

Aufgabe – Device

Es soll die Klassenhierarchie mit Vererbung sowie der Zugriff mit Hilfe von Zeigervariablen geübt werden.

Als Anwendung dient die softwaremäßige Unterstützung (Simulation) einer Fernbedienung für ein „Smart House“, mittels derer man die Steuerung von elektrischen Haushaltsgeräten durchführen kann.

Dazu entwerfe man eine Klassenhierarchie bestehend aus einer Basisklasse (allgemeines Gerät = CDevice) und drei abgeleitete Subklassen (spezielle Geräte: Kaffeemaschine, Heizung, Radio).

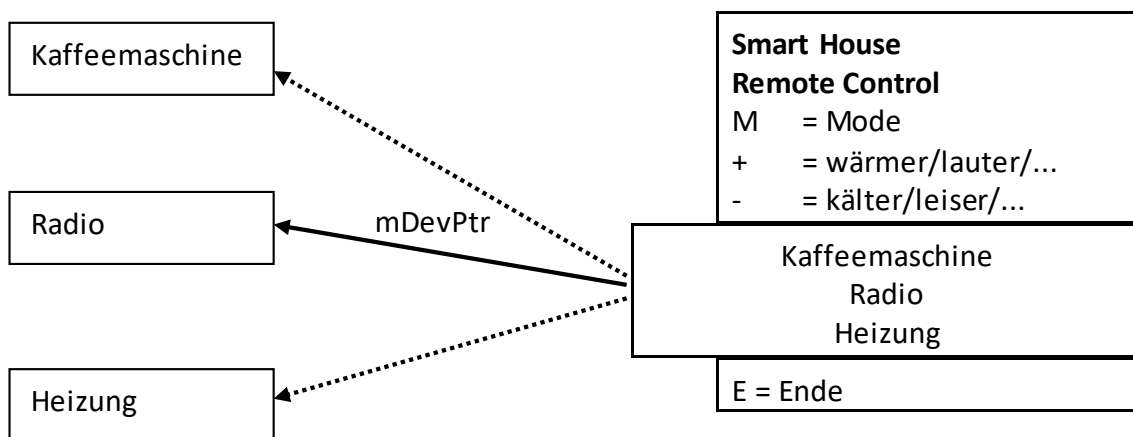
Neben den Konstruktoren sehe man die Methoden „Plus“ und „Minus“ zum Hoch- und Herunterregeln eines speziellen Attributs jedes Geräts vor (Kaffeemenge, Temperatur, Lautstärke) sowie eine einfache Ausgabe des Namens und des Zustandswertes des jeweiligen Objekts vor.

Aufgabenstellung

Das Programm soll zunächst im Konsolendialog die Tasten der Fernbedienung abfragen und die entsprechende Aktion auslösen.

Dafür wird in dem Fernbedienungsobjekt die Methode des entsprechenden Device -Objekts aufgerufen.

Es gibt eine Taste zum Weiterschalten des aktuellen Geräts (Mode), das per Zeigervariable auf das ausgewählte Gerät zeigt.



Realisierung

Mit der Kennzeichnung einer Methode als `virtual Fkt = 0`; teilt man dem Compiler mit, dass die Basisklasse keine Implementierung anbietet, jede vererbte Klasse die Methode aber implementieren muss. Man nennt diese Basisklasse eine 'abstrakte Klasse'. Von ihr können auch keine Objekte instanziiert werden. Außerdem sorgt die `virtual`-Deklaration dafür, dass per 'late Binding' die Klassenfunktion des Objektes aufgerufen wird, auf das der Zeiger aktuell zeigt.

