



Programování

Testovací analýza a metody návrhu testu

Debugging vs Testing



Debugging

Odstraňuje chyby vzniklé během vývoje

Vyhledávání a analýza hlášených defektů

Obvykle vykonávány vývojářem



Testing

Poukazování na chyby

Reportování a správa seznamu objevených chyb, které jsou opravovány

Typicky prováděné testery



Testovací analýza

- **CO JE TŘEBA TESTOVAT?**
- Testovací podmínka
 - Položka případně událost, která má být ověřena jedním nebo více testy
- Analýza je ovlivněna vhodnou návazností
 - Testovacích podmínek
 - Specifikace
 - Požadavků na funkcionalitu
- Analýza zahrnuje rovněž výběr technik pro návrh testů



Whitebox, Blackbox, Experience-based

➤ **White-box**

- Zakládá se na analýze struktury komponent nebo systému řešení
- „Máme možnost nahlédnout dovnitř“

➤ **Black-box**

- Zakládá se základě testovací dokumentace a specifikací
- „Vidíme pouze vnějšek“

➤ **Experienced-based**

- Jednotlivé testovací scénáře se iterativně vylepšují na základě znalosti chování systému
- „Víme, co by se mohlo rozbít“

Metody a heuristiky pro návrh testů

- Třídy ekvivalence
- Hraniční hodnoty
- Rozhodovací tabulka
- Přechody stavů
- Uživatelský scénář

Třídy ekvivalence

- Vstupy, které nám přichází od uživatele jsou rozdělené do množin
- Očekáváme, že všechny hodnoty v dané množině mají stejné chování
- Žádné dvě množiny nemají společný průnik
- Základní dvě množiny, které můžeme určit jsou validní a nevalidní vstupy
- **Příklad**
- Student může za test získat maximálně 100 bodů, pro úspěšné složení je třeba získat minimálně 65 bodů. Určete třídu ekvivalence pro úspěšné studenty

Hraniční hodnoty

- Hraniční metody „rozšiřují“ třídy ekvivalence
- Hraniční hodnoty hledáme na okrajích množiny
- Sledujeme chování na hranici ekvivalence i v jejím okolí
- Pro vhodný návrh testu je třeba dobrá specifikace
- S využitím této techniky vytváříme více než dvojnásobné množství testů než pro třídy ekvivalence
- **Příklad**
- Lednice má dvě světla indikující optimální teplotu. Optimální teplota je mezi 3 a 8 °C. Pokud je v lednici optimální teplota svítí zeleně, jinak svítí červeně. Určete hraniční hodnoty pro uvedený příklad.



Rozhodovací tabulka

- Požadavek obsahuje logické podmínky k řešení
- Používá se k určení komplexních pravidel pro daný problém
- Cíle je vytvořit všechny možnosti na základě pravdivostních hodnot
- Každý sloupec představuje jedno pravidlo
- Pokrytí testy odpovídá alespoň jednomu testu na sloupec
- Vhodné pro případy s velkým počtem vstupů



Přechody stavů

- Od aplikace se očekává odlišné chování na základě aktuálního nebo předchozího stavu aplikace
- Přechod stavů je často zobrazován jako diagram
- Popisuje jaké možné stavy mohou nastat
- Jaké jsou povolené přechody mezi stavy
- Stavy musí být oddělitelné, definovatelné a jejich počet musí být konečný
- Při návrhu testu je třeba pokrýt všechny stavy a všechny přechody
- Stejně tak ověřit zda nelze provést nevalidní přechod mezi stavy



Uživatelský scénář

- Návrh testu je definován dle případu užití (use case)
- Use case = interakci mezi aktory (typicky aplikace a uživatel), která produkuje výsledek jednomu z aktorů
- Use case popisuje chování aplikace na nejvyšší úrovni
- Popisujeme stav před provedením a po provedení testu
- **Gherkin syntaxe**
 - Given
 - When
 - Then

