

Facade Pattern

Entwurfsmuster

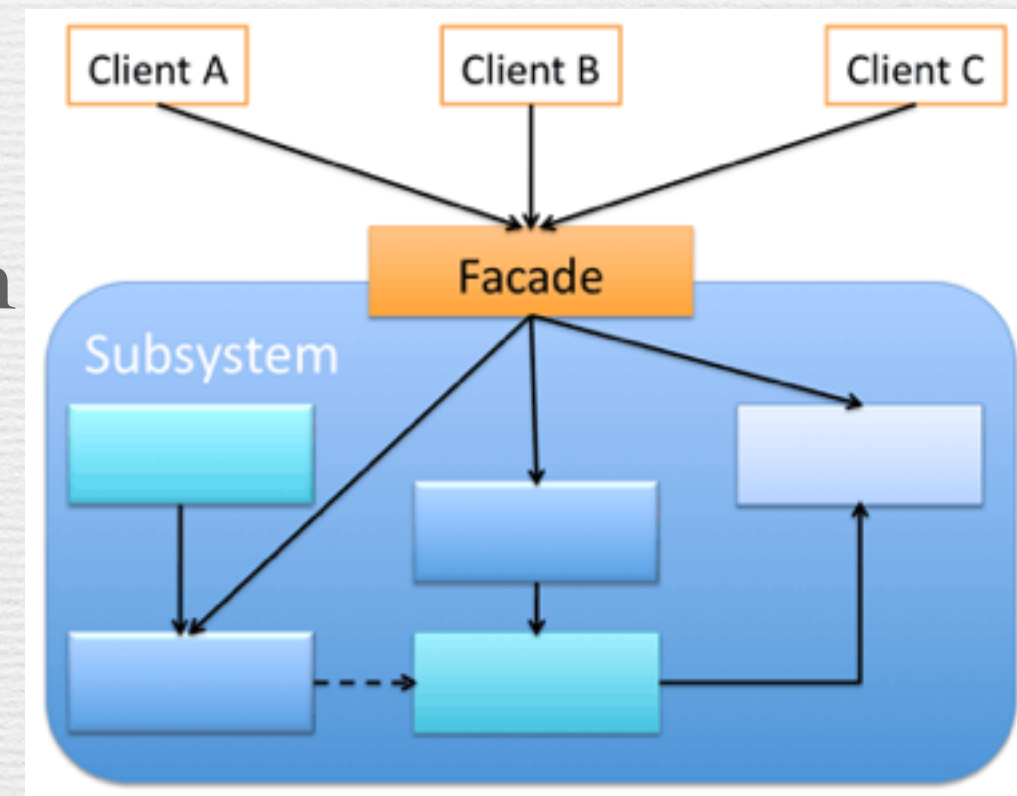
Inhalt

- Entwurfsmuster
- Definition
- Vor/ Nachteile
- Beispiel

Entwurfsmuster

Definition

- Vereinfachte Schnittstelle zur Benutzung eines System, oder Objekten
- Zwischen Client und Subsystem
- Kapselt Subsystem
- Bietet Client eine Schnittstelle
- ohne die einzelnen Klassen zu kennen



