

Dependency Injection

Martin Macák & Jaroslav Davídek

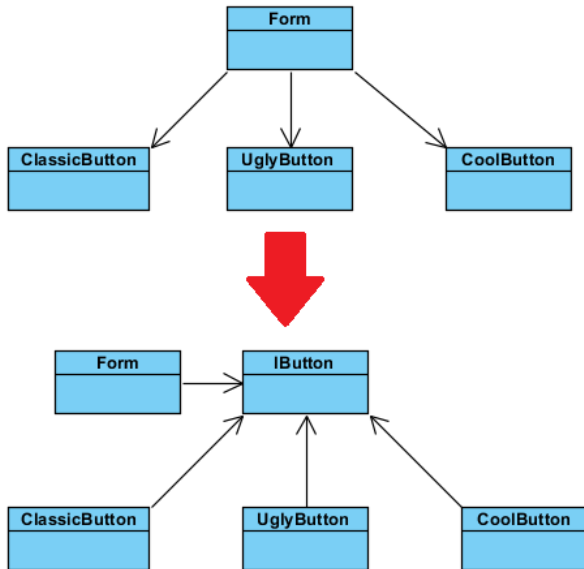
Fakulta informatiky, Masarykova univerzita, Brno

11. 10. 2016

Dependency Inversion

- **Princíp** používaný pri navrhovaní programu
- Vyššie postavené moduly nemajú byť závislé na nižších moduloch
- Treba definovať rozhranie, ktoré budú nižšie moduly spĺňať

Dependency Inversion



Inversion of Control

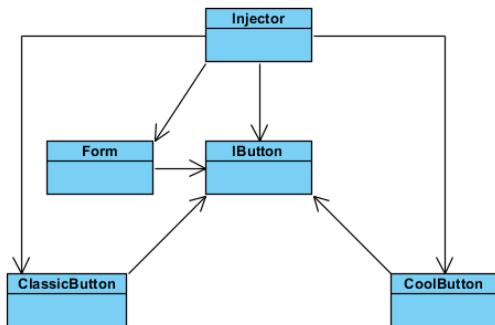
- **"Pattern"**, ktorý sa dá použiť na implementáciu Dependency Inversion
- Rieši **AKO** predať inštanciu nižšieho modulu do vyššieho
- Interface Inversion, Flow Inversion, Creation Inversion

Creation Inversion

- Factory
- Service Locator
- Dependency Injection
- ...

Dependency Injection

- Zoberiem všetky závislosti modulu, vytvorím konkrétne objekty a zviažem ich k inštancii modulu zvonku



Dependency Injection

- Constructor Injection

```
IButton button = new UglyButton();  
var form = new Form(button);  
-----  
public class Form  
{  
    private readonly IButton button;  
  
    public Form(IButton button)  
    {  
        this.button = button;  
    }  
}
```

- Setter Injection

```
IButton button = new UglyButton();  
var form = new Form {Button = button};  
-----  
public class Form  
{  
    public IButton Button { get; set; }  
}
```

- Interface Injection

Dependency Injection Container

- Jednoduchá realizácia DI
- Umožňuje konfigurovať závislosti pre daný modul
- Následne ich dokáže riešiť (rekurzívne) podľa toho, ako je kontajner nakonfigurovaný
- [Castle Windsor](#), Unity, Ninject, ...

Predávam slovo Jardovi :)