

Jméno:..... Datum:..... Body:

Login:

--	--	--	--	--	--	--

Login pište velkými písmeny a ČITELNĚ.

Forma studia:

prezenční

PJ1 – programovací test 2

Procvičovací varianta

- 1.) Vytvořte generický interface **IIentifier**. Interface bude mít jednu fci `getID()`, která bude vracet generický typ.
- 2.) Vytvořte třídu **Animal**, která bude implementovat rozhraní **IIentifier<String>**. Třída **Animal** bude obsahovat:

Atributy:

- `Name(String)`
- `Species(String)`
- `LifeTime(int)`

Konstruktor:

- `Animal(String Name, String Species, int LifeTime)`

Metody:

- `getLifeTime`
- `getSpecies`

implementaci metody `getID`, která vrátí řetězec tvaru „rok/species-name“, kde species a name jsou příslušné atributy a rok je aktuální zkrácený rok (např. 09 pro 2009)

Přetíženou metodu `toString()`, která vrátí řetězec „Animal [Name:name, Species:species, Lifetime:lifetime]“

- 3.) Vytvořte generický interface **IGroup**. Omezte použití generického typu pouze pro interface **IIentifier** (`<T extends IIentifier<?>>`). Interface bude mít tyto fce:

- `additem(T a)` throws `AlreadyContainedException`
- `removedItem (Object id)` – vrací boolean
- `findMax()` – bude vracet hodnotu generického datového typu

- 4.) Vytvořte třídu **MyZoo**, která implementuje rozhraní **IGroup<Animal>**:

- pro ukládání zvířat použijte kontejner `Map<String, Animal>` a jako id použijte hodnotu, kterou vrací metoda `getID` u zvířete.
- Metoda `auditem` vloží zvíře, pokud tam zvíře se stejným id neexistuje, jinak vyhodí výjimku `AlreadyConteinedException`.
- Metoda `removeltem` smaže zvíře s daným id, pokud v kolekci toto zvíře je vrátí `true`, jinak vrátí `false`.
- Metoda `findMax` vrátí zvíře v zoo s nejdelší délkou života.