Verwenden Sie die folgende Headerdatei "CDevice.h":

```
1  #pragma once
2  #include <iostream>
3  using namespace std;
4
5  class CDevice
6  {
7  public:
8      virtual void mPlus() = 0;
9      virtual void mMinus() = 0;
10     virtual void mZustandsausgabe() = 0;
11     protected:
12         int mInternerZustand;
13         string mDeviceName;
14 };
15
16 class CFernbedienung :public CDevice
17 {
18 public:
19     enum Mode { Kaffeemaschine, Radio, Heizung };
20     CFernbedienung(string deviceName, CDevice* Dev1Ptr, CDevice* Dev2Ptr, CDevice* Dev3Ptr);
21         // Hier werden die Adressen der Device-Objekte übergeben.
22     virtual void mPlus();
23     virtual void mMinus();
24     virtual void mZustandsausgabe();
25     void mModeChange();
26
27 private:
28     CDevice* mDevPtr;        // Dieser Zeiger zeigt auf das aktuelle, per Mode-Taste selektierte Objekt.
29                             // Damit können dann direkt dessen Methoden aufgerufen werden.
30     CDevice* mDev1Ptr;      // Hier werden die Adressen der Device-Objekte aufgehoben.
31     CDevice* mDev2Ptr;
32     CDevice* mDev3Ptr;
33     Mode    mMode;         // Hier wird der Status der Mode-Taste aufgehoben.
34 };
35
36
37 class CKaffeemaschine :public CDevice
38 {
39 public:
40     CKaffeemaschine(string deviceName, int i);
41     virtual void mPlus();
42     virtual void mMinus();
43     virtual void mZustandsausgabe();
44 };
45
46 class CRadio :public CDevice
47 {
48 public:
49     CRadio(string deviceName, int i);
50     virtual void mPlus();
51     virtual void mMinus();
52     virtual void mZustandsausgabe();
53 private:
54 };
55
56
57 class CHeizung :public CDevice
58 {
59 public:
60     CHeizung(string deviceName, int i);
61     virtual void mPlus();
62     virtual void mMinus();
63     virtual void mZustandsausgabe();
64 };
```

Ein Beispielhafter Programmlauf könnte wie folgt aussehen:

```
Microsoft Visual Studio-Debugging-Konsole
Fernbedienung_1 - Mode: Tschibo-Automat_1 - Kaffeemenge steht auf 80
  bitte Eingabe, M(ode), +, -, E(nde)
+
Tschibo-Automat_1: Mehr Kaffeepulver
Fernbedienung_1 - Mode: Tschibo-Automat_1 - Kaffeemenge steht auf 81
  bitte Eingabe, M(ode), +, -, E(nde)
-
Tschibo-Automat_1: Weniger Kaffeepulver
Fernbedienung_1 - Mode: Tschibo-Automat_1 - Kaffeemenge steht auf 80
  bitte Eingabe, M(ode), +, -, E(nde)
M
Fernbedienung_1 - Mode: Radio_1 - Volume steht auf 5
  bitte Eingabe, M(ode), +, -, E(nde)
+
Radio_1: Lauter
Fernbedienung_1 - Mode: Radio_1 - Volume steht auf 6
  bitte Eingabe, M(ode), +, -, E(nde)
-
Radio_1: Leiser
Fernbedienung_1 - Mode: Radio_1 - Volume steht auf 5
  bitte Eingabe, M(ode), +, -, E(nde)
M
Fernbedienung_1 - Mode: Heizung_1 - Temperatur steht auf 20
  bitte Eingabe, M(ode), +, -, E(nde)
+
Heizung_1: Heisser
Fernbedienung_1 - Mode: Heizung_1 - Temperatur steht auf 21
  bitte Eingabe, M(ode), +, -, E(nde)
-
Heizung_1: Kaelter
Fernbedienung_1 - Mode: Heizung_1 - Temperatur steht auf 20
  bitte Eingabe, M(ode), +, -, E(nde)
E

C:\Users\ewju0001\source\repos\Device\Debug\Device.exe (Prozess "19620") wurde m
it Code "0" beendet.
Um die Konsole beim Beenden des Debuggens automatisch zu schließen, aktivieren S
ie "Extras" > "Optionen" > "Debuggen" > "Konsole beim Beenden des Debuggings aut
omatisch schließen".
Drücken Sie eine beliebige Taste, um dieses Fenster zu schließen.
```

Geben Sie zusammen mit Ihrer Lösung wie gewohnt einen Screendump mit Ihrem Testlauf ab.

Viel Spaß bei der Bearbeitung!