

# Bárdosi Bence

## Dokumentáció a 3. OAF beadandóhoz

### Feladat

Készítsünk C++ programot a következő feladat megoldására! A hobbi állatoknak az életkedvük megőrzéséhez a táplálékon túl egyéb dolgokra is szükségük van: a halaknak oxigén dús, megfelelő hőmérsékletű vízre; a madaraknak tágas, tiszta kalitkára; a kutyáknak rendszeres foglalkoztatásra. Pisti számos hobbi állatot tart: halakat, madarakat és kutyákat. Állatainak van neve és ismerhető az életkedvüket mutató 0 és 100 között szám (0 esetén az állat elpusztul). Pistinek vannak jobb és rosszabb napjai. Mikor nagyon jó kedvű, egyik állatáról sem feledkezik meg: ilyenkor a halak életkedve 1- gyel, a madaraké 2-vel, a kutyáké 3-mal nő. Átlagos napokon csak a kutyáival foglalkozik, a többi állat életkedve ilyenkor csökken: a halaké 3-mal, a madaraké 1-gyel. Amikor rosszkedvű, csak a legszükségesebb teendőket látja el és ezért minden állat egy kicsit szomorúbb lesz: a halak 5 egységgel, a madarak 3-mal, a kutyák 10-zel.

Az állatok adatait egy szöveges állományban találjuk. Az első sor tartalmazza az állatok számát, amelyet külön-külön sorban az állatok adatai követnek. Ebben egy karakter azonosítja az állat fajtáját (H – hal, M – madár, K – kutya), amit szóköz után az állat neve követ, majd újabb szóköz után a kezdeti életkedve. Az állományban az állatok felsorolását követő utolsó sorban egy betű sorozat (sztring) írja le Pisti kedvének az egymás utáni napokon való alakulása: j – jó kedvű, a – átlagos, r – rosszkedvű. Feltehetjük, hogy a fájl formátuma helyes.

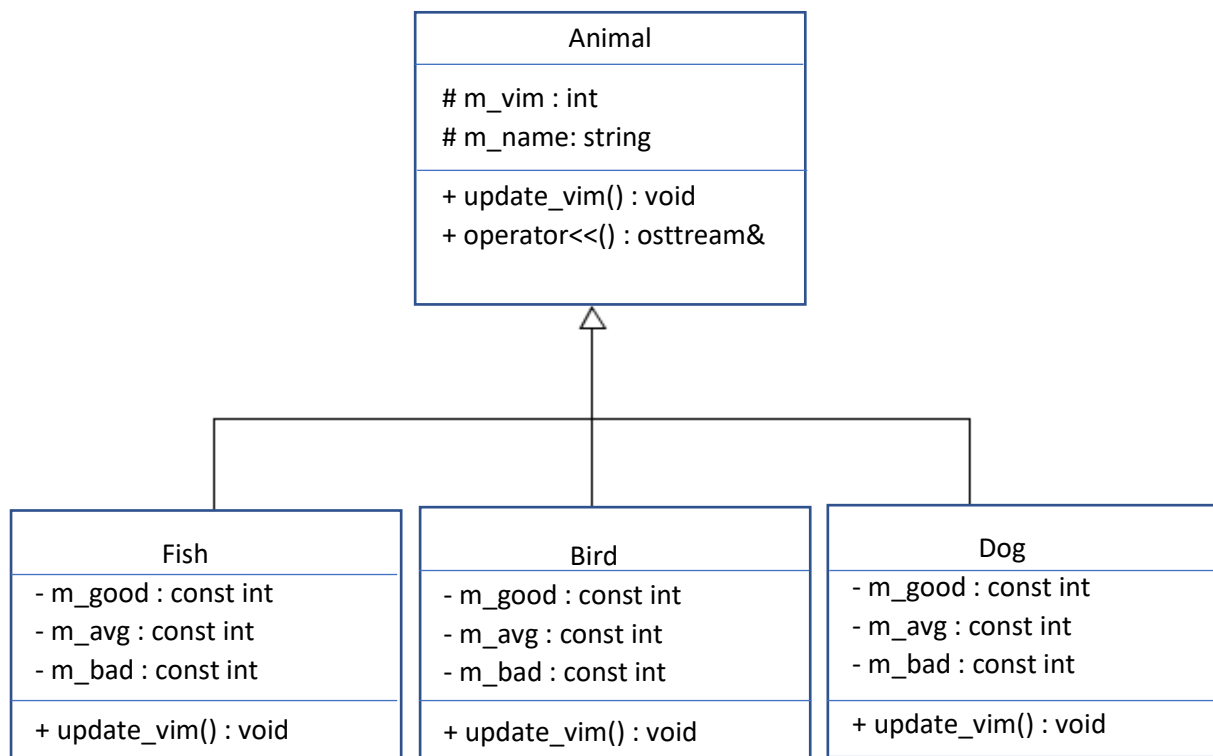
***Szimuláljuk az állatok életkedvének változását Pisti kedvének alakulása során és írja ki az állatok adatait minden nap végén!***

