Adapter

Wrapper,
Adaptér

Mení rozhranie triedy na rozhranie, ktoré požaduje klient. Adaptér umožňuje spoluprácu tried, ktoré by inak kvôli nekompatibilnému rozhraniu spolupracovať nemohli.

Converts the interface of a class into another interface clients expect. Adapter lets classes work together that couldn‘t otherwise because of incompatible interfaces.

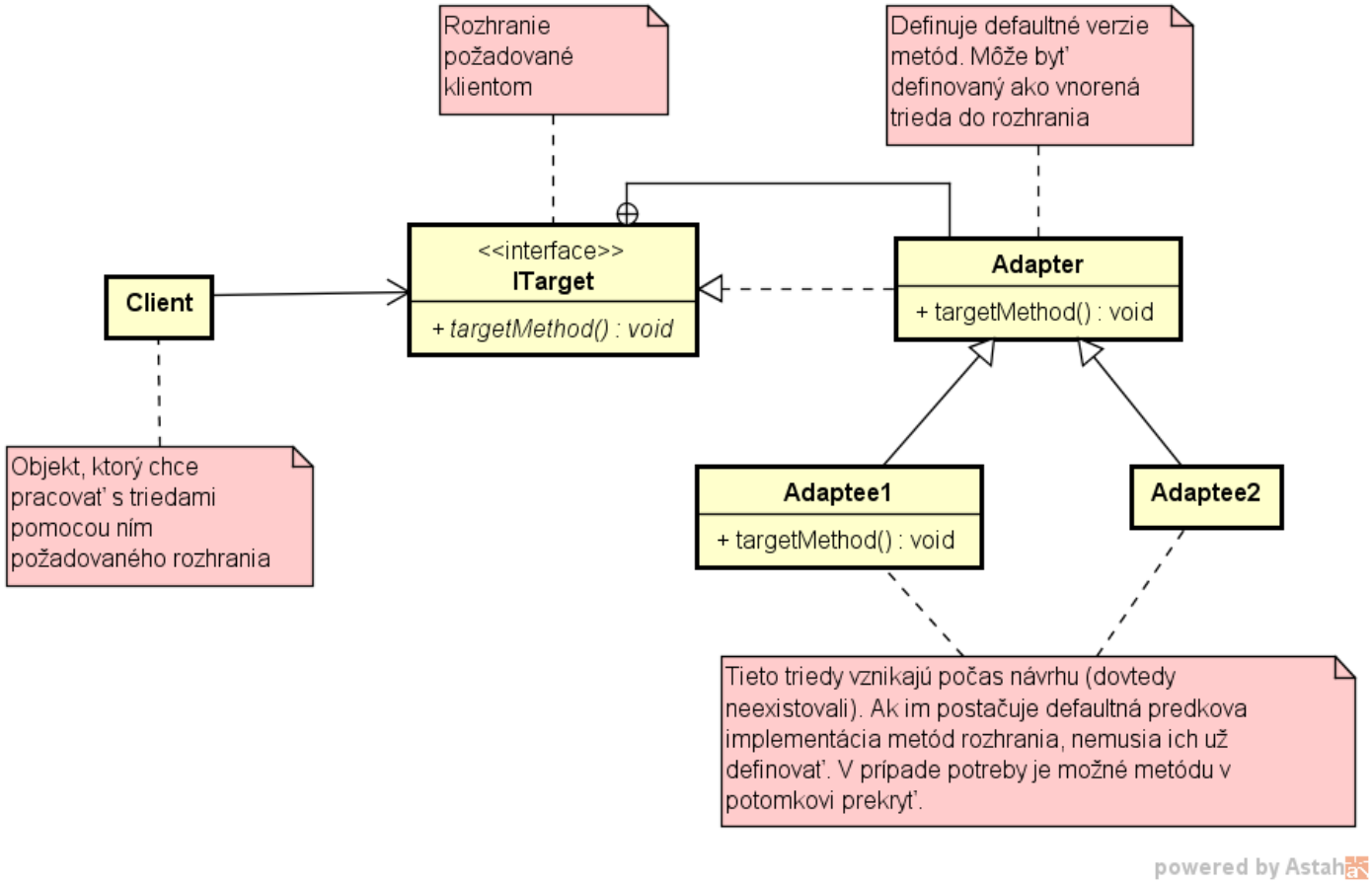
Motivácia

- Majme grafický editor, ktorý umožňuje kresliť primitívne tvary.
- Predkom (rozhraním) všetkých tvarov, ktoré sa kreslia môže byť `IDrawable` (abstraktný predok môže implementovať defaultné verzie metód).
- Je jednoduché kresliť primitívne tvary, problém nastáva pri textoch.
- Existujú knižnice, ktoré však vedia vykresliť text, nemajú však požadované rozhranie (v našom prípade `IDrawable`).
- Ako je možné využiť existujúcu knižnicu a stále pri tom zachovať nami požadované rozhranie?

