

Diagramy komponent

Diagramy komponent

- Ilustrují vztahy mezi softwarovými komponentami, jejich závislosti, komunikaci, umístění aj.

Diagramy komponent

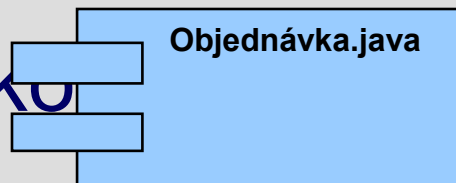
- Znázorňují statickou implementaci pohledu na systém
 - diagramy tříd, které jsou zaměřeny na systémové komponenty
- Jsou jedny ze dvou druhů diagramů, které modelují fyzickou organizaci OO systémů

Struktura diagramu komponent

- Diagramy komponent se obvykle skládají z:
 - Komponent
 - Rozhraní
 - Závislostí, dědičností, asociací a jiných vazeb

Komponenta

- Komponenta je fyzická součást systému, která využívá a poskytuje realizaci rozhraní
- Proč modelovat komponenty?!?
- Graficky je komponenta znázorněna:



Komponenta

Komponentou může být:

- Tabulka
- Datový i spustitelný soubor
- Dynamická knihovna
- Dokument aj.

Typy komponent

- Rozmíst'ované - (*.dll, prvky ActiveX, spustitelné soubory)
- Podpůrné - (datové soubory, soubory se zdrojovým textem)
- Prováděcí - (vznikají za běhu programu)

Komponenty vs. Třídy

Podobnosti:

- Jména
- Realizují se přes rozhraní
- Možnost dědičnosti, asociace a závislosti
- Mohou mít instance

Komponenty vs. Třídy

Rozdíly:

- Třídy reprezentují logické abstrakce, komponenty fyzické
- Třídy pracují s atributy a operacemi přímo; komponenty mají operace, které jsou dosažitelné pouze přes rozhraní

Komponenty a Rozhraní

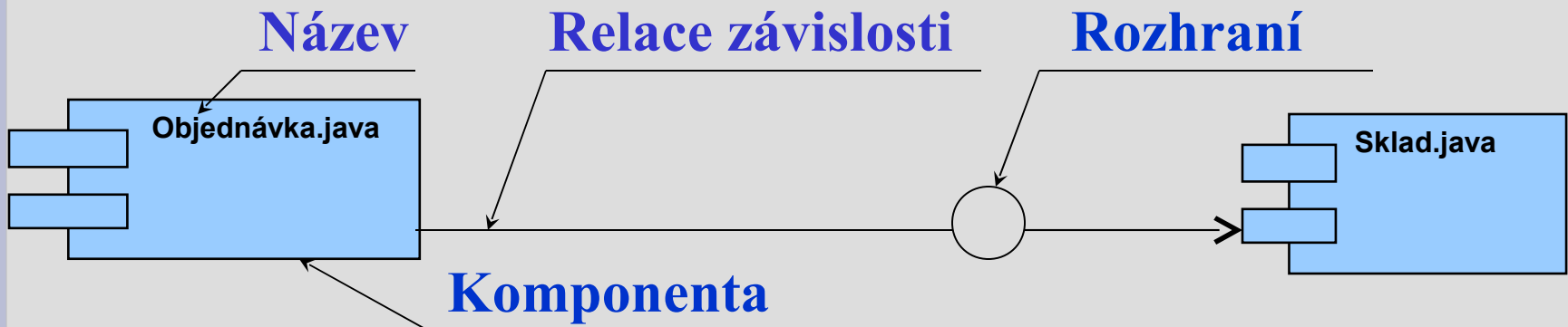
- Rozhraní je sada operací, které komponenta nebo třída nabízí okolnímu světu
- Vztah mezi komponentou a rozhráním se nazývá *realizace*
- Rozhraní, které komponenta poskytuje jako službu jiným komponentám, se nazývá *exportní rozhraní*

Komponenty a Rozhraní

- Rozhraní, které komponenta využívá se nazývá *importní rozhraní*
- Dané rozhraní může být poskytováno (exportováno) jednou komponentou a využíváno (importováno) jinou

