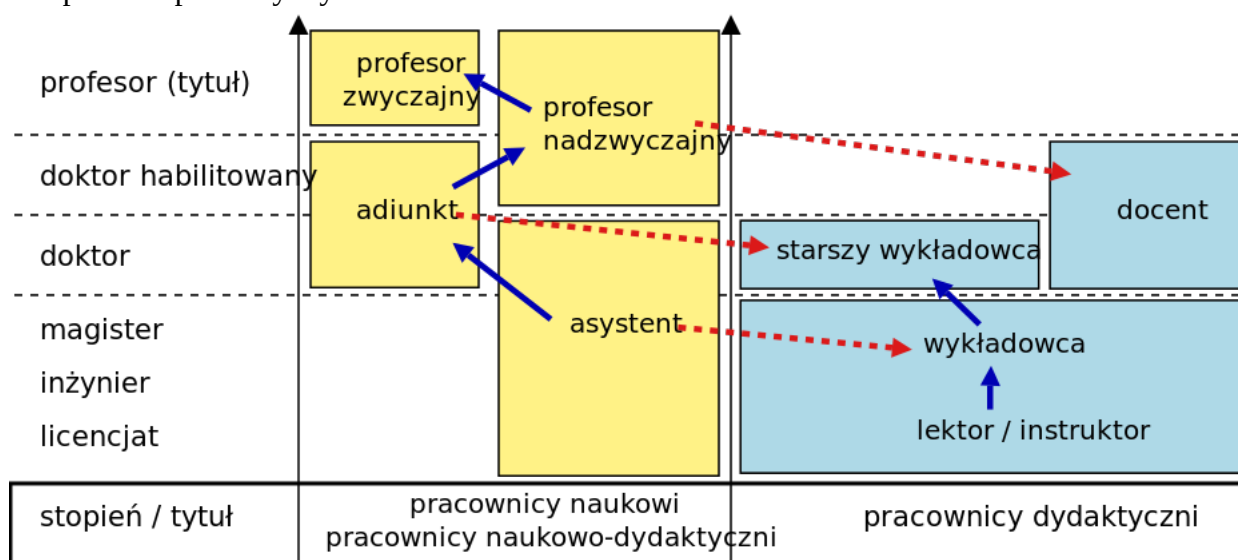


Zadanie: Napisać program symulujący kadre wydziału Elektroniki i Technik Informacyjnych, z uwzględnieniem stopni naukowych, zajmowanych stanowisk, i przynależności do instytutów oraz zakładów.

Założenia implementacyjne:

- kolejność promocji, oraz sposób degradacji pracowników naukowych zostały zaimplementowane w oparciu o poniższy wykres:



źródło: https://pl.wikipedia.org/wiki/Pracownicy_uczelni_w_Polsce

Oprócz promocji i degradacji istnieje także możliwość ręcznej zmiany zajmowanej pracy naukowej na wyspecyfikowaną przez użytkownika. Program nie sprawdza czy użytkownik spełnia wymagania, aby mieć możliwość zająć dane stanowisko, przytoczone przez powyższy wykres, gdyż po krótkim przeszukaniu kadry pracowniczej znaleźć można wiele przypadków dla których owe wymagania są nieprawdziwe.

- domyślna kadra pracownicza, znajdująca się w pliku elka.txt pochodzi ze strony:
<https://www.bip.pw.edu.pl/Sklad-osobowy/Wydzialy/Wydzial-Elektroniki-i-Technik-Informacyjnych>

- pracownicy administracyjni mogą być jednym z: kierownik zakładu, dziekan, prodziekan, dyrektor instytutu, zastępca dyrektora instytutu. Promować i degradować nie można tylko kierownika zakładu, ze względu na brak oczywistej ścieżki promocji.

- w przypadku gdy pracownik zajmuje 2 różne stanowiska (np. prodziekan i profesor zwyczajny) program tworzy 2 instancje danego pracownika (1 administracyjną i 1 naukową). Na potrzeby operacji na pracownikach (promocja, degradacja, zwolnienie itd.), działają one jak jeden pracownik. Jeżeli jednak chciałoby się takiego pracownika dodać, trzeba dodać najpierw jego instancję naukową, a potem jego instancję administracyjną. Tak też pracownik zostanie wypisany przez program.

- wpisanie 'exit' w dowolnym momencie wykonania programu, spowoduje powrót do menu głównego

Enumeracje:

Stopien – stopień naukowy pracowników

Praca_n – stanowisko dla pracowników naukowych

Praca_a – stanowisko dla pracowników administracyjnych

Klasy:

Nauowiec – klasa bazowa dla pracowników naukowych i administracyjnych, sama w sobie nie jest wykorzystywana nigdzie w programie.

