

IW5 Fáza 1 – časté chyby

Toto sú najčastejšie chyby, ktoré sme pozorovali pri opravovaní.

Ich poradiu neprikladajte žiadnu dôležitosť, nie sú radené podľa závažnosti.

V maili s hodnotením máte vypísané, ktoré z týchto chýb sa na vás vzťahujú, prejdite si ich ale všetky, je možné, že nám pri opravovaní niečo uniklo, takže ak nájdete niektorú chybu, ktorú máte v programe, ale nie v hodnotení tak sa nás buď na ňu opýtajte, alebo ju rovno opravte aby ste pri ďalšom odovzdávaní zbytočne nestratili body ak si ju všimneme v 2. fázi projektu.

1. Nepoužitá/nesprávne použitá dedičnosť

Dedičnosť nebola použitá vôbec (implementácia rozhrania nie je to isté ako dedičnosť).

Prípadne ak dedičnosť použitá bola, tak od základnej triedy dedila len 1 trieda (napr. Ingredi ence) ale ostatné nie (aj recepty potrebujú svoje Id...).

Ak implementujete property Id a Name v 3 triedach a v každej to píšete znovu tak presne na toto je vhodná rodičovská trieda.

Stačí sa pozrieť do cvičení, viacero tímov napadlo proste zobrať BaseMode l a rozhrania z cvičení a použiť ich. To nám stačilo. Niektoré tímy ešte pridali zložitejšiu dedičnosť a nejaké ďalšie rozhrania, ktoré boli vhodné.

2. Riešenie množstva ingrediencií

Vytvoriť triedu Ingrediente, ktorá má v sebe property množstvo nie je správne riešenie. Zamyslite sa nad tým, čo sa s ingredienciami stane v databázi. Budete mať tabuľku Ingrediente a v nej záznamy:

Id	Name	Amount
1	Vejce	2
2	Vejce	5
3	Vejce	4

Množstvo ingredientie by malo byť riešené iným spôsobom – pomocou ďalšej triedy, ktorá obsahuje property Ingredient a Amount.

3. Decorator implementovaný dedičnosťou

Niektoré tímy sa rozhodli, že by bolo vhodné použiť Decorator. Ale implementácia pomocou dedičnosti nie je práve šťastná. Pozrite sa na vozorvú implementáciu:

<http://www.dofactory.com/net/decorator-design-pattern>

4. Chýbajúce Id

Ak chcete ukladať dáta do databázy, musia mať tabuľky primárny kľúč. Na cvičeniach sme si ukazovali, že sa za týmto účelom používa dátový typ Guid. Databázové tabuľky sa vám generujú zo C# tried, takže aj tie musia obsahovať Id. Pozrite sa do riešenia 2. cvičenia a tam máte pekne ukázané, že triedy, ktoré sa ukladajú do databázy dedia z BaseClass a teda obsahujú Id typu Guid.

5. Vytváranie väzobných tried

Návrh tried nie je návrh databázy. V databáze väzbu medzi tabuľkami riešite tak, že do tabuľky A dáte foreign key tabuľky B a naopak. V triedach ale tá spätná väzba nie je nutná. Keď recept obsahuje nejaké ingrediencie, nemusí aj ingrediencia obsahovať v akom je recepte. Zamyslite sa nad tým, čo vám to vytvorí v databázi. V tabuľke ingrediencií budete mať záznamy ako:

Name	Recipe
Vejce	Míchaná vejce
Vejce	Omeleta

6. TFS – chýbajúce/nič nehovoriace komentáre

Do TFS väčšina tímov nahrávala projekt postupne, po jednotlivých častiach. Ale dosť často sa stávalo, že ste vaše commity neokomentovali. Prehrabovať sa kopou commitov, v ktorých nie je napísané čo vlastne menia, prípadne s komentárom typu „Kopa zmien“ – toto vám moc nepomôže keď sa budete chcieť vrátiť ku staršej verzii kódu.