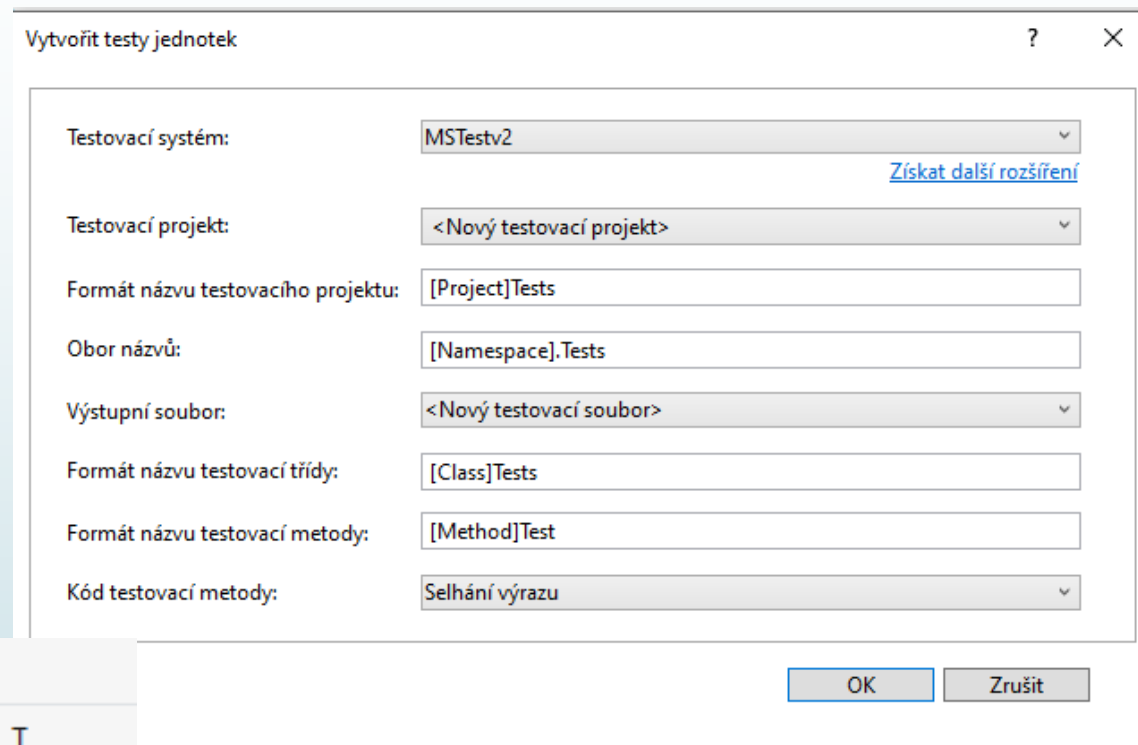
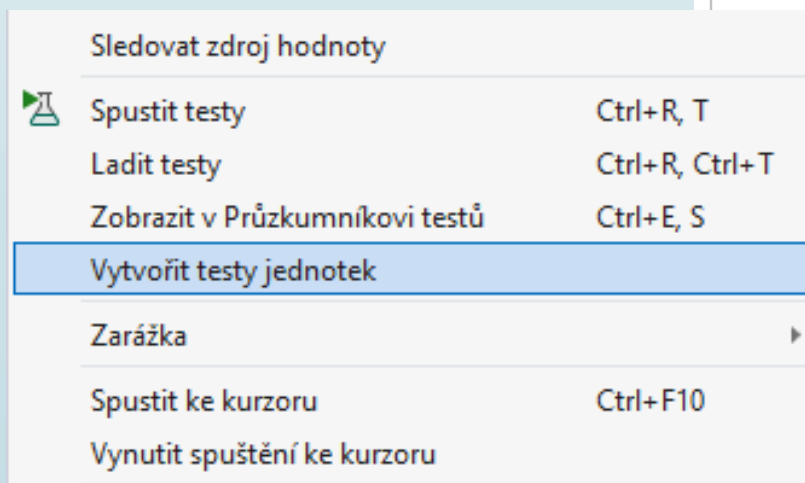


Unit test v C#

- Pro tvorbu unit testů nabízí Visual Studio dvě možnosti
- NUnit Framework
- **MSTest**
- **Unit testy lze vytvářet pouze pro veřejné třídy**
 - Ve výchozím nastavení se přidávání třídy označují jako **internal**
- Projekt testů přidáváme jako řešení k aktuálnímu projektu

Unit test v C#

- Pro vytvoření testu dané funkce stačí funkci označit a přes kontextové menu vybrat možnost **Vytvořit testy jednotek**
- V následném dialogu vyplníme potřebné údaje a do projektu se projekt testů vloží automaticky
- Testy lze pak pouštět samostatně



The background of the image is a stylized world map divided into four quadrants by a vertical and a horizontal line. The top-left quadrant is red, the top-right is blue, the bottom-left is yellow, and the bottom-right is green. The word "Kahoot!" is written in a large, white, rounded, sans-serif font across the center of the image, spanning across all four quadrants.

Kahoot!