Komponenty a Rozhraní

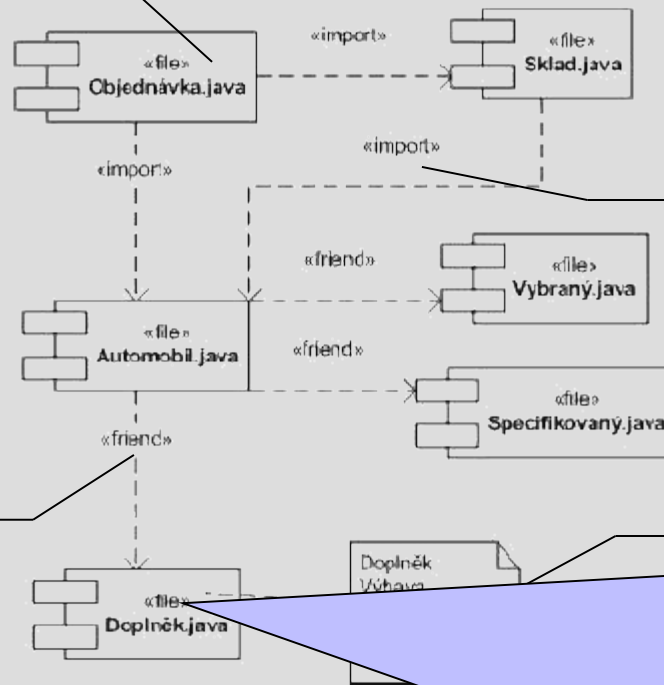
- Ukázka komunikace mezi komponentami:



Příklad prodeje aut

Třídy jsou popsány v
jazyce JAVA
A jsou to soubory

Relace závislosti:
Komponenty ze
stejného balíčku



```
package auta;
public abstract class Doplněk {
    public void přidej(Doplněk d) {}
    public void odeber(Doplněk d) {}
    public Doplněk potomek(int index) {
        return null;
    }
    public abstract float cena();
}

class Klimatizace extends Doplněk {
    public float cena() {
        return 2000.0f;
    }
}

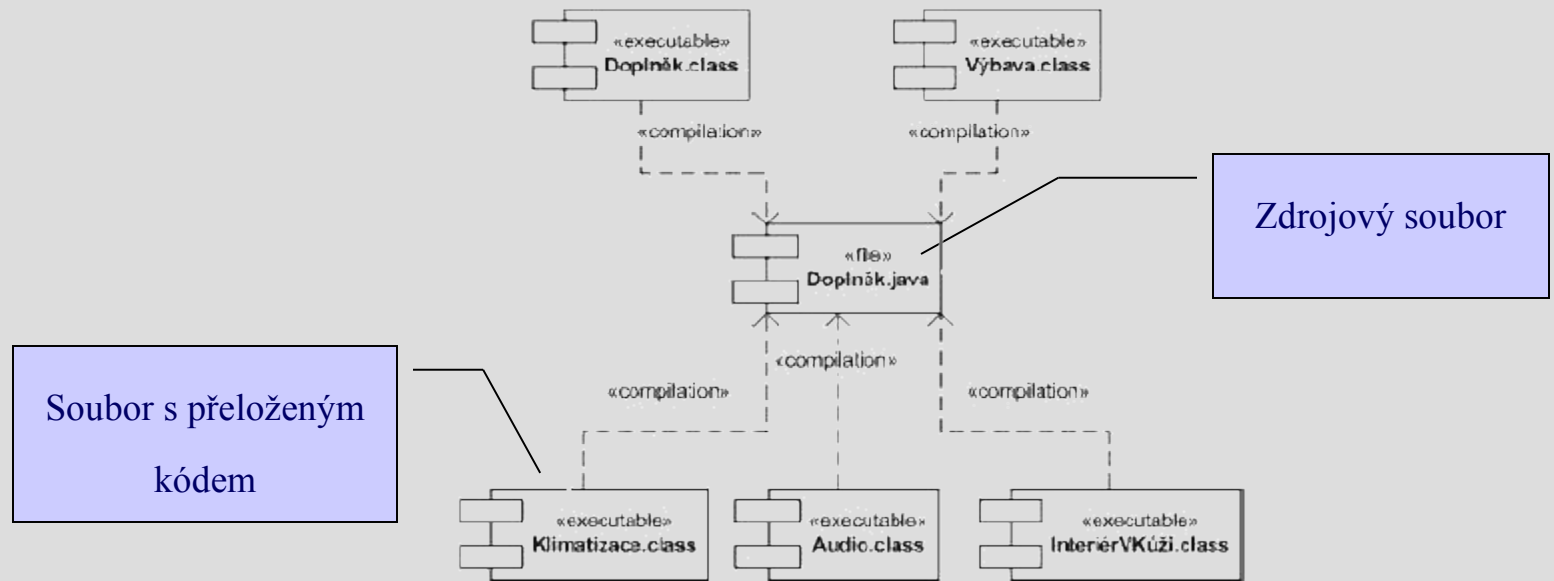
// ostatní třídy
class Výbava extends Doplněk {
    private Doplněk[] doplňky;

    public void přidej(Doplněk d) /* kód */
    public void odeber(Doplněk d) /* kód */
    public Doplněk potomek(int index) {
        return doplňky[index];
    }

    public float cena() {
        float sum=0;
        for (int i=0; i < doplňky.length; i++) {
            sum = sum + doplňky[i].cena();
        }
        return sum;
    }
}
}
```