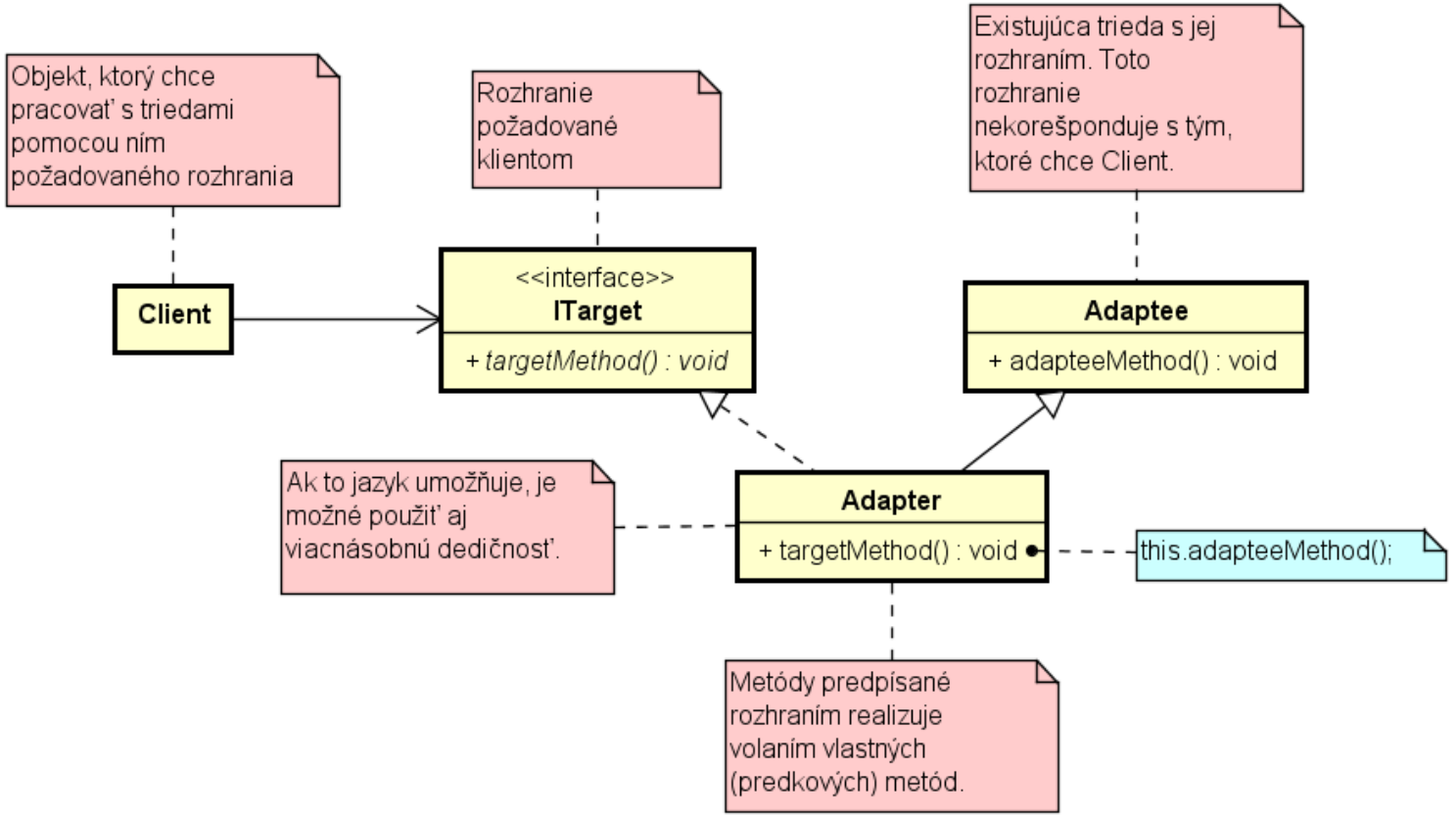
Aplikovateľnosť

- Adaptér je vhodné použiť, keď:
  - chcete použiť existujúcu triedu, ale jej rozhranie nekorešponduje s rozhraním, ktoré potrebujete.
  - chcete vytvoriť znovupoužiteľnú triedu, ktorá bude schopná kooperovať s triedami, ktoré s ňou nie sú priamo spojené (teda nemusia mať kompatibilné rozhrania).
