

Téma 6: Metody tříd

Metoda je stejně jako třída tvořena hlavičkou a tělem. Hlavička začíná specifikátorem přístupu (public, private,...) za ní následuje návratový typ, což je hodnota, kterou metoda musí vrátet. Pokud metoda nemá vrátet žádnou hodnotu, tak zde zapíšeme klíčové slovo void. Další položkou je název metody a v kulatých závorkách čárkou oddělené parametry volání.

Statické metody jsou součástí třídy, ale lze je použít, aniž by byla vytvářena instance třídy. Chceme-li označit danou metodu jako statickou, použijeme klíčové slovo static.

Finální metody mohou v odvozených třídách překrývat, ale existují případy, kdy chceme mít jistotu, že danou metodu nebude možné změnit. Slouží k tomu konečné metody, které nemohou být „přepsány“ v žádné z odvozených tříd. Chceme-li označit danou metodu jako konečnou, použijeme klíčové slovo final.

Deklarace metody

- Metoda je stejně jako třída tvořena hlavičkou a tělem.
- Hlavička obsahuje specifikátor přístupu (public, private,...)
- Následuje návratový typ, což je hodnota, kterou metoda musí vrátet.
- Pokud metoda nemá vrátet žádnou hodnotu, tak zde zapíšeme klíčové slovo void.
- Další položkou je název metody a v kulatých závorkách čárkou oddělené parametry volání.

Návratový typ je typ hodnoty, kterou vrátí podprogram volajícímu. Pokud budete programovat výše uvedenou metodu na sčítání, bude program vracet součet celých čísel a tudíž int. Metody mohou vracet všechny datové typy a navíc i typ void, který znamená prázdno, metoda nic nevrací. Taková metoda se například využívá k výpisům na obrazovku.

Návrat hodnoty se zapisuje, následovně return hodnota, klíčové slovo return zařizuje vrácení hodnoty, metoda vždy může vrátet jen jednu proměnnou a to stejného typu jako je uveden v deklaraci metody. Pokud metoda nic nevrací, return se neuvádí. Pokud ovšem metodu návratového typu void a chcete ji ukončit předčasně lze použít return; za kterým jak je vidět nic neuvedete.

Seznam parametrů je seznam hodnot, které metoda dostane od svého volajícího. Metoda může převzít libovolný počet parametrů, dokonce to nemusí být ani určeno, stejně tak ale nemusí převzít žádné parametry, pak se uvedou prázdné kulaté závorky. Parametry metody se deklarují v kulaté závorce za jménem metody, uvede se nejdříve typ proměnné a poté její jméno. Pokud metoda přebírá více hodnot, mezi proměnnými se udělá čárka.

Skutečné parametry jsou hodnoty, jež jsou předány metodě při volání, skutečné proto, že přivolání je jasné jejich hodnota.

Formální parametry jsou parametry, jež jsou deklarovány v metodě. Tyto parametry zastupují skutečné hodnoty, které převezmou při volání metody.

Když voláme metodu, které předáváme primitivní datový typ (int, char, boolean), tak ho **předáváme hodnotou**, to znamená, že hodnota proměnné se během vykonávání metody nezmění.

Odkazem se předávají ne primitivní datové typy, jako jsou objekty. Když změním vlastnost tohoto objektu v metodě, změna se v celém programu.

Metody v programu vystupují svojí signaturou (vlastně jméno), které tvoří jméno (seznam parametrů), dle signatury se v programu metody rozeznávají. V třídě nesmí být dvě metody se stejnou signaturou. Pokud ovšem potřebuji sčítat dvě čísla a nejsou jen celá, ale mohou být buď celá, nebo reálná pak musím využít **přetížené metody**.

Přetížená metoda má stejné jméno, ovšem různý seznam parametrů. Pokud pak program pustím, vyhledá se metoda, která svým seznamem parametrů odpovídá a ta se zavolá. Někteří autoři nedoporučují používat přetížené metody, ovšem pokud potřebujete vyvolat metodu pro různé parametry je to nejlepší možný způsob, než vymýšlet pro každou metodu jiný název.