

## 1. Pole objektů

Ve stávajícím solution Csharp\_OOP\_cv vytvořte projekt Rozhrani typu Console App. Vytvořte soubor Rozhrani.cs (namespace změňte z Rozhrani na Rozhrani1). V tomto jednom souboru vytvořte abstraktní třídu **Animal**, která bude mít jen datovou složku **name** a konstruktor s parametrem typu **name**. Dale podřízené třídy **Dog**, **Cat**, **Turtle**, každá z nich bude mít navíc nějakou datovou složku, například **Boolean isPedigree** (s rodokmenem) u psa i kočky, **int speed** u želvy. Tyto dodatečné datové složky nebudeme využívat (jsou jen pro větší podobnost s reálnými programy, a taky, aby se třídy trochu lišily), proto konstruktory těchto tří tříd budou mít jen parametr name, budou volat konstruktor třídy Animal. Dále bude pes a kočka mít metodu **sound()**, která bude vracet řetězec („haf“ a „mňau“). Na konec souboru napište třídu **Rozhrani**, kde bude jen metoda **Mainx**. V ní vytvořte několik instancí daných tříd, pak vytvořte pole nazvané **animals** a instance do pole vložte. Protože pole musí být homogenního typu, musí se jednat o pole typu, který je všem společný, tedy typu **Animal**. Potom ve smyčce vypište všechna jejich jména.

**Tento první krok projektu je velmi podobný etapě 5 v projektu První. Akorát není tříúrovňová dědičnost a metoda (zde sound) není v báze třídy, protože želva by ji nemohla dědit. A také si povšimněte, že třídy pes, kocka a zelva jsou až na pár písmenek totožné, takže poté co vytvoříte první z nich, tak ostatní zkopírujete.**

Druhý krok: aby byl výpis přehledný, tak u každého jména uveďte datový typ, tedy zda se jedná o psa, kočku či želvu.

Použijeme k tomu jednu ze čtyř metod zděděných od třídy Object (už jsme pracovali s metodou ToString), a to GetType(). Metoda však vrací objekt. Abychom dostali čitelný výsledek, musíme vypsát jen datovou složku Name, tedy dohromady GetType().Name.

**1b.** V druhé fázi pole zdokonalte tak, aby si program zjistil počet vytvořených instancí a podle toho si nastavil rozměr pole. K tomu ale potřebujeme mít datovou složku **countOfAnimals**, kterou budeme inkrementovat při vzniku každého zvířete.

Jakého typu tato datová složka bude? A kde se bude inkrementovat?

Druhý krok: vytvořené instance vložte do pole při jeho deklaraci, budou ve složené závorce. v tom případě nepotřebujeme datovou složku countOfAnimals – Inicializátor pole/kolekce

Třetí krok: Nevyužijeme již vytvořené instance, vytvoříme je (pomocí new) až v {} při deklaraci pole (předchozí způsob deklarace pole zakomentujeme)

2. Předělejte na foreach (nebo for, podle toho, co jste použili výše), místo pole použijte kolekci.

3. A nyní modifikace: Kromě třídy a jména máme vypsát ve smyčce i zvuk (sound), který zvířata vydávají. Pokud pouze doplníte ve smyčce volání metody **sound()**, tak program nepůjde skompilovat, protože třída **Animal** tuto metodu nemá. A nemůžeme ji do této třídy doplnit, protože třída **Turtle** ji nemůže dědit. Takže tudy cesta nepovede, ve smyčce nám zvuk vypsát nepůjde.

2. krok: zkuste tedy aspoň v metodě Mainx vypsát zvuk pro konkrétní buňku pole, o které víte, že je v ní např. Pes. Zkuste přetypování a potom operátor as

3. krok: zkuste výše uvedené (přetypování a as) ve smyčce, ale jen pro psa. Tedy před WriteLine musíte doplnit if. Testujte pomocí operátoru is. Pak zkuste i testování pomocí GetType

4. krok: upravit, aby se metoda sound spouštěla i pro kočky: nejprve pomocí elseif a potom pomocí if plus ternárním výrazem