  - potrebujete použiť niekoľko existujúcich potomkov, ale je nepraktické upravovať (adaptovať) ich rozhrania po jednom – dedením. V takomto prípade je možné využiť Objektový adaptér, ktorý je schopný adaptovať rodičovskú triedu.

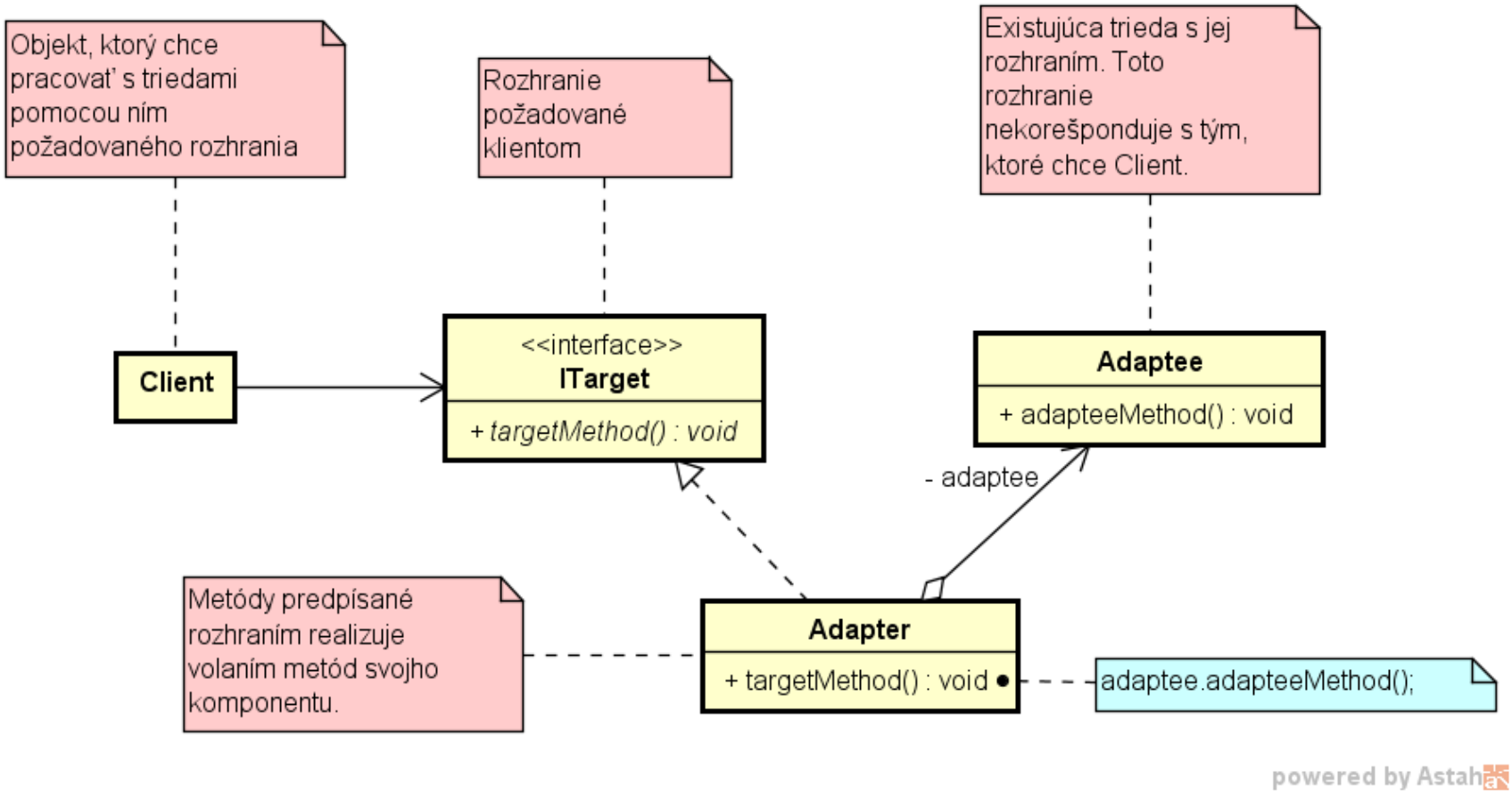
Implementácia 1 – Rodič adaptovanej triedy



