**Maturitní příklad 2018-2019 – PRG**

Vytvořte aplikaci, která bude jednoduchým registrem tiskových dokumentů.

Objekt **dokument** :

* Vlastnosti :
  + **id** … jednoznačné Id (celé číslo) v systému
  + **typ** (řetězec „doc“,“pdf“,“txt“,“jiný“)
    - varianta může být výčtový typ jako bonus
    - Hodnoty, které bude property nabývat nemusíte na úrovni objektu řešit
  + **velikost (v bytech)** – celé číslo
  + **k\_archivaci** ... je/není určen k archivaci (bool)
* Metody:
  + ***Vypis()*** – vrátí string - všechny informace o dárci
    - Vlastnost k\_archivaci převeďte va výpisu na string *archivovat/nearchivovat*
  + **get** pouze na ***id***, ***velikost***, a ***k\_archivaci*** !! vše ostatní bude zapouzdřené uvnitř objektu
  + Konstruktor ***dokument(int \_id,.., bool \_k\_archivaci)***

Objekt **registr\_dokumenty:**

* Vlastnosti:
  + Kolekce – ArrayList (můžete vybrat jinou vhodnou na daný problém, třeba List)
* Metody:
  + Public Konstruktor
    - **registr\_dokumenty *()***
  + Public ***int Pridej(dokument* d*)***
    - Přidá dokument do kolekce a vrátí Id přidaného dokumentu, existuje-li už dokument 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 dokument se stejným id v seznamu
  + Public string ***Vypis\_all()***
    - Vrátí řetězec - seznam všech dokumentů v seznamu  
      v hlavičce výpisu bude údaj, kolik dokumentů je v registru a jaká je celková velikost všech dokumentů v seznamu.  
      K výpisu hlavičky použijte private metodu na výpočet celkové velikosti všech dokumentů popsanou níže.
    - Pro získání údaje o dokumentu samozřejmě využijte metodu ***Vypis()*** na objektu **dokument**
  + Public string ***Vypis\_k\_archivaci (bool \_k\_archivaci )***
    - Vrátí řetězec - seznam dokumentů určených k archivaci, nebo ostatních (podle parametru)
    - v hlavičce výpisu bude informace – text – za jde o výpis dokumentů určených k archivaci nebo neurčených k archivaci.
    - Pro získání údaje o dokumentu samozřejmě využijte metodu ***Vypis()*** na objektu **dokument**
  + Public int ***Pocet\_dokumentu\_typ(string \_typ)***
    - Vrátí počet dokumentů, které mají stejný typ, jako parametr metody
  + Private double ***Celkova\_velikost()***
    - Vrátí celkovou velikost **všech dokumentů v registru**
    - Výsledek přepočtěte na MB a zaokrouhlete na 2 desetinná místa
    - Pokud při výpočtu dojde k chybě vrátí metoda hodnotu –„1“

**Hlavní program:**

Metody

***Zadej()***: Slouží k načtení údajů o dokumentu, vytvoří objekt dokument 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.
* Velikost musí být větší než 0 .
* Skupina musí být řetězec „doc“,“pdf“,“txt“,“jiný“ – řetězec nula.

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

.

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

Volby:

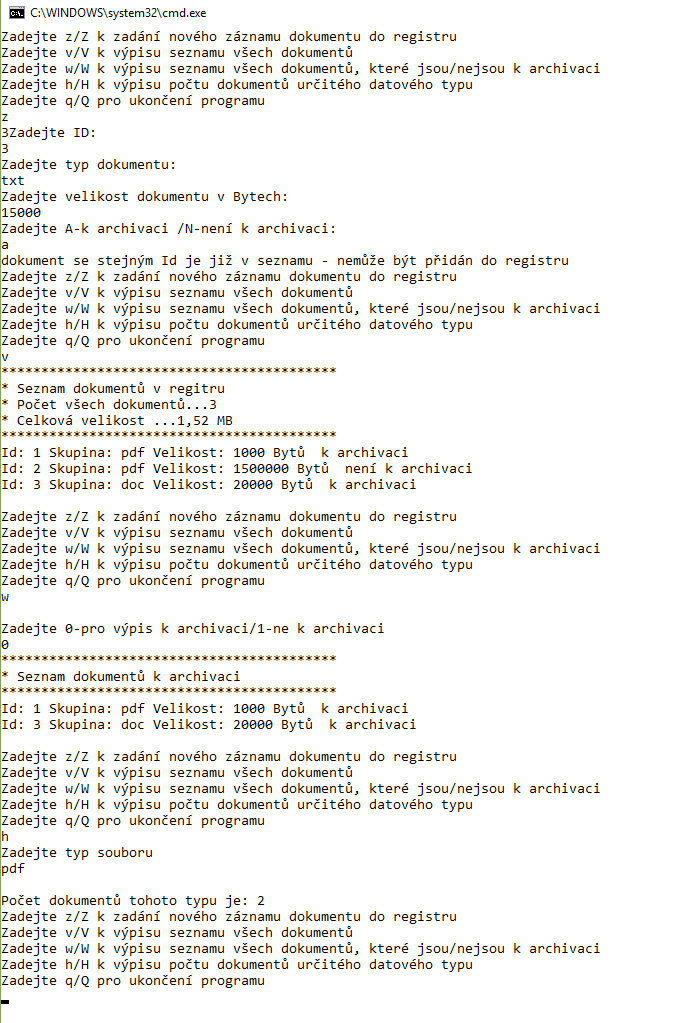
* Stiskněte v/V k výpisu všech dokumentů
* Stiskněte w/W k výpisu na základě parametru – uživatel bude zadávat  
  0-jsou/1-nejsou archivovány :  
  na základě tohoto vstupu budou vypsáni jen dokumenty vyhovující zadání (tedy k archivaci/ne k archivaci)
* Stiskněte h/H k vyhledání konkrétního počtu dokumentů zadaného typu – typ bude zadávat uživatel (jiný - považujte za typ)
* Stiskněte z/Z k zadání dalšího dokumentu
* 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\_2019***(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!!!!