Zmienne:

std::string imie – imię pracownika

Stopien stop – najwyższy stopień naukowy zdobyty przez pracownika (np. magister)

Przyjaciele:

Pracownik_n (dziecko)

Pracownik_a (dziecko)

Pracownik_n (is-a Nauowiec) – pracownicy zajmujący stanowiska naukowo-dydaktyczne.

Zmienne:

Praca_n praca – stanowisko zajmowane przez pracownika

Pracownik_a (is-a Nauowiec) – pracownicy zajmujący stanowiska administracyjne.

Zmienne:

Praca_a praca – stanowisko zajmowane przez pracownika

Zakład (has-a Pracownik_a (containment)) – zakład na wydziale, w instytucie.

Zmienne:

std::string imie – nazwa zakładu

Pracownik_a kierownik – kierownik zakładu

Instytut (has-a vector of Pracownik_n (private inheritance), has-a Pracownik_a (containment), has-a vector of Pracownik_a (containment), has-a vector of Zakład (containment)) – zasadnicza klasa przechowująca dane. Wydział interpretowany jest jako vector obiektów typu Instytut. Szczególnym przypadkiem Instytutu jest dziekanat, przechowywany w programie na 0 indeksie głównego vectora Instytutów.

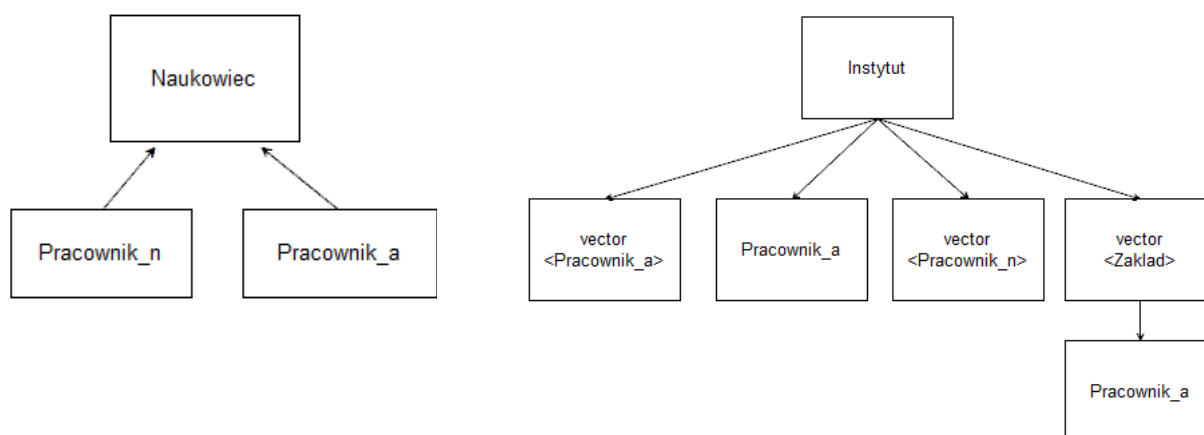
Zmienne:

std::string imie – nazwa Instytutu

Pracownik_a dyrektot – dyrektor Instytutu/dziekan

vector<Pracownik_a> zastepcy – zastępcy dyrektora Instytutu/prodziekani

vector<Zakład> zaklady – zakłady



Interfejs:

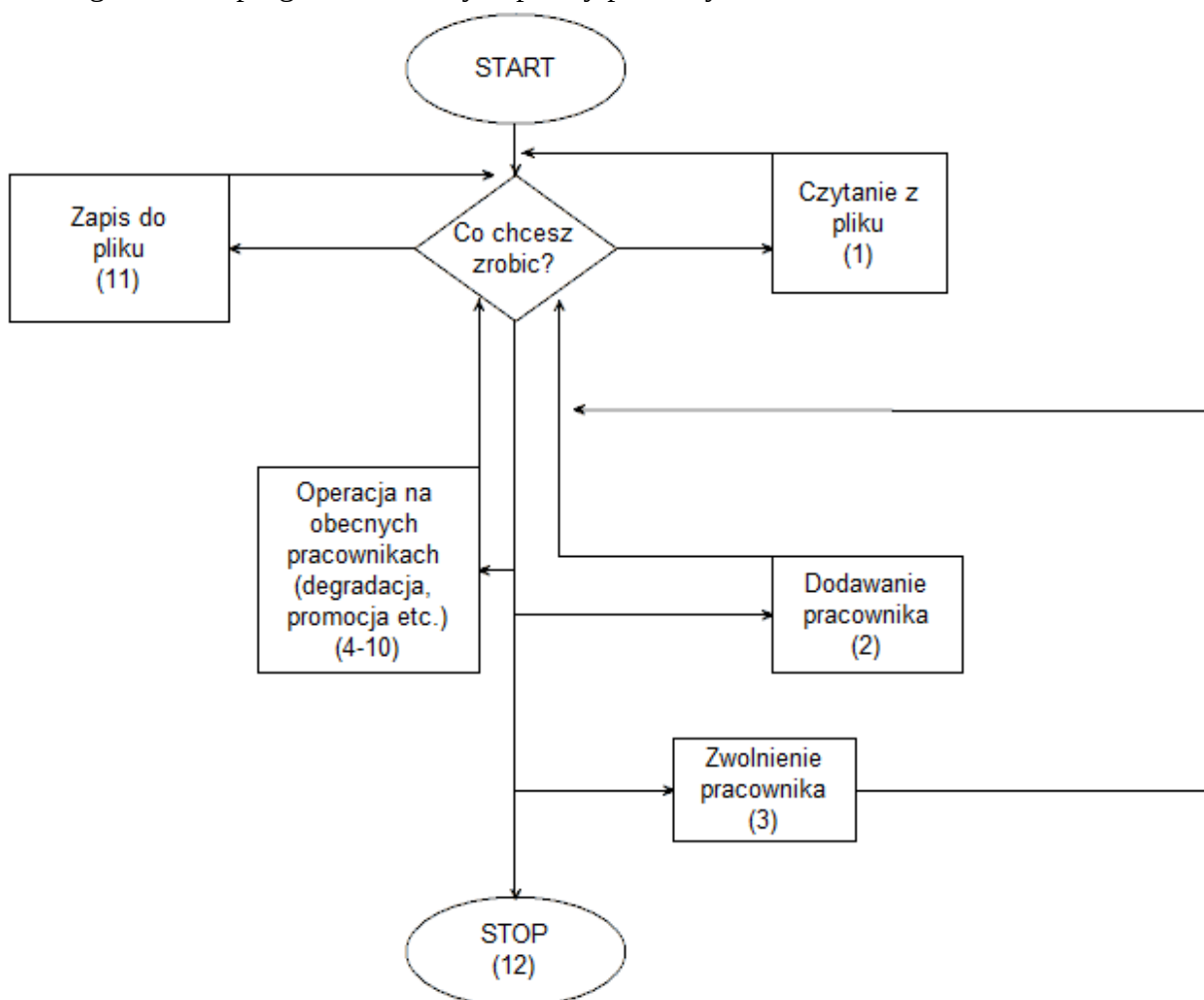
Interakcja z programem opiera się na wpisywaniu przez użytkownika prostych komend. Po włączeniu programu, wyświetli się menu z instrukcjami i opisem co każda komenda robi. Wygląda ono tak:

MENU

Wpisz:

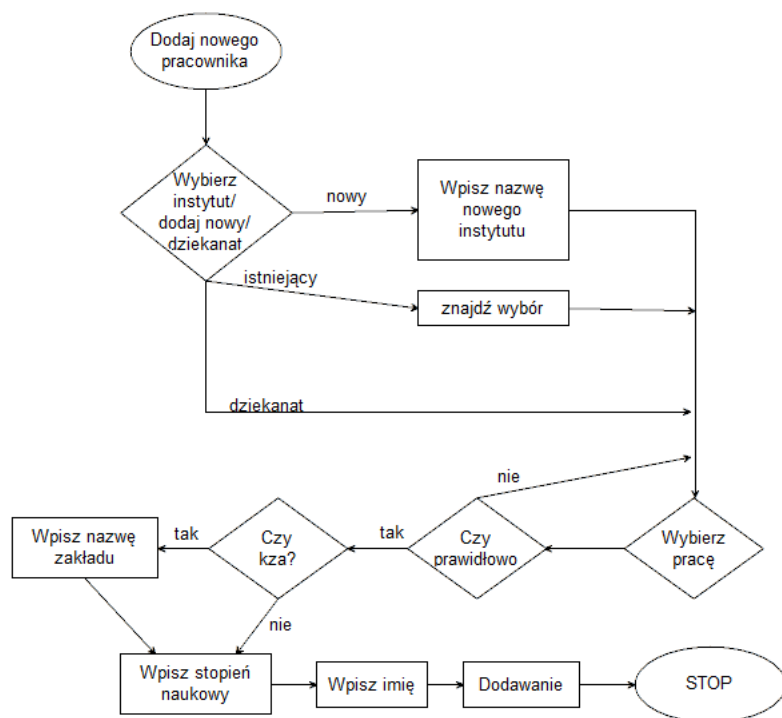
- 'elka' zeby zainicjalizowac dane EiTi
- 'dodaj' zeby dodac pracownika, instytut, badz zaklad
- 'zwolnij' zeby zwolnic pracownika
- 'degraduj' zeby zdegradowac pracownika
- 'promuj' zeby dokonac promocji pracownika na stanowisku naukowym (np. wykladowca->starszy wykladowca)
- 'promuj_n' zeby dokonac promocji naukowej (np. inzynier->magister)
- 'zmien' zeby zmienic stanowisko zajmowane przez pracownika recznie
- 'zmien_n' zeby zmienic tytul naukowy pracownika
- 'pokaz' zeby wydrukowac obecny stan kadry
- 'zapisz' zeby zapisac dane do pliku elkat.txt
- 'zabij' zeby zakonczyc dzialanie programu