Implementácia 2 – Potomok adaptovanej triedy



Implementácia 3 –
Agregácia (kompozícia) s adaptovanou triedou



Dôsledky

- To, aké náročné je vytvoriť adaptér závisí od toho, čo všetko musí adaptér urobiť pre to, aby previedol rozhranie adaptovanej triedy na požadované rozhranie.
- Využitie adaptérov prudko zvyšuje znovupoužiteľnosť tried, nakoľko nie je ich použitie obmedzené iba na triedy, ktoré poznajú presné rozhranie.

Dôsledky – Potomok adaptovanej triedy

- Tým, že je dedičnosťou viazaný na konkrétnu triedu, tak ho nie je možné použiť na inú triedu z hierarchie `Adaptee`.
- Je potomkom triedy `Adaptee`, čo mu umožňuje prekryť rodičovské metódy.
- Ak chce klient získať skutočnú inštanciu `Adaptee`, nemusí ju získavať getterom (ako v prípade agregácie), keďže samotný adaptér je touto triedou.

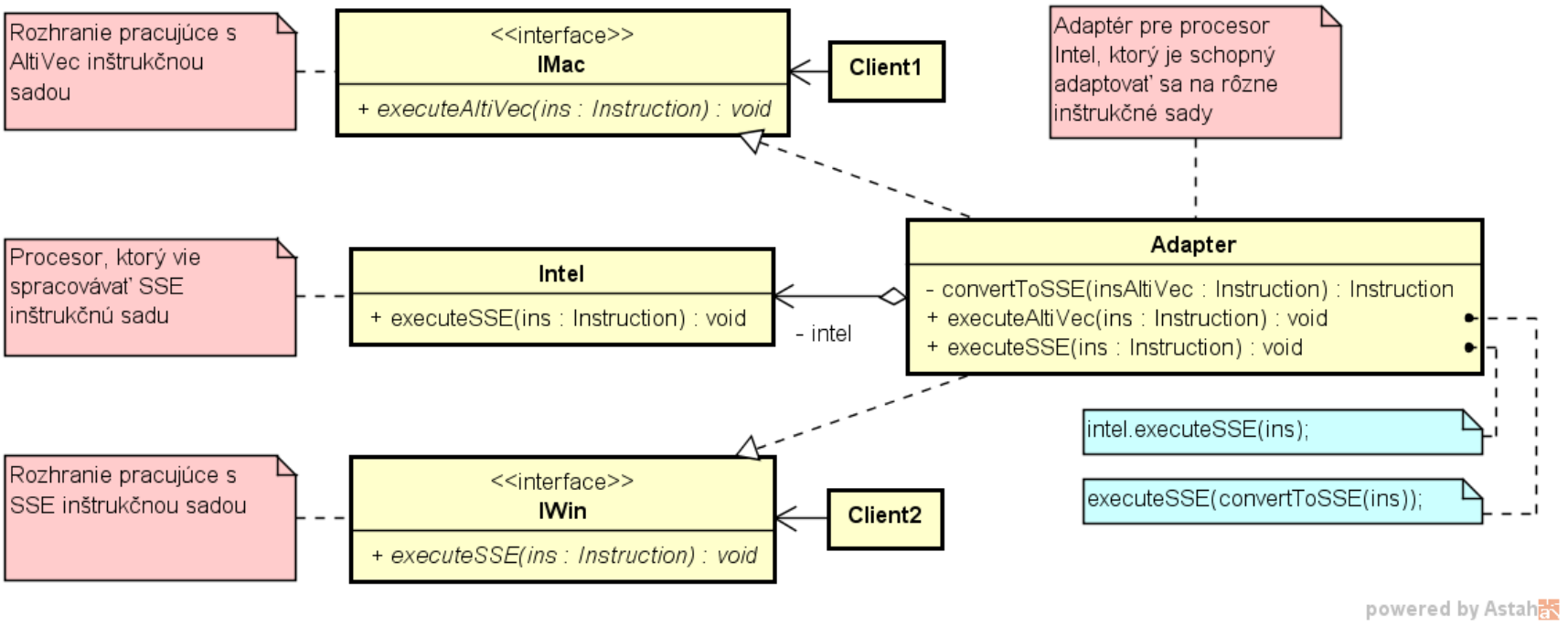
Dôsledky –
Agregácia (kompozícia) s adaptovanou triedou

- Umožňuje využiť jeden adaptér na všetkých potomkov triedy `Adaptee`.
- Je ťažšie dostať sa ku skutočnej implementácii.
- Nie je také jednoduché zmeniť správanie adaptovanej triedy.

Dôsledky – viacsmerný adaptér

