jeweiligen Seite zurück.

Die Funktion GetRoomId() gibt die Id zurück.

void Room::PrintRoomInfo () const

Diese Funktion gibt die Id und die Information der einzelnen Seiten aus. Ruft PrintInfo() von Side auf, welches dort näher beschrieben wird.

4.4 Klasse Side

Die Klasse Side ist eine abstrakte Basisklasse für die Klassen Wall und Entry. Enthält nur pure-virtual Funktion, kann also auch nicht implementiert werden.

Konstruktoren:

Side () {}

Default Konstruktor.

Side (Typ typ) : mTyp(typ) {}

Direktes Zuweisen des Typs.

pure virtual Methoden:

virtual Typ GetTyp () const = 0;

Gibt Typ der Seite zurück (WALL/ENTRY).

virtual void PrintInfo () const = 0;

Gibt Infos über die abgeleitete Klasse aus.

virtual Side *Clone() = 0;

Klont die abgeleitete Klasse.

4.5 Klasse Wall

Die Klasse Wall repräsentiert eine Wand. Ist abgeleitet von Side und implementiert dessen Methoden.

Konstruktoren:

Wall () : mColour(WHITE), Side(WALL) {}

Default Konstruktor.

Wall (Colour col) : mColour(col), Side(WALL) {}

Konstruktor der die Farbe übergeben bekommt und diese richtig setzt.

Wall (Wall const& wall)

Copy - Konstruktor. Kopiert Typ und Colour vom übergebenen Objekt.

Get Methoden:

GetTyp () gibt Typ zurück.

GetCoulour () gibt Farbe zurück.

PrintInfo () gibt Typ und Farbe auf der Konsole aus.

Clone Methode:

Wird verwendet um ein Objekt der Klasse Wall zu kopieren. Erstellt eine neues Objekt Wall und gibt den Zeiger darauf zurück.

4.6 Klasse Entry

Die Klasse Entry repräsentiert einen Durchgang. Ist abgeleitet von Side und implementiert dessen Methoden.

Konstruktoren:**Entry (int currentId, bool opened, Dir const& dir)**

Dieser Konstruktor bekommt die derzeitige Id des Raumplans, ob der Durchgang geöffnet ist und die Richtung übergeben.

Typ, Id und Opened werden entsprechend gesetzt. Wenn der Durchgang nach Norden zeigt, wird die Id des angrenzenden Raums auf id - 1 gesetzt, wenn er nach Süden zeigt auf id + 1 und sonst auf -1 also der Durchgang geht nach aussen.

Entry::Entry (Entry const& entry)

Copy Konstruktor: Kopiert Member vom übergebenen Objekt.

Get Methoden und Print

Typ GetTyp () const;

bool IsOpened () const;

int GetMyRoomId () const;

int GetOtherRoomId () const;

void PrintInfo () const;

Geben die jeweiligen Informationen zurück bzw. druckt sie auf der Konsole aus.

Clone Methode:

Wird verwendet um ein Objekt der Klasse Entry zu kopieren. Erstellt eine neues Objekt Entry und gibt den Zeiger darauf zurück.

5 Testprotokollierung

6 Source Code

7 Testausgaben