Przebieg dzialania programu moze byc opisany ponizszym flowchart-em:

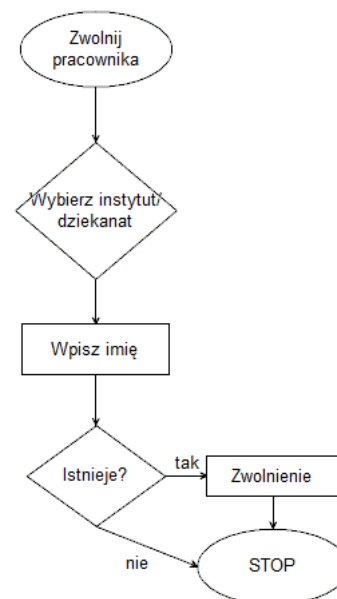


Ponizej znajduja sie rowniez flowcharty wybranych, interaktywnych funkcji, oznaczone liczba w nawiasach jak wyzej.

(2) dodawanie pracownika



(3) zwalnianie



Rozwiązanie:

Dane przechowywane są w pliku elka.txt, z którego program może je przeczytać, bądź je do niego zapisać po edycji. Po załadowaniu, program umożliwia dodanie nowych pracowników, edycję już istniejących i oraz zwalnianie. Dane przechowywane są w zmiennej typu `vector<Instytut>` nazwanej `wydział`. Na indeksie 0 tego wektora, znajduje się dziekanat, inicjalizowany na początku programu, a w nim dane na temat dziekana i prodziekanów. Dopuszczalny jest tylko jeden dziekan na całym wydziale. Na pozostałych indeksach znajdują się instytuty, ładowane z pliku, bądź dodawane przez użytkownika. Każdy instytut ma dane na temat swoich pracowników naukowych (lektorzy, wykładowcy, starsi wykładowcy, docenci, asystenci, adiunkci, profesorzy nadzwyczajni i profesorzy zwyczajni), swoich pracowników administracyjnych (zastępcy dyrektora instytutu i sam dyktor Instytutu), zakładów wraz z ich kierownikami, oraz swoje własne imię, które nie może się powtarzać. Każdy pracownik reprezentowany jest swoim imieniem, stopniem naukowym, oraz pracą jaką wykonuje.

Testy:

Założmy poniższy przebieg programu:

załaduj dane → dodaj pracownika → dodaj pracownika → zwolnij pracownika → promuj pracownika → promuj stopień naukowy pracownika → degraduj pracownika → zapisz dane.

Po otwarciu programu wyświetli się nam menu:

MENU

Wpisz:

- 'elka' żeby zainicjalizować dane EiTi
- 'dodaj' żeby dodać pracownika, instytut, bądź zakład
- 'zwolnij' żeby zwolnić pracownika
- 'degraduj' żeby zdegradować pracownika

- 'promuj' zeby dokonac promocji pracownika na stanowisku naukowym (np. wyklawowca->starszy wyklawowca)
- 'promuj_n' zeby dokonac promocji naukowej (np. inzynier->magister)
- 'zmien' zeby zmienic stanowisko zajmowane przez pracownika recznie
- 'zmien_n' zeby zmienic tytul naukowy pracownika
- 'pokaz' zeby wydrukowac obecny stan kadry
- 'zapisz' zeby zapisac dane do pliku elkat.txt
- 'zabij' zeby zakonczyc dzialanie programu

Zeby zaladowac nalezy wpisac 'elka'. Tak tez zrobmy.

→ elka

Program teraz laduje dane z pliku. Moze to chwile potrwać, gdyz danych jest duzo. Po zaladowaniu ponownie wyswietli sie menu.