- Adaptér nie je adaptovaným objektom, takže sa nemôže používať ako jeho náhrada.
- Toto môže byť obmedzujúce, ak viac, ako jeden klient chce pristupovať k adaptovanému objektu a každý požaduje vlastné rozhranie.
- Riešenie je viacsmerný adaptér, ktorý rieši problém veľmi efektívne pomocou viacnásobnej dedičnosti (resp. realizácie viacerých rozhraní).

Dôsledky – viacsmerný adaptér



Príbuzné vzory

- *Bridge* – Podobá sa na implementáciu Adaptéra s využitím kompozície, ale využíva sa v inej súvislosti:
  - Most oddeľuje abstrakciu a implementáciu, aby ich bolo možné nezávisle meniť, zatiaľ čo
  - Adaptér je určený na zmenu rozhrania existujúceho objektu.
- *Decorator* –vylepšuje funkčnosť iného objektu bez zásahu do jeho rozhrania. Umožňuje rekurzívnu kompozíciu Dekoratérov, čo nie je možné dosiahnuť s Adaptérmi.
- *Proxy* – definuje zástupcu pre iný objekt a nemení jeho rozhranie.

Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu 5II132 Návrhové vzory (Design Patterns) na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené.

Ing. Michal Varga, PhD.
Katedra informatiky
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Michal.Varga@fri.uniza.sk