

Napisać program wspomagający organizację w firmie.

W programie znajduje się lista pracowników (instancji klasy) posiadających następujące cechy (prywatne):

- imię (imię, String)
- nazwisko (nazwisko, String)
- płeć (plec, char) // 'K' – kobieta, 'M' – mężczyzna
- numer działu (nr_działu, int)
- płaca (placa, float)
- wiek (wiek, int)
- liczba dzieci (dzieci, int)
- stan cywilny (stan_cywilny, boolean) // true – mężatka / żonaty

oraz metody:

- wyświetlanie wszystkich danych o pracowniku, // procedura bez parametrów;
- wyświetlanie okrojone – tylko imię, nazwisko i płaca // procedura bez parametrów;
- wyświetlanie specjalne – tylko imię i nazwisko (wszystko drukowanymi literami), // procedura bez parametrów;
- sprawdzenie czy pensja jest powyżej podanej wartości // funkcja (zwraca true – powyżej, false – poniżej lub równa) z parametrem, który oznacza pensję do porównania (float);
- obliczanie podwyżki // procedura z parametrem, który określa procent podniesienia pensji – dodatkowo, za każde dziecko pracownika zwiększa mu się pensję o 2 procent, natomiast pracownicy będący w formalnym związku dodatkowo dostają 3 procent podwyżki) ;
- i inne metody pozwalające na dostęp do odpowiednich pól (zwracanie wartości każdego pola, zmiana wartości dla pól: numer działu, płaca, wiek, stan cywilny i liczba dzieci.

Podstawowe funkcje jakie musi zawierać zaimplementowany program, to:

1. Wypisywanie listy wszystkich pracowników (okrojone) – pozycja 1 w menu.
2. Możliwość dodawania nowych pracowników – pozycja 2 w menu.
3. Eksport – pozycja 3 w menu. Eksport pliku elementowego (obiektów) do pliku tekstowego (nazwa pliku tekstowego jest podawana przez użytkownika) w postaci:
nazwisko imię płeć numer_działu płaca wiek liczba_dzieci
4. Usuwanie pracowników – pozycja 4 w menu
Po wyświetleniu wszystkich danych o wszystkich pracownikach następuje wybór pracownika do usunięcia.
5. Edycja danych – pozycja 5 w menu
Po wyświetleniu wszystkich danych o wszystkich pracownikach następuje wybór pracownika do edycji. Wybrany pracownik zostaje wyświetlony jeszcze raz (wyświetlanie specjalne), a użytkownik z nowego menu wybiera pole które chce edytować (istnieje możliwość edycji wszystkich pól poza płcią i imieniem, a nazwisko może być edytowane tylko w przypadku kobiet).
6. Dodatkowe funkcje – pozycja 6 w menu.

Wybranie tej opcji powoduje wyświetlenie nowego menu z poniższymi opcjami (funkcjonalnościami):

- a) Obliczanie liczby pracowników z pensją nie mniejszą niż podana przez użytkownika // funkcja z parametrem float – podana pensja.
- b) Obliczanie średniej płacy w dziale (numer działu podaje użytkownik) // funkcja z parametrem int – podany numer działu.
- c) Wyświetlanie największych pensji z wszystkich kobiety i wszystkich mężczyzny // procedura bez dodatkowych parametrów.

*d) Wyświetlenie wszystkich działów (tylko tych, które aktualnie mają pracowników) i informacji, czy większość pracowników jest kobietami, mężczyznami, czy ich liczba rozkłada się równomiernie. Do tego, przy każdym dziale wyświetlana zostaje średnia pensja w tym dziale (można skorzystać z funkcji napisanej na potrzeby podpunktu 'b') // procedura bez dodatkowych parametrów.

e) Wyświetlenie stosunku średniej płacy kobiet do średniej płacy mężczyzn // funkcja bez dodatkowych parametrów.

f) Zwiększanie pensji wszystkim pracownikom o 10 procent oraz dodatkowo za specjalne cechy (zgodnie z metodą obiektu).

