

Name(1): Daniel Weyrer

Abgabetermin: 14.01.2020

Name(2): Viktoria Streibl

Punkte:

Übungsgruppe: 1

korrigiert:

Geschätzter Aufwand in Ph: 6 | 6

Effektiver Aufwand in Ph: 6 | 3

**Beispiel 1 (24 Punkte) Robotersteuerung:** Entwerfen Sie aus der nachfolgend gegebenen Spezifikation ein Klassendiagramm, instanzieren Sie dieses, implementieren Sie die Funktionalität entsprechend und verwenden Sie dazu das Command-Pattern:

Für eine Robotersteuerung soll eine Software implementiert werden die verschiedenen Robotern (Hexapod-Roboter, Radroboter,...) unterschiedliche Kommandos geben kann.

Die Steuerung hat unter anderem folgende Schnittstelle:

```
1 void AddCommand(*Command*); //adds a command to command list
2 void Start(); //starts execution of all commands in list
3 void Undo(size_t const count); //undo and removes specified count of commands
4 void Reset(); //removes all commands
```

Folgende Kommandos sollen implementiert werden:

- TurnLeft: Der Roboter dreht sich um eine Vierteldrehung nach links.
- TurnRight: Der Roboter dreht sich um eine Vierteldrehung nach rechts.
- Forward: Der Roboter bewegt sich in die aktuelle Richtung.
- MacroMovement: Dem Roboter können mehrere Kommandos in einem mitgeteilt werden.

Ein Roboter speichert einen Namen, die aktuelle Position (x,y-Koordinaten) und die Richtung (Norden, Osten, Süden, Westen).

Eine Client-Klasse verwaltet die einzelnen Roboter und gibt deren aktuellen Zustände auf `std::cout` aus. Dazu bieten die Roboter folgende Schnittstelle an:

```
1 class IRobot
```

```
2 {  
3 public:  
4     virtual void Info(std::ostream& os) const = 0;  
5     virtual ~IRobot();  
6 };
```

Die Ausgabe der Roboterinformation sieht folgendermaßen aus:

```
WheelRobot: Robin, Pos(300,900), Direction(WEST)  
Hexapod: Maximus, Pos(1300,900), Direction(EAST)  
Hexapod: Paulo, Pos(-200,400), Direction(SOUTH)  
...
```

Testen Sie ausführlich alle Funktionen und die Steuerung der Roboter. Treffen Sie für alle unzureichenden Angaben sinnvolle Annahmen. Verfassen Sie weiters eine Systemdokumentation (Funktionalität, Klassendiagramm, Schnittstellen der beteiligten Klassen, etc)!

**Allgemeine Hinweise:** Legen Sie bei der Erstellung Ihrer Übung großen Wert auf eine **saubere Strukturierung** und auf eine **sorgfältige Ausarbeitung!** Dokumentieren Sie alle Schnittstellen und versehen Sie Ihre Algorithmen an entscheidenden Stellen ausführlich mit Kommentaren! Testen Sie ihre Implementierungen ausführlich! Geben Sie den **Testoutput** mit ab!