

Készíts egy `Kutya` nevű osztályt.

1. Egy kutya objektumról az alábbi attribútumokat szeretnénk tárolni: név, fajta, kora. Az attribútumok legyenek `private` (csak az osztály számára) láthatóságúak.
2. Írj konstruktort a `Kutya` osztálynak, amely a paraméterek alapján inicializálja az attribútumokat (`this...`).
3. Írj egy véletlenszámot előállító függvényt, amit a leszármazott osztályok érnek el. A fgv. 2 paramétert várjon [ah, fh] és itt generáljon egy egészszámat!
`protected int rnd(int ah, int fh)`
4. Írj publikus függvényeket, amelyekkel lekérdezhetőek az attribútumok. (Getter-ek)
5. Mivel minden kutyának speciális szőre van, ezét szabályként írd meg egy `abstract` metódust `getSzore()` névvel, amelyik mindegyik kutyának a szőrzetét éri el.
6. Mivel minden kutyának speciális tulajdonságai is vannak, ezét szabályként írd meg egy `abstract` metódust `getTulajdonsag()` névvel, amelyik mindegyik kutyának a tulajdonságát éri el.
7. Definiáld felül a `toString()` metódust, úgy, hogy az visszatérjen a kutya adataival,
pl.: [fajta] vagyok a nevem [név], [getTulajdonsag()] [kora] éves szőrzetem: [getSzore()].
pld: Spániel vagyok a nevem Fáni, vidám 3 éves szőrzetem: hosszú selymes

Készíts egy `Agár` nevű osztályt, ez az osztály származzon a `Kutya` osztályból.

1. Az örökölteken felül 3 saját, privát adatmezeje legyen: `szorzet`, `tulajdonsag`, `tulajdonsagok[]`
2. `tulajdonsagok[] = {"erős", "kitartó", "félénk", "elegáns", "gyors"}`
3. Írj konstruktort az osztálynak, amely két paramétert vár: `név`, `kor`
 - hívja meg a szülőosztály konstruktorát úgy, hogy a fajta attribútum az Agár objektumok esetében mindig "Agár" legyen! (`super`)
 - `this`-ek
 - a `tulajdonsag` az értéket a `tulajdonsagok[]`-ből a szülő osztály `rnd` függvényét használva állítja elő (a felsorolt tulajdonságok közül véletlenszerűen kiválaszt egyet).
4. A szülőosztály két szabályt állított ki. Ezeket a Getter-ek megválaszolják. (`@Override`)
5. Az agár szőre: "rövid"

Készíts egy `Spaniel` nevű osztályt, ez az osztály származzon a `Kutya` osztályból.

Az osztály pontosan olyan, mint az Agár, csak a saját speciális tulajdonságokban különbözik:

- `tulajdonsagok[] = {"szeretetteljes", "embercentrikus", "vidám", "társaságkedvelő", "hűséges"}`
- a spániel szőre: "hosszú selymes"

Készíts egy `Puli` nevű osztályt, ez az osztály származzon a `Kutya` osztályból.

Az osztály pontosan olyan, mint az Agár, csak a saját speciális tulajdonságokban különbözik:

- `tulajdonsag = "terelek"`
- a puli szőre: "rasztás szalagos"

Futtatható osztály: feltöltés és lekérdezések feladatok.

Definiáljon egy 70 elemű tömböt a kutyák tárolására: pld. `Kutya kutyamenhely[]`

Definiáljon egy tömböt a kutyák nevének: pld. `nevek[] = {"Böbe", "Lángos", ..., }`

Készítsen **rnd** függvényt ugyan úgy, mint a `Kutya` osztályban (másolja ide), melynek 3 feladata lesz:

- véletlenszerűen válasz egy nevet a `nevek[]`-ből
- véletlenszerűen generál egy életévet
- a véletlenszerűen választ egy kutyaajtát

`f1()` : Feltölti a `kutyamenhely[]`-t ciklussal.

név: véletlenül választ egy nevet a `nevek[]`-ből `rnd(0, nevek.hossz-1)`

év: [1, 10]-ből egy egészszám `rnd(1, 10)`

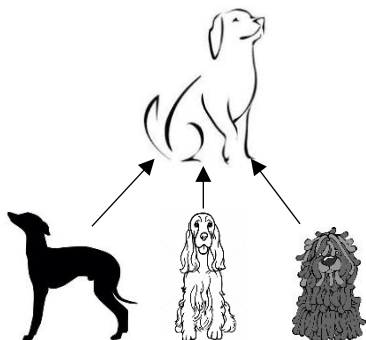
kutya fajta kiválasztása: [0, 2]-ből egy egészszám `rnd(0, 2)`

`f2()` – `f4()` : Lekérdezések *Kiválogatás-tétele*

A `main` függvényben az `f1()` metódus feltölti a kutyákat és az `f2()` – `f4()` lekérdezi őket.

nyilvános (**public**; jelölése: +)
védett (**protected**; jelölése: #)
privát (**private**; jelölése: -)

- az objektum adatai és metódusai bárhol is elérhetők;
- az objektum adatait és metódusait csak az adott objektum és annak leszármazottjai érhetik el;
- csak az adott objektum számára elérhetők az adatok és a metódusok.



abstract Kutya	
- fajta, név	:String
- kor	:int
+ Kutya (fajta :String, név :String, kor :int) this...	
+ getFajta()	:String
+ getKor()	:int
+ getNév()	:String
+ toString()	:String
# rnd(ah :int, fh :int)	:int
absztrakt getSzőrzet()	:String
absztrakt getTulajdonság ()	:String

AppKutyák	
- kutyamenhely [70]	:Kutya
- nevek={...}	:String
- rnd(ah :int, fh :int)	:int
- f1()	:void
- f2()	:void
- f3()	:void
+ main () :void f1() f2() f3()	

Agár extends Kutya	
- tulajdonságok[]={...}	:String
- tulajdonság, szőrzet	:String
+ Agár (név :String, kor :int) super("Agár", név, kor) this.tulajdonság=... this.szőrzet=...	
+ getTulajdonság()	:String
+ getSzőrzet()	:String

Spániel extends Kutya	
- tulajdonságok[]={...}	:String
- tulajdonság, szőrzet	:String
+ Spániel (név :String, kor :int) super("Spániel", név, kor) this.tulajdonság=... this.szőrzet=...	
+ getTulajdonság()	:String
+ getSzőrzet()	:String

Puli extends Kutya	
- tulajdonság, szőrzet	:String
+ Puli (név :String, kor :int) super("Puli", név, kor) this.tulajdonság="terelek" this.szőrzet=...	
+ getTulajdonság()	:String
+ getSzőrzet()	:String