C# & .NET

http://d3s.mff.cuni.cz

CHARLES UNIVERSITY

Faculty of Mathematics and Physics

Cvičení

RNDr. Filip Krijt

krijt@d3s.mff.cuni.cz

http://d3s.mff.cuni.cz/~krijt/

Plán

- Návrat k testování - TDD
- Rekapitulace předchozí úlohy
- Návrhové vzory
  - Adapter, Proxy, Factory
- Překvapení
- Zadání další úlohy
- Architektonický vzor MVC

Rekapitulace předchozí úlohy

- Dobrá práce, je vidět snaha!

Nevhodné použití

- Pozor na různé typy kolekcí – ty doporučené jsou v System.Collections.**Generic**
  - Pokud používáte kolekci z jiného namespace, měl by pro to být **velmi** dobrý důvod a měli byste kolekci mít dobře prozkoumanou
- Obecně pozor na používání tříd které dobře neznáte – může tam být nějaký předpoklad o kterém nevíte
  - Array.Count() vs Array.Length
    - Obecně věci z System.LINQ
  - SortedDictionary vs. Dictionary – MSDN Library Help

Department of Distributed and Dependable Systems D3S

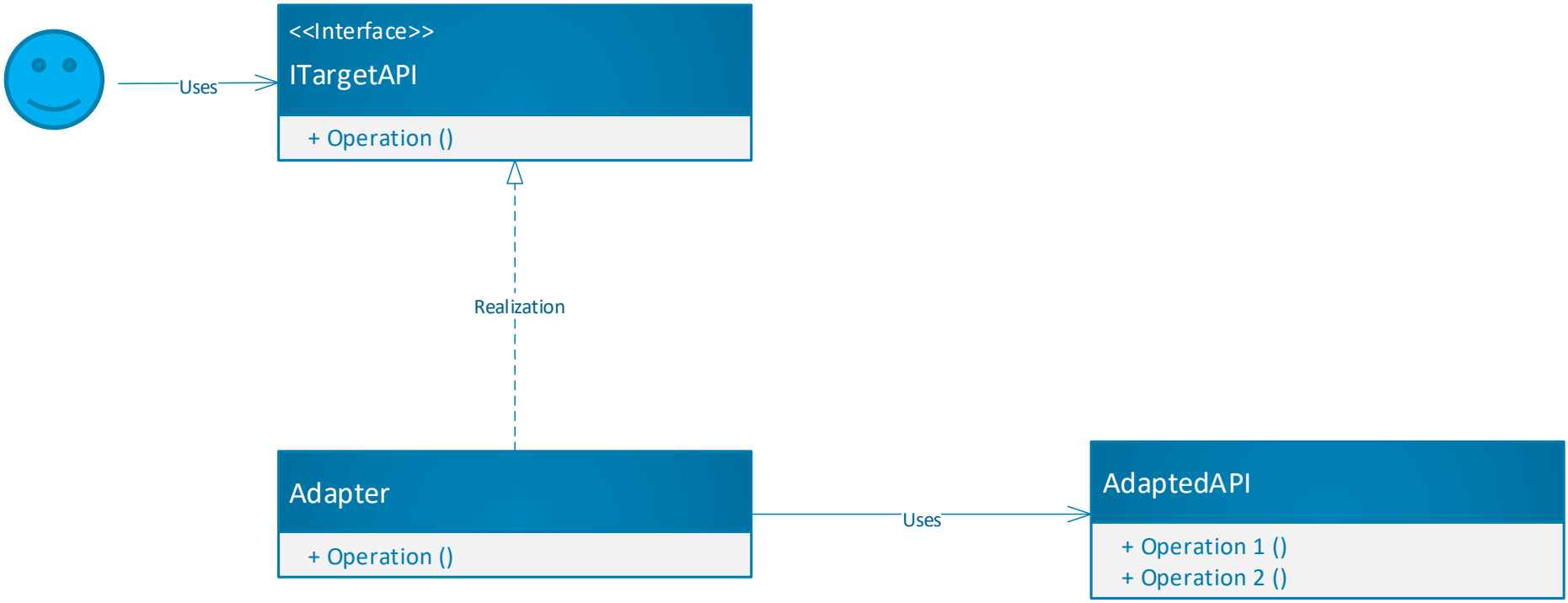
Designování na rozšiřitelnost

Department of Distributed and Dependable Systems
D3S

Návrhové vzory (Design patterns)