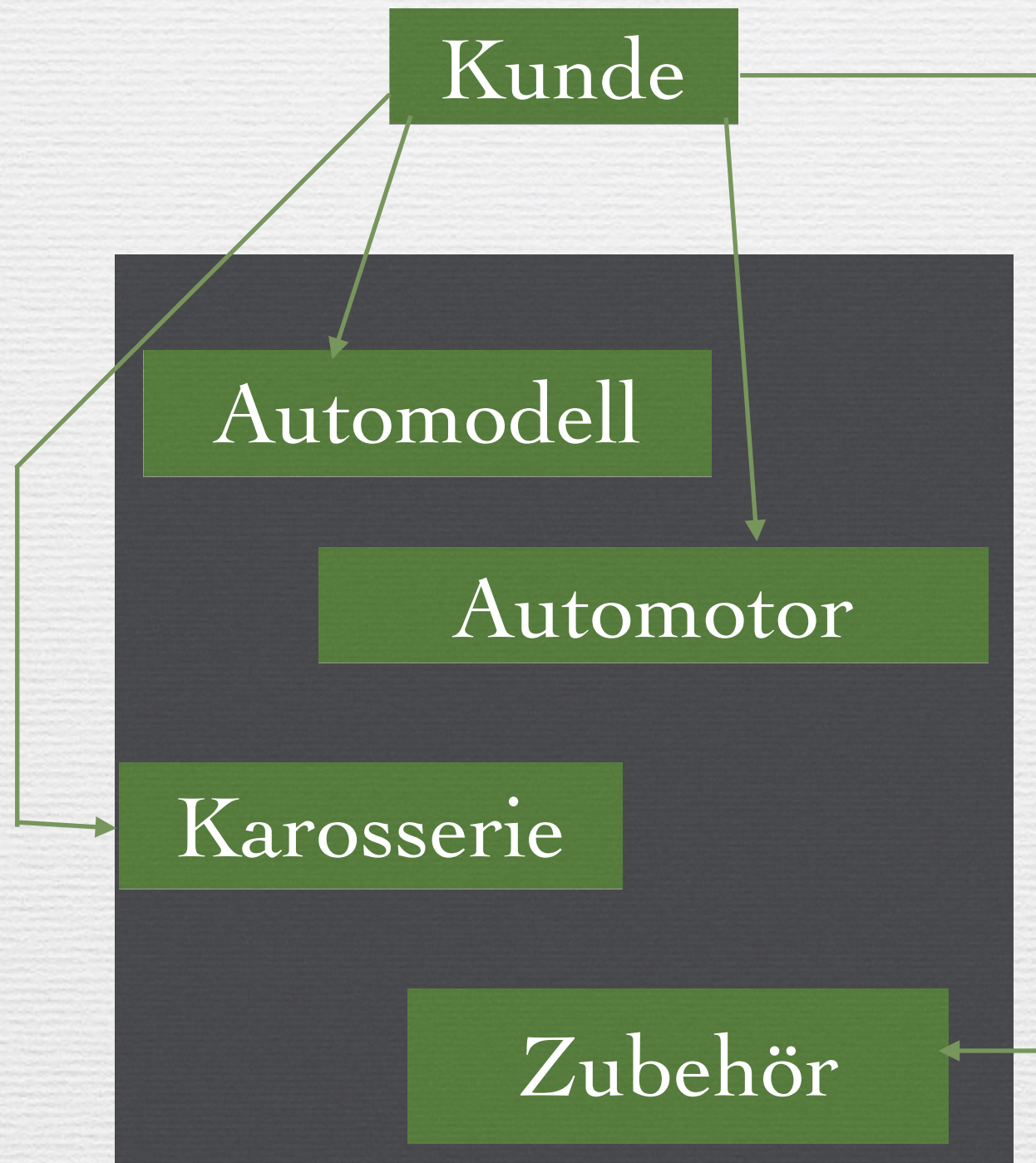
Vor/ Nachteile

- Vorteile:
 - ✦ Einfacher Zugriff auf Komplexe Systeme
 - ✦ Entkopplung
 - ✦ regelt Zugriff
 - ✦ Verschachteln von Systemen
 - ✦ verringerte Abhängigkeiten

Vor/ Nachteile

- Nachteile:
 - ✦ Nicht immer sinnvoll
 - ✦ jedes System bekommt eine zusätzliche Ebene

