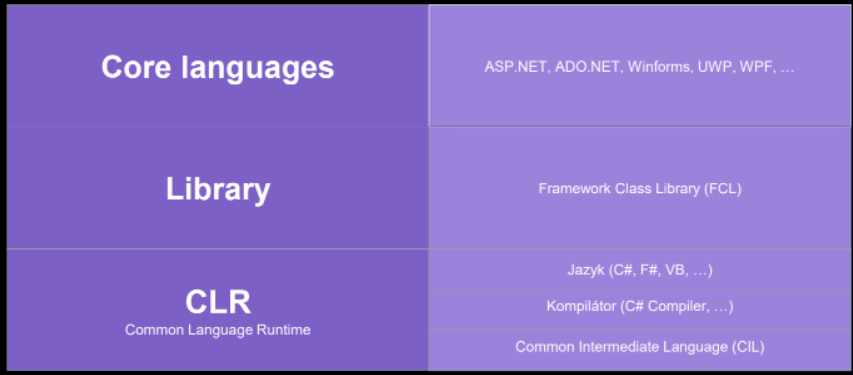
# Obsah obrázku text Popis byl vytvořen automaticky11 Architektury .NET

* Open source framework od Microsoftu
* Cross platform
* 2002 – .NET framework 1.0
* 2019 – .NET framework 4.8
* Nástupcem .NET Core, teď .NET

## Podporované jazyky

* C#
* F#
* C++/CLI
* VB .NET
* …

## .NET architektura



## Prostředky architektury .NET pro cílové platformy

### Common language instrastructure (CLI)

* Standardizovaná specifikace popisující vlastnosti kódu a runtime prostředí
* HLL mohou běžet na více platformách bez nutnosti jejich modifikace
* .NET jazyky jsou sestaveny do platformě nezávislého jazyka Common Intermediate Language (CIL) ~ nejnižší člověkem čitelný kód
* CLR poté převádí CIL do machine-readable kód

### Common Language Runtime (CLR)

* Společné virtuální prostředí pro běh aplikací
* Převádí cil na strojový kód, který je proveden na CPU PC
* Další operace:
  + Správa paměti
  + Type safety (kontrola datových typů, nelze sečíst int a string)
  + Řešení výjimek
  + Garbage collection
  + Vláknová bezpečnost

## Framework Class Library (FCL)

* Organizovaná hierarchie jmenných prostorů
* Poskytuje kolekci tříd, rozhraní a typů (většinou systém. Nebo microsoft.)
* Ty poskytují operace jako čtení/ zápis souborů, lokalizace, komunikace po síti, využití více vláken, reflexe (zjištění informací o objektech za běhu)
* Dále zahrnuje knihovny např pro ASP.NET, WPF, LINQ, ….

## MVVM Model view view-model

* Mimo rodinu .net také znám jako Model view Binder
* Sw architektura, lze považovat za návrhový vzor
* Oddělení gui od logiky aplikace
* **ViewModel** – stará se o převedení dat z modelu do podoby prezentované ve View, používá logiku svou a modelu
  + Nemá odkaz na View, neví o něm (na rozdíl od prezenteru v MVP), objevuje se v něm redundantní kód (boilerplate)
* **View** – neobsahuje žádnou logiku, s VM komunikuje pomocí bindingu, např XAML

## Binding

* Vazba mezi vlastnostmi view a VM
* Data jsou synchronizována
* Poskytovatel při změně upozorní všechny konzumenty ( INotifyPropertyChanged)
* Existuje také obousměrný binding

## Command (příkaz)

* Událost připravena na pozdější použití zapouzdřena do objektu
* Metoda, respektive delegát, který je v commandu uchován, je volán, pokud nastane nějaká událost ve View
* ICommand

Obsah obrázku text

Popis byl vytvořen automaticky

## Observer (pozorovatel)

* Návrhový vzor, závislost 1:0..N mezi poskytovatelem a konzumenty
* Při změně poskytovatele jsou všichni konzumenti upozorněni a mohou na změnu reagovat
* Příkladem tohoto je právě binding
  + Změna vlastnosti upozorní pozorovatele, ten zavolá její getter a podle něj se aktualizují

## Událostmi řízené programování

Obecný pojem pro typ asynchronního programování, který je základem tvorby GUI aplikací

Tok programu je řízen eventy (událostmi), které jsou obvykle spuštěny určitou uživatelskou akcí

**Základní princip** – v programu běží smyčka kontrolující nastání události a volající patřičné metody