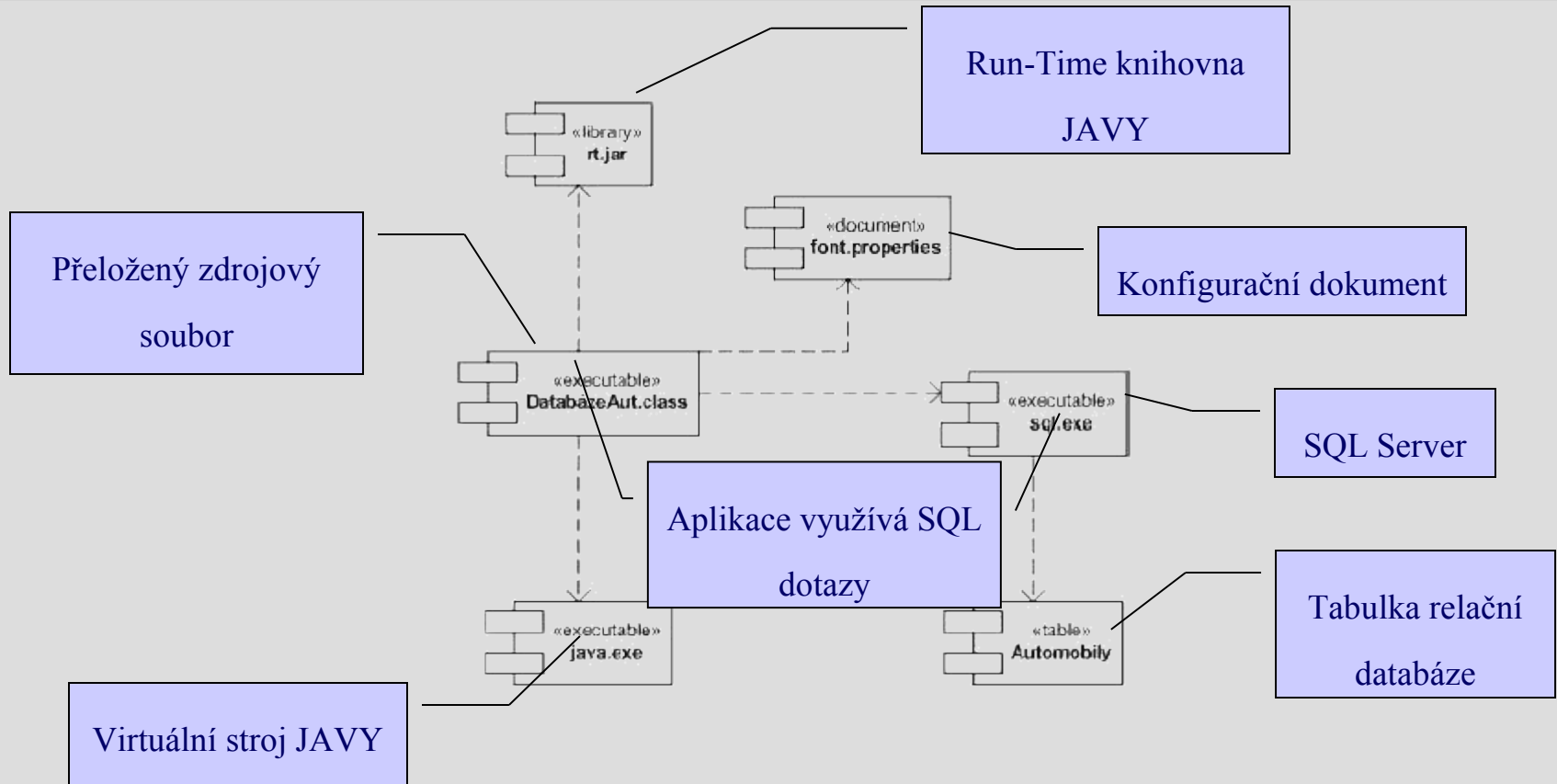
•Obr.1 Zdrojový kód

Příklad prodeje aut



•Obr.2 Binární kód

Příklad prodeje aut



•Obr.3 Běh programu

Použité zdroje

- www.unicorn.cz
- www.uml.org
- <http://www.objects.cz>
- http://vondrak.cs.vsb.cz/download/Uvod_do_softwaroveho_inzenyrstvi.pdf