# Bárdosi Bence

## Dokumentáció a 3. OAF beadandóhoz

### Specifikáció

A házi kedvencek leírásához bevezetünk négy osztályt: a kedvencek általános tulajdonságait megadó ősz osztályt (Animal), és ebből származtatjuk a konkrét fajú állatokat, a halak (Fish), a Madarak (Bird) és a Kutyák (Dog) osztályait. Attól függetlenül, hogy egy állat konkrétan kicsoda vagy mi a fajtája, számos közös tulajdonsággal rendelkezik. Mindegyiknek van neve (`m_name`) és életerejéje (`m_vim`) az `operator<<` segítségével egy outstreambe lehet írni az állat adott állapotát (név életerő pár), szimulálni lehet egy napot attól függően, hogy milyen napja volt Pistinek. Ez utóbbi művelet (`update_vim()`) módosítja az állat életerejét. Az `operator<<` metódus már az ősz osztály szintjén implementálható, de az `update_vim()` csak a konkrét osztályok szintjén. Ennek hatása ugyanis attól függ, hogy a kedvenc milyen fajú. Ennél fogva az általános házi kedvenc típusát leíró osztály absztrakt lesz, hiszen a `update_vim()` metódus absztrakt, másrészt úgyszemint akarunk ilyen objektumot létrehozni.



A speciális osztályok konstruktorai meghívják az ősz osztály konstruktorát, majd inicializálják az életerőt, és felüldefiniáljuk a `update_vim()` metódust.

## Bárdosi Bence

### Dokumentáció a 3. OAF beadandóhoz

Ezen felül szükségünk lesz még egy osztályra, amivel a házi kedvencek életerejét szimuláljuk Peti hangulatváltozásai alapján (Simulator). Az osztály privát részében egy `Animal` típusú objektumokra mutató pointereket tartalmazó tömb (`m_animals`), valamint egy `fstream` típusú változó (`m_file`) található. Az előbbi Pisti házi állatait, az utóbbi a bemeneti fájlt. Az osztály publikus részében két metódus található (`run()`, `create()`). A `run()` szimulálja az állatok életerejét Pisti hangulatváltozása szerint, valamint ezt ki is írja a standard outputra. A beolvasás a `create()` függvényben történik.

### Megoldás C++-ban

Az `Animal` osztály, valamint összes alosztályának a deklarációját a `Animal.h` fejlécállományban találjuk, metódusaikat a `Animal.cpp` forrásállományban implementáljuk. A `Simulator` osztályt a `Simulator.h` és a `Simulator.cpp` fájlokban találjuk. Az osztály publikus része a típusspecifikációban felsorolt műveleteken kívül a konstruktort és a destruktort tartalmazza.

Bárdosi Bence  
Dokumentáció a 3. OAF beadandóhoz

## **Tesztelési terv**

### *Szürke doboz tesztesetek*

#### *Állatok száma szerint*

1. Nulla állat
2. Egy állat
3. Több állat
  - a. különbözőek
  - b. ugyan olyanok típusúak