Beispiel - ohne Entwurfsmuster



Beispiel Code

```
class Automodell
{
public:
    void SetModel()
    {
        string carm;
        //.. definition des ausgewählten Automodells
        cout << "Automodell: " << carm << endl;
    }
};

class Automotor
{
public:
    void SetMotor()
    {
        string careng;
        //.. definition des ausgewählten Automotors
        cout << "Automodell: " << careng << endl;
    }
};

class karosserie
{
public:
    void SetMkarosserie()
    {
        string karosserie;
        //... definition der ausgewählten Karosserie
        cout << "Automodell: " << karosserie << endl;
    }
};
```

```
class zubehör
{
public:
    zubehör()
    {
        //...
    }

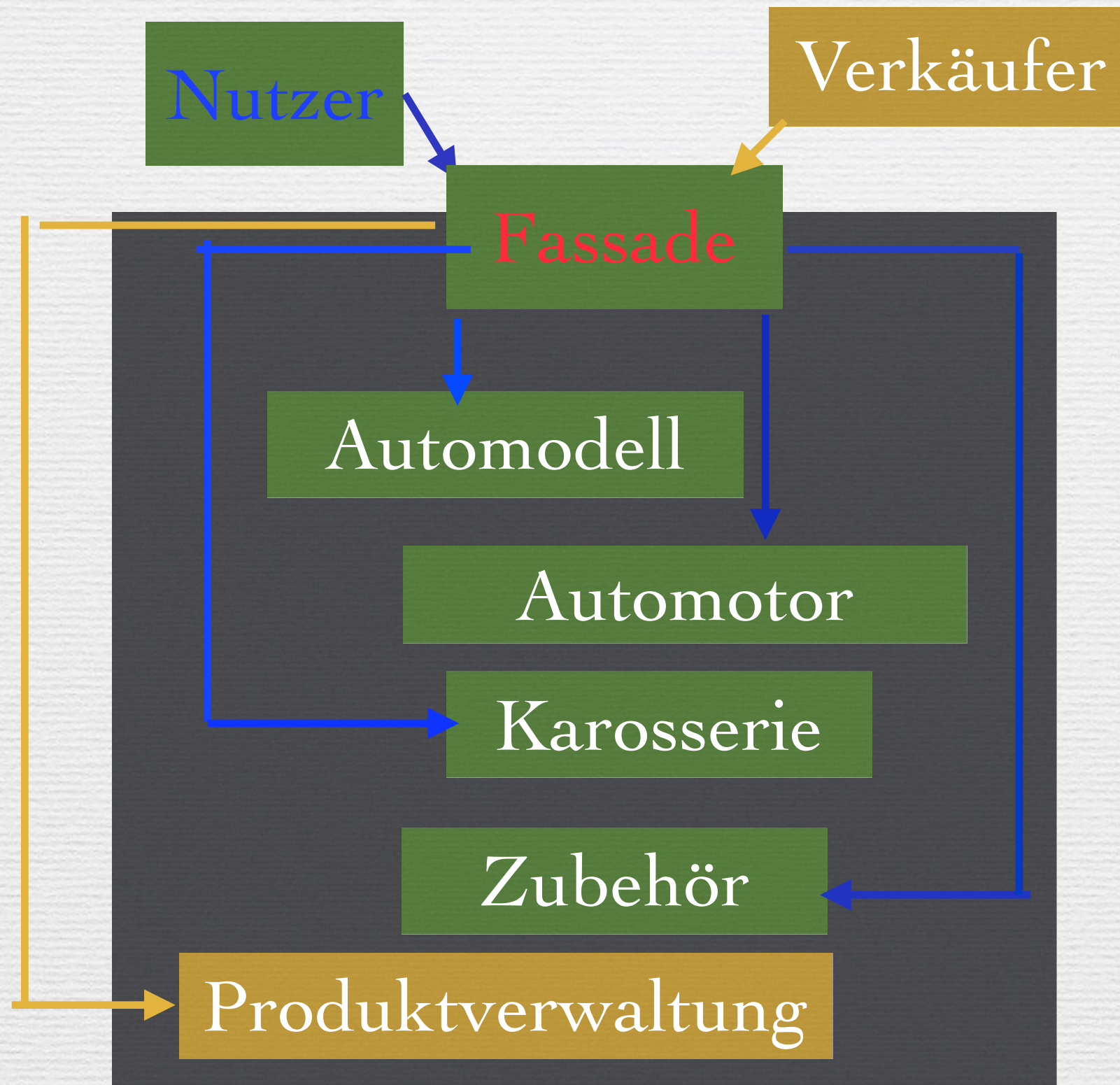
    void Setzubehör()
    {
        string zubehör;
        //... definition des ausgewählten Zubehörs
        cout << "Automodell: " << zubehör << endl;
    }
};

class käufer
{
public:
    Automodell modell;
    Automotor motor;
    karosserie karosserie;
    zubehör zubehör;

    käufer()
    {
        autozusammenstellen();
    }

    void autozusammenstellen()
    {
        cout << "Auto wird zusammen gestellt" << endl;
        modell.SetModel();
        motor.SetMotor();
        karosserie.SetMkarosserie();
        zubehör.Setzubehör();
        cout << "Auto wurde zusammen gestellt" << endl;
    }
};
```


Beispiel - mit Fassade



Beispiel mit Fassade- Code

```
class fassade
{
public:
    Automodell modell;
    Automotor motor;
    karosserie karosserie;
    zubehör zubehör;

    fassade()
    {
        // Klassen mit NullWerten initialisieren, ggf andere Routinen starten
    }

    void autozusammenstellen()
    {
        cout << "Auto wird zusammen gestellt" << endl;
        modell.SetModel();
        motor.SetMotor();
        karosserie.SetMkarosserie();
        zubehör.Setzubehör();
        cout << "Auto wurde zusammen gestellt" << endl;
    }

    void produktverwaltung()
    {
        //.... Produktverwaltung
    }
};
```

```
class käufer_fass
{
private:
    fassade fass;

public:
    käufer_fass()
    {
        fass.autozusammenstellen();
    }
};

class verkäufer_fass
{
private:
    fassade fass;

public:
    verkäufer_fass()
    {
        fass.produktverwaltung();
    }
};
```


Vielen Danke, für
ihre Aufmerksamkeit

Quellen

- <http://www.philippbauer.de/study/se/design-pattern/facade.php>
- [http://de.wikibooks.org/wiki/Muster: Facade](http://de.wikibooks.org/wiki/Muster:_Facade)
- <http://www.thewebhatesme.com/allgemein/entwurfsmuster-fassade-facade/>
- <http://www.dofactory.com/net/facade-design-pattern>