*g) Zwiększenie pensji wszystkim pracownikom o kwotę podaną przez użytkownika, wyświetlona zostaje suma podwyżek oraz stosunek podwyżek dla kobiet i mężczyzn. // funkcja z parametrem float – podana kwota podwyżki. Suma podwyżek wyświetlona zostaje w funkcji, natomiast zwrócony zostaje wymagany stosunek.

*h) Sortowanie pracowników (w pliku) według nazwiska // procedura z parametrem boolean – true, to rosnąco; false, to malejąco.

*i) Sortowanie pracowników (w pliku) według wysokości ich pensji. // procedura z parametrem boolean – true, to rosnąco; false, to malejąco.

7. Dodatkowe funkcje dla plików tekstowych – pozycja 7 w menu.

Wybranie tej opcji powoduje eksport danych do pliku tekstowego i wyświetlenie nowego menu z poniższymi opcjami (funkcjonalnościami):

a) Wyświetlenie danych (linii z pliku) o osobie z najdłuższym nazwiskiem. Jeśli kilka osób ma taką samą długość nazwiska wystarczy wyświetlić jedną. // procedura bez dodatkowych parametrów.

b) Obliczenie średniego wieku osób posiadających dzieci. // funkcja bez dodatkowych parametrów.

c) Zakodowanie niektórych danych w pliku (tym samym – tekstowym) w ten sposób, że nazwisko wpisane zostaje na zasadzie:

N*****i zamiast Nowakowski

Należy zakodować nazwiska tylko tych osób, których zarobek jest niższy od średniego zarobku wszystkich pracowników.

*d) Utworzenie pliku 'pracownicy.html' (tekstowy), w którym na podstawie pliku tekstowego o pracownikach tworzona jest tabela zapisana w HTML w taki sposób, na początku nazwane zostają nagłówki (w apostrofach wyróżnione zostały kolejne kolumny):

'Nazwisko' 'Imię' 'Płeć' 'Numer działu' 'Płaca' 'Wiek'

a później odpowiednio wpisane zostają wszystkie dane. // procedura bez dodatkowych parametrów.

8. Informacja o programie – pozycja 8 w menu.

9. W chwili uruchomienia (i tylko w tej chwili) należy przyjąć, że nazwa pliku obiektów (elementowego) to „baza.dat”, ale użytkownik poprzez kolejną opcję (9 pozycja menu) może wprowadzić własną nazwę pliku.

Dodatkowe informacje:

1. Wszystkie pola klasy są prywatne, a metody publiczne. Dostęp do pól powinien odbywać się za pomocą metod (np. setPole / getPole).
2. W przypadku **każdej funkcji (i procedury) z punktu 6**, jako pierwszy parametr przekazana zostaje nazwa pliku elementowego i dopiero w funkcji należy wykonać wszelkie operacje na pliku. Poza tym parametrem i innymi (wymienionymi w powyższym opisie) nie można przekazywać do funkcji nic innego.
3. W przypadku **każdej funkcji (i procedury) z punktu 7**, jako pierwszy parametr przekazana zostaje nazwa pliku tekstowego (tworzonego tuż przed wykonaniem funkcji, poprzez eksport z pliku elementowego).
4. Nie należy stosować zmiennych globalnych.
5. Liczba pracowników zapisanych w programie nie będzie większa niż 100. W przypadku konkretnych rozwiązań można zastosować tablicę 100 elementową (lokalną wewnątrz funkcji).
6. Po wczytaniu imienia lub nazwiska należy doprowadzić je do takiej postaci, że wszystkie litery zostają zmienione na małe, a pierwsza ma być wielka (np. użytkownik podaje naZWisko, a pole obiektu przyjmie wartość Nazwisko).
7. Imiona oraz nazwiska są jednym ciągiem liter (bez białych znaków).
8. Nie należy stosować niestandardowych bibliotek.
9. Program musi być napisany w wersji konsolowej.