

Készítsen egy programot, amely dolgozók adatait fogja tárolni!

Hozzon létre egy Alkalmazott osztályt. Az alábbi ábrán látható adattagokat és metódusokat tartalmazza:

Alkalmazott
- String nev - int fizetes - int gyerekek - int eletkor - int tulora
+ Alkalmazott(String nev, int fizetes , int gyerekek , int eletkor, int tulora) + String toString(); + String getNev(); + int getFizetes(); + int getGyerekek(); + int getEletkor(); + int getTulora();

Adattagok (privát láthatóságúak):

- A nev az alkalmazottak nevét fogja tárolni.
- Az fizetes az alapfizetéseket tárolja.
- A gyerekek az alkalmazott gyermekeinek számát tárolják.
- Az eletkor az alkalmazott életkorát tárolja.
- A tulora az alkalmazott túlóráit tárolja.
- Az Alkalmazott konstruktora a paramétereiben megkapott értékeket, értékül adja az osztály adattagjainak.
- toString() metódusa formázottan kiírja az alkalmazott adatait.
- A get metódusok az adott adattag értékeit adják vissza.

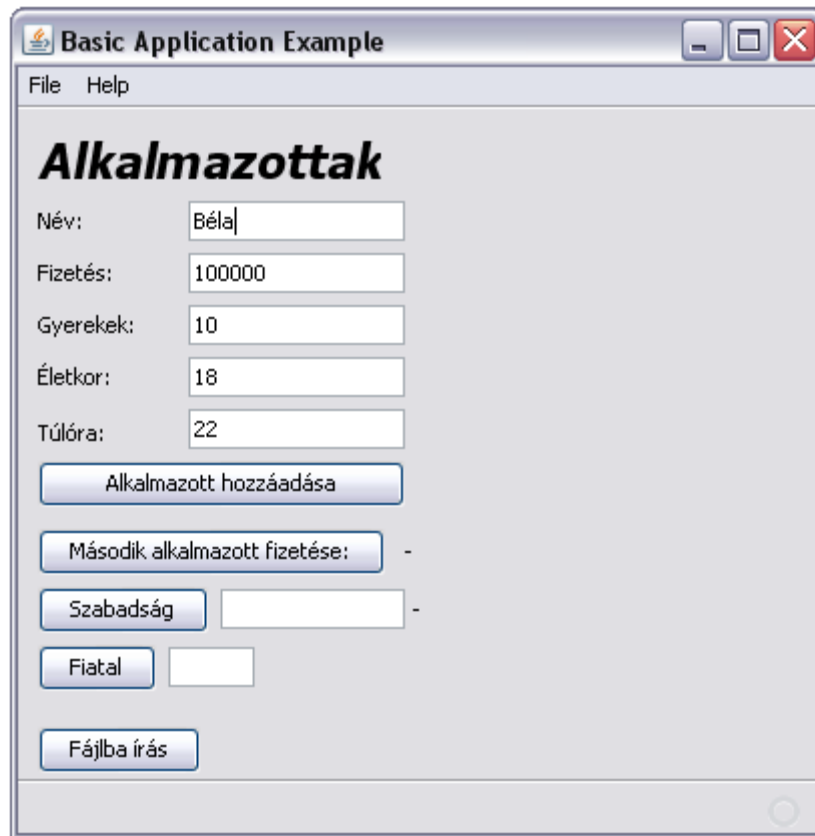
Hozzon létre egy AlkalmazottKezelo osztályt. Ez az osztály alkalmazottakat fog kezelni. Láncolt listákban vagy vektorokban, helyezkednek el az alkalmazottak adatai. Az AlkalmazottKezelo osztály az alábbi ábrán látható adattagokat és metódusokat tartalmazza:

AlkalmazottKezelo
- Vector<Alkalmazott> alkalmazottak
+ void add(String nev, int fizetes, int gyerekek, int tulora, int eletkor) + int getFizetes(int index) + int getSzabadsag(String nev) + void printAlkalmazott(int fiatalabb) + void writeToFile()

Metódusok (publikus láthatóságúak):

- Az add() metódus a paramétereiben megadott alkalmazott adatait felveszi a vektorokba.
- A getFizetes() metódus kiszámolja, a paraméterében megkapott indexű alkalmazott fizetését és visszaadja annak értékét. A fizetés úgy számolandó, hogy az alapfizetéshez hozzáadjuk a túlórából bejött pénzt. A túlórákra 1000Ft számolunk.
- A getSzabadsag() a paraméterében megkapott nevű alkalmazott szabadságát adja vissza. A szabadság úgy számolandó, hogy az életkort elosztjuk 4-el és hozzáadjuk a gyermekek számának kétszeresét.
- A printAlkalmazott() metódus kiírja a standard outputra azon alkalmazottak adatait, amelyek fiatalabbak a paraméterként megadott értéktől.
- A writeToFile() metódus kiírja az alkalmazottak adatait az alkalmazott.txt fájlba.

A program rendelkezzen grafikus felhasználói felülettel, ami az alábbi ábrán látható:



A grafikus felhasználói felület a következő komponenseket tartalmazza:

- Öt beviteli mezőt, amelybe az alkalmazottak adatait írhatjuk be.
- Egy gomb (Alkalmazott hozzáadása). Ezt lenyomva az alkalmazott adatai hozzáadódnak a vektorokhoz.
- Egy gomb (Második alkalmazott fizetése). Ezt lenyomva meghívódik a `getFizetes()` metódus, és kiszámolja a második indexű alkalmazott fizetését és kiírja a mellette lévő címkére.
- Egy gomb (Szabadság). Ezt lenyomva meghívódik a `getSzabadsag()` függvény és a mellette lévő szövegbeviteli mezőben megadott alkalmazottat megkeresi a program és kiszámolja a szabadságát, majd a mellette lévő címkére kiírja az eredményt.
- Egy gomb (Fiatal). Ezt lenyomva meghívódik a `printAlkalmazott()` metódus, paraméterül kapja a gomb mellett lévő beviteli mező értékét.
- Egy gomb (Fájlba írás). Ezt lenyomva meghívódik a `writeToFile()` metódus és a program kiírja az alkalmazott.txt fájlba az alkalmazottak adatait.

Ügyeljen arra, hogy adatbekéréseknél, fájlba írásnál a hibás adatokat lekezelje a program egy kivételkezelővel.

A projektfájlokat tömörítse egy állományba és mentse el a vizsgáztató által megadott helyre. A tömörített fájl neve legyen a vizsgázó teljes neve.