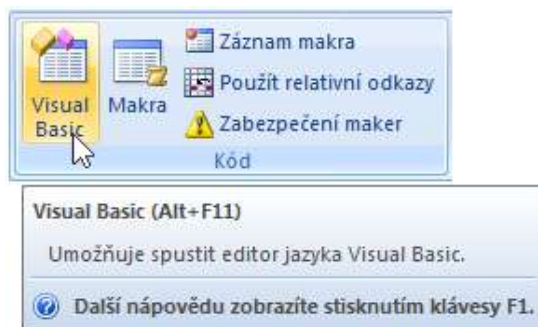


Pokud klepnete na tlačítko Upravit, dostanete se do režimu jazyka Visual Basic pro Aplikace. Zde je potřeba znát alespoň základy tohoto jazyka, aby se dala makro změnit.

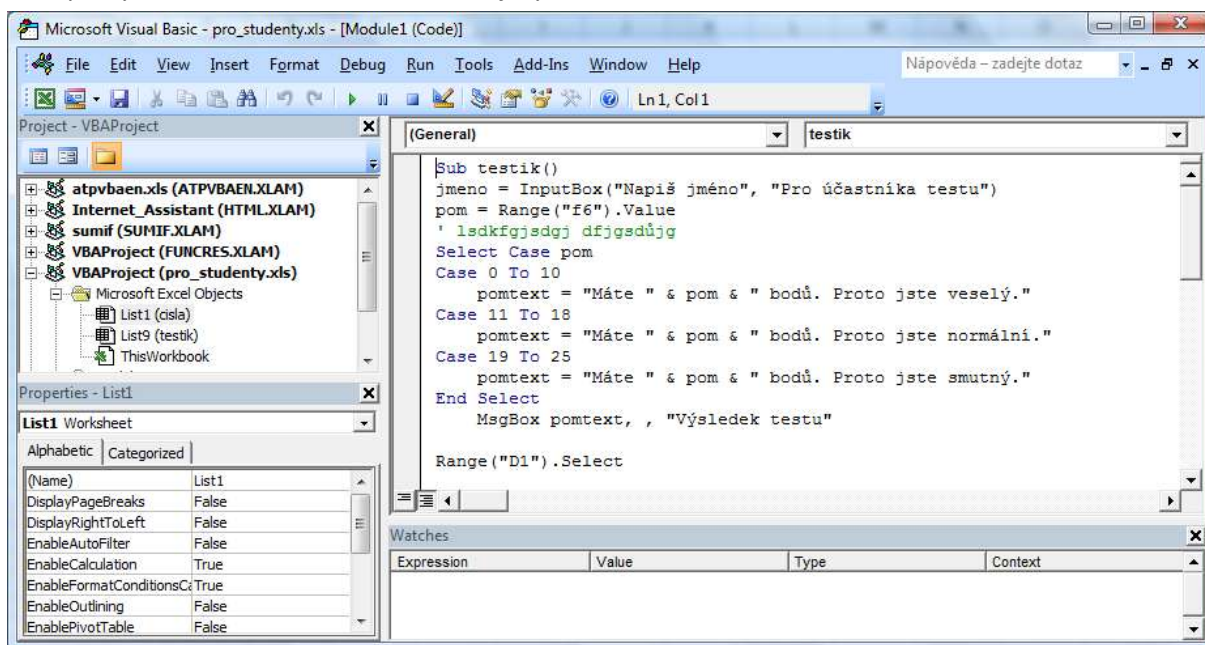
Jinak do režimu pro úpravy maker se můžete dostat i přímo, pokud klepnete na kartě **Vývojář** na ikonu **Visual Basic**. Důležité je také říci, že v jazyce Visual Basic můžete přímo i makra psát.



Poznámka: Makra, která jsou ve skrytém sešitě (osobní sešit maker Personal.xls) se dají upravovat až po zobrazení sešitu.

Popis okna editoru jazyka VBA

Pro úpravy maker se otevře okno editoru jazyka VBA, které se skládá z několika částí.



Části okna editoru VBA

Okno Project Explorer

V okně **Project Explorer** jsou uvedeny všechny projekty (skupiny kódu) a příslušná makra ve stromovém zobrazení, které lze použít podobně jako stromové zobrazení v Průzkumníku Windows.

Okno Properties

Hned pod oknem Project Explorer je zobrazeno okno **Properties**, které slouží k prohlížení a změně různých vlastností souvisejících s vybraným objektem. U modulů je obvykle k dispozici pouze vlastnost Name (Název), u sešitů lze změnit i další vlastnosti, jako StandardWidth (Standardní šířka) a DisplayPageBreaks (Zobrazení zalomení stránek). Tyto vlastnosti mají velký význam především u práce s formuláři a jejich jednotlivými částmi.

Okno Code

Okno **Code** je největším oknem editoru VBA. Vidíte v něm prakticky celý kód zapsaného nebo nahraného makra a také v něm můžete dělat jednotlivé změny nebo úpravy.