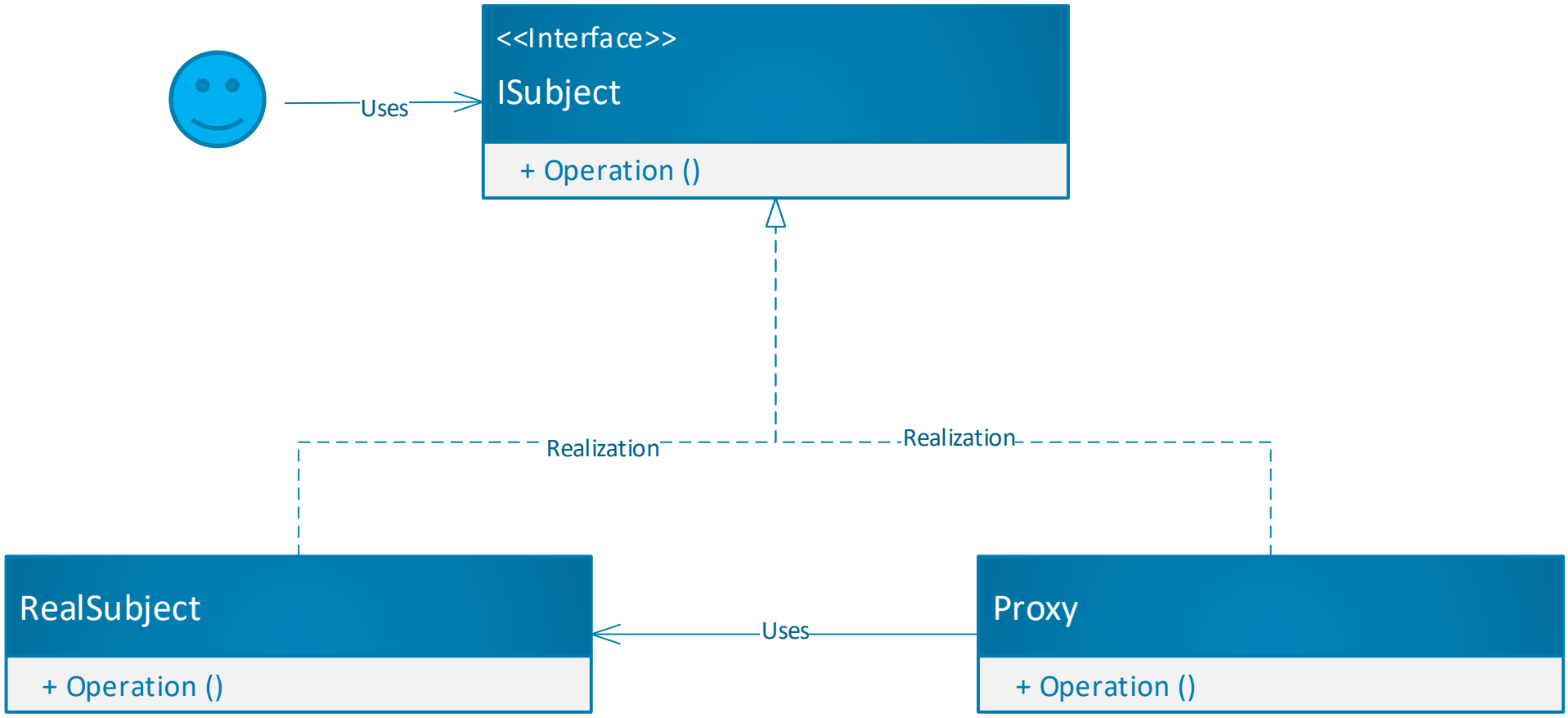
- Pojmenované a popsané řešení nějakého typického problému
- K čemu to je?
  - Komunikace
  - Eliminace zbytečných chyb
  - Menší mentální overhead
  - Nevynalézáme znovu kolo – proč to řešit, když už to někdo vyřešil za nás :)
  - Inspirace
- Nesnažit se pattern napasovat na program za každou cenu!