Pozor – jedná sa o komentáre pri nahrávaní do TFS, nie komentáre v kóde. Pri komentároch v kóde sa riadte podľa 3. cvičenia. Komentáre v kóde sme v tejto fázi ešte nehodnotili (keďže časť tímov mala 3. cvičenie až po odovzdaní 1. fázy projektu).

7. WPF/konzolový projekt namiesto class library

Niektoré tímy založili namiesto class library projektu nejaký iný typ. Najčastejšie to bol WPF projekt, našli sa aj nejaké konzolové. Pozrite sa do štruktúry projektov používanej na cvičeniach.

Modely by mali byť v samostatnom projekte, aby sa tento projekt dal referencovať v iných. Či už tejto štruktúre MVVM projektu – aby sa dali použiť aj vo ViewModeloch aj vo WPF projekte. Alebo ak chcete robiť rôzne typy aplikácií s rovnakou dátovou štruktúrou (konzolová, WPF, mobilná, iné platformy...)

8. Viac tried v jednom súbore, enum v súbore s triedou

Bežné použitie tried a enumov je také, že sa nachádzajú v súbore, ktorý je pomenovaný rovnako ako daná trieda/enum. Za toto sme v 1. fáze nestrhávali body, ale teraz na to upozorňujeme – rozdeľte si triedy tak, aby ich nebolo viac v jednom súbore. Ak máte nejaký súbor, ktorý sa volá inak ako trieda, ktorá v ňom je tak ho premenujte. A presuňte enumy do samostatných súborov. Projekt je potom o dosť prehľadnejší a lepšie sa vám v ňom bude orientovať (nemusíte si pamätať, že enum IngredientType ste nechali v súbore Recipe.cs)

HINT: Kedyž v solution exploreru označíte nejhornější file (Název vašej sln) a zmáčknete alt+R+R, ušetří vám to spoustu práce.

9. Nepreložiteľný projekt, chýbajúci sln súbor, chýbajúce projekty

Niektorým tímom sa podarilo odovzdať nefunkčný projekt. Niekde boli nesprávne premenované triedy, chýbajúci sln súbor, prípadne sln súbor referencoval projekty, ktoré ste už medzi časom stihli zmazať v zložke, ale v solution stále zostali.

Nechovali sme sa tým štýlom, že keď je to nepreložiteľné dostávate 0 bodov. Všetky projekty sa nám podarilo opraviť tak, aby boli preložiteľné a aby sme ich mohli opraviť. Ale ak sme mali problém s preložením projektu tak sme nejaké body strhli.

10. Prístup k property pomocou metód

Ak máte v triede:

```
public property List<Ingredience> Ingredients
```

Tak nepotrebuje metódy `AddIngredience()`, `RemoveIngredience()`, `GetFirstIngredience()` a podobne. `List` má tieto metódy priamo v sebe, takže nemusíte ich obaľovať ešte ďalšou metódou. Potom užívateľ vašej triedy nevie, či keď zavolá `Ingredients.Add()` a `AddIngredience()` stane sa to isté, alebo bude v tom vložení nejaký rozdiel.

Zároveň na priamy prístup k `private fieldom` by ste mali používať property, ktorá v getteri daný field vráti. Nie to obaľovať metódou, ktorá nerobí nič iné, ako zavolať funkciu daného fieldu.

Ak chcete aby sa field nedal prepísať tak sú na to iné mechanizmy (`const`, `readonly`, `ReadOnlyList...`).

11. Česko-anglické pomenovanie

Preferovali by sme, keby ste pomenovávali triedy, premenné a v podstate všetko v projekte anglicky. Ak pridete do nejakej firmy tak to po vás pravdepodobne bude požadované. Ale ak nechcete, pomenovávajú to aspoň česky. Mať celý projekt česky a potom triedu `BaseModel`, `ReceptType`, `GetRecept()`, prípadne mať triedu `Inkredient` – toto nie sú české slová. Takže buď pomenovávajú anglicky alebo česky (alebo slovensky), je to na vás, ale buďte konzistentní.

Za toto sme v 1. fázi body nestrhávali, ale v ďalších fázách v rámci Clean Code budeme. Takže pozor na to