Toto okno v horní části obsahuje dva rozevírací seznamy:

- seznam vlevo – **pole Object** – slouží k výběru objektu, se kterým chcete pracovat - pokud používáte pouze kód, zobrazí se v tomto poli výchozí objekt s názvem General (Obecné)
- seznam vpravo – **pole Procedure** (Procedura) – lze vybrat jednotlivá makra v rámci aktuálního modulu

Okno Watches

Toto okno můžete a nemusíte mít zobrazené. Nejčastěji se používá při odladování hotových maker nebo při hledání chyb, protože do něho si můžete zadat proměnné, jejichž hodnoty chcete sledovat.

Co je to VBA?

VBA je objektově orientovaným jazykem, to znamená, že si MS Office můžete představit jako stavebnici s kostkami (objekty), kde každá kostka má své **vlastnosti (Properties)** představované například barvou, tvarem,....

Pak si můžete představit to, že s kostkou lze **udělat určité činnosti aplikovat metody (Method)**, například přesunout, vymazat, ukončit,....

A některé kostky mohou navíc i **rozpoznávat události (Event)**, kdy kód proběhne na základě určité události, například klepnutí myši, změna obsahu, otevření sešitu,....

V rámci VBA se ještě používá pojem **kolekce**. To je skupina objektů stejného typu obsažených v jiném objektu, např. *sešit obsahuje kolekci jednoho nebo více listů*. Má-li se provést tatáž změna ve všech listech sešitu, můžete procházet jednotlivé listy v kolekci a provést změny pomocí kódu programu. K tomu pak slouží smyčka **For...Next** nebo **For Each...Next** (viz dále).

Několik dalších pojmů pro začátek

- **Projekt (Project)** – představuje souhrn všech modulů a formulářů, které patří danému sešitu
- **Objekt (Object)** – nejmenší stavební prvek a může být představován různě, např. buňka, tlačítko, list, graf,...
- **Modul (Module)** – deklarace a procedury s programovým kódem, které tvoří logickou část celku
- **Formulář (UserForm)** – tvoří uživatelské dialogové okno ke komunikaci s uživatelem a je doprovázen programovým kódem vztahujícím se k jednotlivým prvkům formuláře
- **Procedura (Procedure)** – pojmenovaná posloupnost příkazů, která je vykonávána jako určitý celek
- **Programový kód (Code)** – zápis instrukcí podle pravidel VBA
- **Deklarace (Declaration)** – zveřejněná definice proměnných

Co je to procedura

Procedura je vlastně programový kód, který se má vykonat. A jak může procedura vypadat?

Sub *Nazev_procedury(argumenty)*

vlastní kód pro činnost (příkazy VBA)

může se předčasně ukončit, pokud je třeba Exit Sub

End Sub

U procedur pak mohou být deklarované nejenom proměnné, ale i samotné procedury mohou být deklarované jako veřejné nebo jen místní - tedy Private nebo Public:

- **Private** – procedura je dostupná pouze v modulu, ve kterém se nachází, pokud jsou deklarované takto proměnné, pak jsou přístupné jejich hodnoty pouze v dané proceduře.

- **Public** – procedura je dostupná pro celý projekt, standardní deklarace, proto se nemusí uvádět. Proměnné deklarované jako veřejné nemohou být deklarovány v dané proceduře, ale v samostatné části v rámci modulu nebo projektu.

Co je to funkce

Funkce je také programový kód, který něco počítá, ale zároveň vrací výsledek v rámci svého jména. A jak může deklarace funkce vypadat?

Function *Nazev_funkce(argumenty)*

vlastní kód nebo výpočet (příkazy VBA)

Nazev_funkce=výpočet (výsledek)

End Function

Příklad:

```
Function povrch_valce(v, r)
    obvod = 2 * 3.14 * r
    povrch = 2 * 3.14 * r * r + obvod * v
    povrch_valce = povrch
End Function
```

Jak je to s deklarací proměnných?

Jestliže v horní části modulu zapíšete slova **Option Explicit**, pak uděláte to, že vynutíte, že každá proměnná v modulu musí být někde deklarovaná. Jakmile na deklaraci některé proměnné zapomenete, makro nespustíte. Proto se tomuto příkazu říká **Vynucená deklarace**.

Jestliže chcete nějaké proměnné používat ve více procedurách v rámci celého modulu, pak je potřeba tyto proměnné deklarovat ještě před první procedurou modulu.

Příklad:

```
Dim ahoj as Integer
Sub aaa()
End Sub
Sub bbb()
End Sub
```

Proměnné mohou být veřejné nebo statické.

- **Veřejné proměnné:** proměnná bude přístupná ve všech modulech VBA v rámci projektu
Public ahoj as Integer
- **Statické proměnné:** ty, které si uchovávají svoji hodnotu i po skončení procedury, jsou definovány jen na úrovni procedury
Sub aaa()
Static ahoj as Integer
End Sub