Adapter



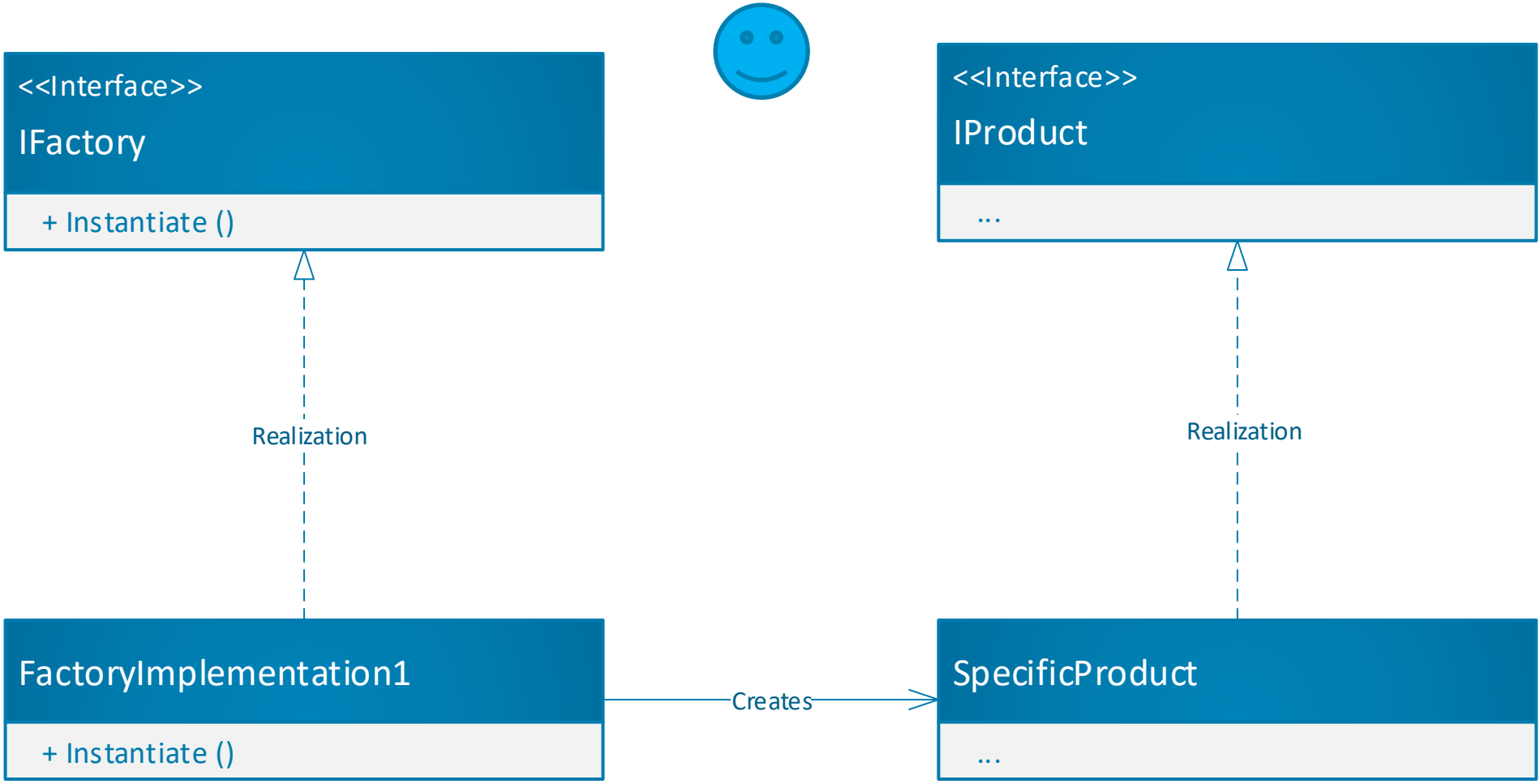
Department of Distributed and Dependable Systems
D3S

