**Maturitní příklad 2017-2018 – PRG**

Vytvořte aplikaci, která bude jednoduchým registrem dárců krve.

Objekt **darce**:

* Vlastnosti :
  + **id** … jednoznačné Id (celé číslo) v systému
  + **skupina\_krve** (řetězec „A“,“AB“,“B“,“0“)
    - varianta může být výčtový typ jako bonus
    - Hodnoty, které bude property nabývat nemusíte na úrovni objektu řešit
  + **vek** – celé číslo
  + **aktivni** ... je/není aktuálně aktivním dárcem krve (bool)
* Metody:
  + ***Vypis()*** – vrátí string - všechny informace o dárci
    - Vlastnost aktivni převeďte va výpisu na string *aktivní/neaktivní*
  + **get** pouze na id, vek, a aktivni !! vše ostatní bude zapouzdřené uvnitř objektu
  + Konstruktor **d*arce(int \_id,.., bool \_aktivni)***

Objekt **registr\_darci:**

* Vlastnosti:
  + Kolekce – ArrayList (můžete vybrat jinou vhodnou na daný problém, třeba List)
* Metody:
  + Public Konstruktor
    - **registr\_darci *()***
  + Public ***int Pridej(darce* d*)***
    - Přidá dárce do kolekce a vrátí Id přidaného dárce, existuje-li už dárce 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 dárce se stejným id v seznamu
  + Public string ***Vypis\_all()***
    - Vrátí řetězec - seznam všech dárců v seznamu  
      v hlavičce výpisu bude údaj, kolik dárců je v registru a jaký je průměrný věk dárců v seznamu.  
      K výpisu hlavičky použijte private metodu na výpočet průměrného věku popsanou níže.
    - Pro získání údaje o dárci samozřejmě využijte metodu ***Vypis()*** na objektu **darce**
  + Public string ***Vypis\_selected (bool \_aktivni )***
    - Vrátí řetězec - seznam budˇ aktivních nebo neaktivních dárců (podle parametru) v seznamu  
      v hlavičce výpisu bude informace – text – za jde o výpis aktivních nebo neaktivních dárců.
    - Pro získání údaje o dárci samozřejmě využijte metodu ***Vypis()*** na objektu **darce**
  + Public int ***Pocet\_darcu\_skupiny(string \_skupina)***
    - Vrátí počet dárců, které mají stejnou skupinu jako parametr metody
  + Private double ***Prumerny\_vek()***
    - Vrátí průměrný věk **všech dárců 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 dárci, vytvoří objekt darce 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 150 včetně krajních hodnot.
* Skupina musí být řetězec „A“,“B“,“AB“,“0“ – řetězec nula.

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

.

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

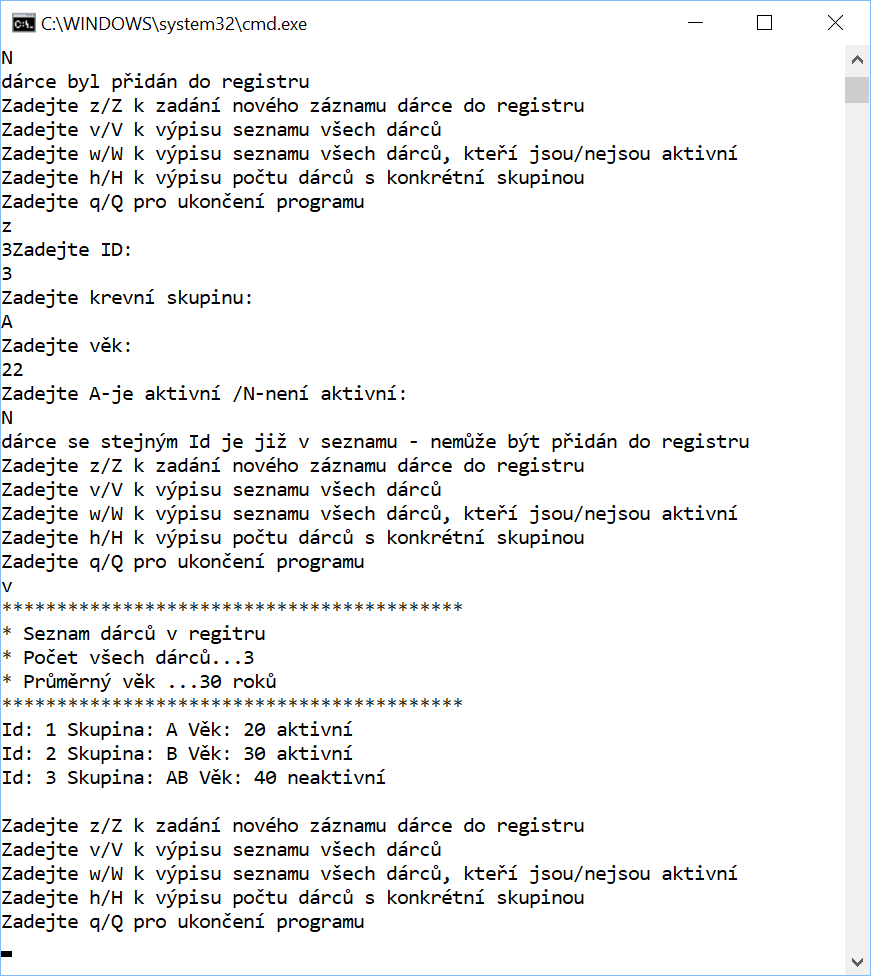
Volby:

* Stiskněte v/V k výpisu všech dárců
* Stiskněte w/W k výpisu na základě parametru – uživatel bude zadávat  
  0-aktivní/1-neaktivní :  
  na základě tohoto vstupu budou vypsáni jen dárci vyhovující zadání (tedy aktivní/neaktivní)
* Stiskněte h/H k vyhledání konkrétního počtu dárců zadané skupiny – skupinu bude zadávat uživatel
* Stiskněte z/Z k zadání dalšího dárce
* Stiskněte q/Q Konec programu

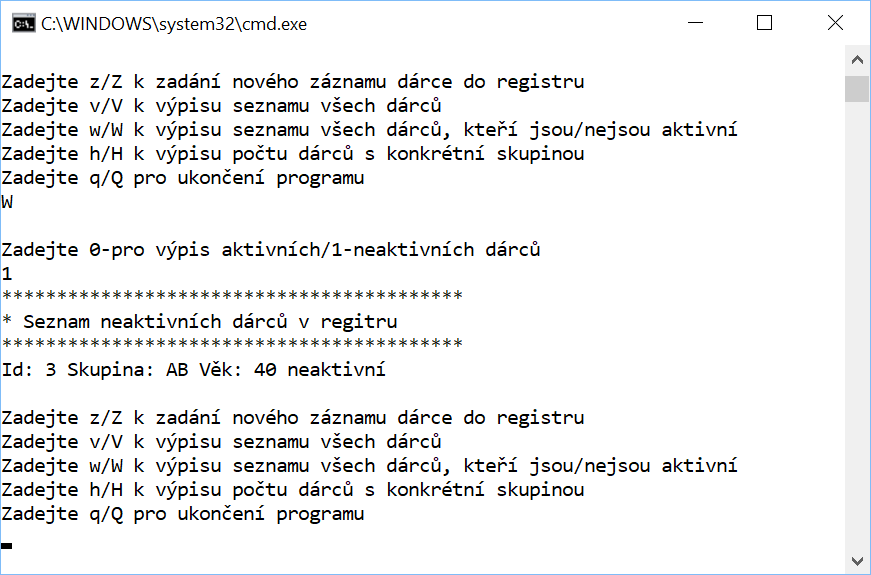
Metoda Menu by měla ignorovat jakékoliv jiné klávesy než ty které slouží k výběru Menu !!!

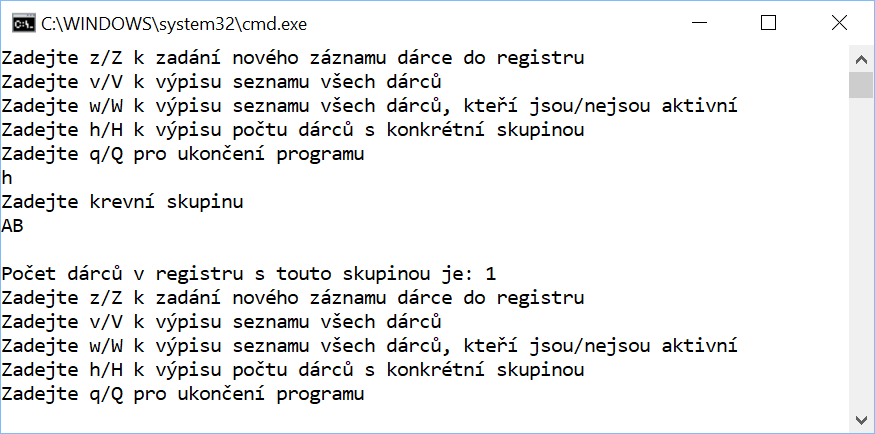
\* dop.: metoda může vracet např. true/false pro jednodušší použití v cyklu hlavního programu.

Výpis programu bude např. 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

**Uložení projektu**

Projekt na konzoli nazvěte ***Vaše příjmení\_2018***(např. Novak\_2018)

Umístění projektu dohodněte před počátkem práce s vyučujícím provádějícím dozor !!!

Před koncem práce **se ujistěte, že celý projekt je uložen v této složce**, na neuložené projekty, nebo částečně uložené nebude brán po době odevzdání zřetel!!!!