**Maturitní příklad 2016/2017 – PRG**

Vytvořte aplikaci, která bude jednoduchým registrem psů

Objekt **pes**:

* Vlastnosti :
	+ **id** … jednoznačné Id (celé číslo) v systému
	+ **rasa** (řetězec)
	+ **vek** – celé číslo
	+ **ockovan...je/není aktuálně očkovaný** (bool)
* Metody:
	+ ***Vypis()*** – vrátí string - všechny informace o psu
		- Vlastnost ockovan převeďte va výpisu nastring *očkován/neočkován*
	+ **get** pouze na id, vek, a ockovan !! vše ostatní bude zapouzdřené uvnitř objektu
	+ Konstruktor ***pes(int \_id,.., bool \_ockovan)***

Objekt **registr\_psi:**

* Vlastnosti:
	+ Kolekce – ArrayList (můžete vybrat jinou vhodnou na daný problém, třeba List)
* Metody:
	+ Public Konstruktor
		- ***registr\_psi()***
	+ Public ***int Pridej(Pes p)*** – přidá psa do kolekce a vrátí Id přidaného psa, existuje-li už pes se stejným Id v seznamu, pak metoda vrátí číslo : -1, metoda využívá metodu popsanou níže.
	+ Private bool ***Je\_v\_registru(int \_ID)*** – vrátí true/false jeli už pes se stejným id v seznamu
	+ Public string ***Vypis(bool \_all, bool \_ockovan)***
		- Pokud ***\_all*** bude **true** bude ignorovat další parametr a vypíše seznam všech psů v seznamu
		v hlavičce výpisu bude údaj, kolik psů je v registru a jaký je průměrný věk psů v seznamu.
		K výpisu hlavičky použijte private metodu popsanou níže.
		- Pokud ***\_all*** bude **false** vypíše pouze ty psy, kteří podle parametru \_ockovan odpovídají tomuto parametru.
		- Pro získání údaje o psovi samozřejmě využijte metodu ***Vypis()*** na objektu **pes**
	+ Private double ***Prumerny\_vek()***
		- Vrátí průměrný věk **všech psů v registru** (hodnotu zaokrouhlete na jedno desetinné místo!!)
		- Pokud při výpočtu dojde k chybě vrátí metoda hodnotu –„1“

**Hlavní program:**

Metody

***Zadej()***: Slouží k načtení údajů o psu, vytvoří objekt psa a přidá ho do registru pomocí metody ***Pridej(…)***. Metoda musí být ošetřena proti chybám. Pokud dojde k chybě v zadání opakujte volání metody.

* Id musí být celé číslo větší než 0.
* Věk musí být větší než 1 a menší než 50 včetně krajních hodnot.
* Rasa nesmí být prázdný řetězec

Na základě výsledku volání metody ***Pridej()*** vypíšte, zda byl či nebyl pes přidán do registru.

***Vypis(true,….)***: Vypíše obsah registru všech psů na konzoli.

.

***Menu()***: Zobrazí jednoduché menu a postará se o volání patřičných metod:

Volby:

* Stiskněte v/V k výpisu všech psů
* Stiskněte w/W k výpisu na základě patametrů – uživatel bude zadávat
0-očkované/1-neočkované :
na základě tohoto vstupu budou vypsáni jen psi vyhovující zadání (očkovaní/neočkovaní)
* Stiskněte z/Z k zadání dalšího psa
* Stiskněte q/Q Konec programu

Metoda Menu by měla ignorovat jakékoliv jiné klávesy než ty které slouží k výběru Menu !!!

\* dop.: metoda může vracet např. true/false pro jednodušší použití v cyklu hlavního programu.

Výpis programu bude např. při volbě v tento:



Znovu se zobrazí Menu programu pokud nebylo stisknuto Q/q.

**Důležité upozornění !!!!! :**

**Ke splnění úkolu můžete používat pouze oficiální HELP vývojového prostředí. Jakýkoliv pokus o pohyb v složkách, kde není uložen Váš projekt (popsáno níže) nebo pohyb po jiných web stránkách než oficiálním helpu aplikace bude znamenat okamžité ukončení zkoušky a pokud bude hodnocen známkou NEDOSTATEČNĚ !!!**

**V případě shody dvou a více kódů různých studentů budou známkou NEDOSTATEČNĚ hodnoceny všichni majitelé identických kódů.**

Hodnocení:

* oop objekty(metody get, konstruktor, specifikátory přístupu)
* alg - použití průchodu kolekce (foreach), cykly while a podmínky,přehlednost kódu, pojmenování proměnných, komentáře
* chyby a kontrola(try catch) návratové hodnoty metod a práce s nimi – v celém programu !
* funkčnost aplikace (zpracování metod v Main()), volání metod, komplexnost splnění úkolu, Menu, kód Programu a použití vytvořených objektů

Za každou kategorii hodnocení může získat student max. 3 body

0b. nezpracoval
1b. částečně zpracováno (hrubé chyby, bránící funkčnosti)
2b. zpracováno s drobnými chybami – částečně mohou být příčinou generování chyb
3b. zpracováno bez připomínek

Známka:

12-11b. 1
10-9b. 2
8-7b. 3
6-5b. 4
<5b. 5