Proxy



Department of Distributed and Dependable Systems
D3S

Factory



Department of Distributed and Dependable Systems
D3S

Nežárka.NET

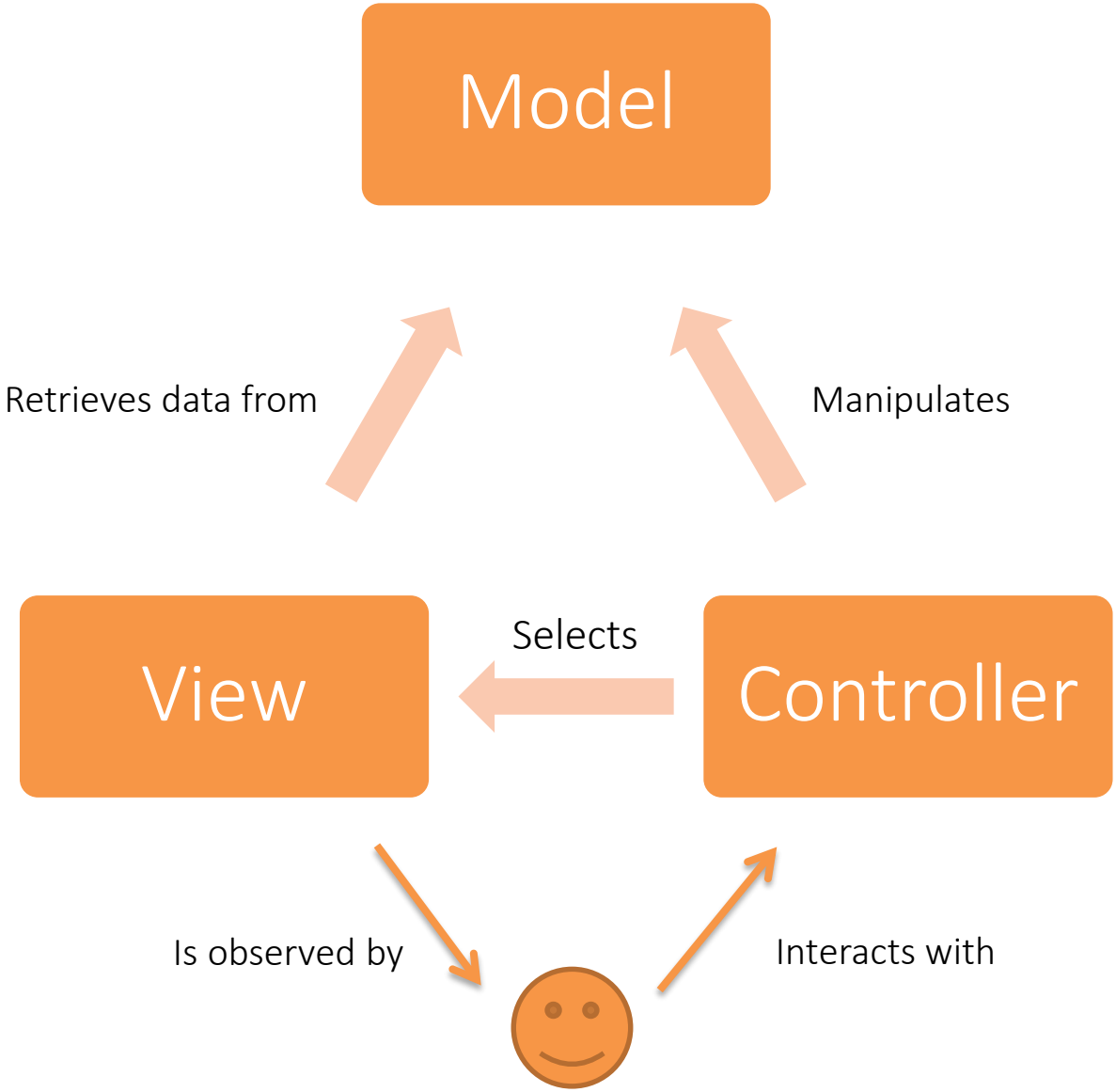
- Konkurence Amazonu a Vltavy :)
- Book Store
- Vstup a výstup přes standardní IO
  - Vstup: Textový popis databáze a sekvence požadavků
  - Výstup: HTML dokumenty odpovídající požadavkům
- Máte k dispozici datové třídy

MVC design pattern

- Architektonický vzor
- Hlavní záměr:
  - Oddělení reprezentace domény od prezentace a řízení
- Použití (v různých variantách)
  - Webové aplikace (ASP.NET i PHP)
  - Složitější desktopové aplikace (typicky v nějaké variantě)
- Tři hlavní součásti
  - Model – Doménová data a jejich reprezentace, případně operace nad nimi
  - View – Vizualizace nějaké části modelu
  - Controller – Řízení (změny modelu, výběr pohledu)

Department of Distributed and Dependable Systems D3S

MVC design pattern



Department of Distributed and Dependable Systems
D3S

MVC design pattern

- Proč?
  - Můžeme mít více různých typů Views (GUI, webová aplikace, konzolové rozhraní…)
  - Můžeme mít více typů modelů (Testovací databáze v paměti, DB, soubor…)
  - Ověřené a používané řešení v mnoha frameworcích (např. ASP.NET MVC)
  - Rozdělení podporuje testovatelnost výsledku
- Simulujeme relativně běžnou praxi – dostali jste doménový Model, máte k němu napsat nějakou aplikaci

Prostor pro dotazy