Aby dodac pracownika nalezy wpisac 'dodaj'.

→ dodaj

Program wyswietli teraz liste instytutow. Zalozmy ze chcemy dodac pracownika Instytutu Informatyki. Taki wiec instytut specyfikujemy.

→ Instytut Informatyki

Program poprosi nas o wybor pracy. Zalozmy, ze chcemy zeby pracownik byl profesorem zwyczajnym. Zgodnie z wyswietlona instrukcja, wpisujemy prz.

→ prz

Program poprosi nas o wybor stopnia naukowego. Zalozmy ze nasz pracownik jest profesorem. Zgodnie z wyswietlona instrukcja wpisujemy pro.

→ pro

W koncu, program poprosi nas o wpisanie imienia.

→ Michalski Kacper

Program wyswietli teraz komunikat o udanym dodaniu pracownika. Nalezy wcisnac enter aby powrocic do menu.

→ <nie ma znaczenia co tu wpisze, i tak wroci do menu>

Dodajmy kolejnego pracownika. Zalozmy ze nasz pracownik jest dyrektorem nowozalozonego instytutu o nazwie „Instytut C++”. Jak wczesniej, wpisujemy 'dodaj'.

→ dodaj

Jednak tym razem naturalnie naszego Instytutu nie ma na liscie. Zgodnie z wyswietlona instrukcja, wpisujemy '+' aby dodac nowy instytut.

→ +

Program poprosi nas o nazwe instytutu.

→ Instytut C++

Program poprosi nas o prace.

→ din

Program poprosi nas o stopien naukowy. Zalozmy, ze nasz pracownik jest magisrem.

→ mag

Program poprosi nas o imie.

→ Michal Kacperski

Program wyswietli teraz komunikat o udanym dodaniu pracownika. Nalezy wcisnac enter aby powrocic do menu.

→ <caly czas nie ma znaczenia co tu wpisze>

Zwolnijmy jakiegos pracownika. Zalozmy, ze Pan Profesor Zieliński Cezary udal sie na emeryture. Pracowal on w dziekanacie jako prodziekan, i w Instytucie Automatyki i Informatyki Stosowanej jako profesor zwyczajny. Szybkie wyszukanie funkcja 'znajdz' potwierdzi to (→ znajdz → Instytut

Automatyki i Informatyki Stosowanej → Zielinski Cezary). Aby zwolnić pracownika wpisujemy 'zwolnij'

→ zwolnij

Program poprosi nas o wybór instytutu. Należy tu zaznaczyć, że jedynie w przypadku, gdy pracownik, którego chcemy zwolnić, pracuje tylko w dziekanacie, będziemy wpisywać 'Dziekanat'. W każdym pozostałym przypadku wpisujemy nazwę instytutu z którym pracownik jest związany. Pan Zieliński związany jest z dziekanatem i z Instytutem Automatyki i Informatyki Stosowanej, wpisujemy więc nazwę instytutu.

→ Instytut Automatyki i Informatyki Stosowanej

Program poprosi nas o imię.

→ Zieliński Cezary

Program wyświetli komunikat o udanym zwolnieniu.

→

Wypromujmy teraz pracownika. Założmy, że Pani Zoltowska Izabela została profesorem nadzwyczajnym (jest adiunktem). Wpisujemy 'promuj', a następnie nazwę instytutu (Instytut Automatyki i Informatyki Stosowanej).

→ promuj

→ Instytut Automatyki i Informatyki Stosowanej

Program poprosi nas o imię.

→ Zoltowska Izabela

Program wyświetli komunikat o udanej promocji. Sprawdzenia można dokonać funkcją 'pokaz'.

→ pokaz

Program wyświetli najpierw stan dziekanatu, potem po kliknięciu enter każdy kolejny instytut. Zobaczymy, że Pani Zoltowska Izabela została profesorem nadzwyczajnym.

Aby dokonać promocji naukowej należy podążać tymi samymi krokami co w poprzednim przypadku, jedynie zamiast 'promuj', wpisujemy 'promuj_n'.

→ promuj_n

→ Instytut Automatyki i Informatyki Stosowanej

→ Zoltowska Izabela

Także tu można sprawdzić poprawność funkcją 'pokaz'.

→ pokaz

To samo tyczy się funkcji degradującej 'degraduj'.

→ degraduj

→ Instytut Automatyki i Informatyki Stosowanej

→ Zoltowska Izabela

→ pokaz

Zgodnie z wykresem na górze dokumentu, Panie Zoltowska Izabela została docentem.

Aby zapisać dane należy wpisać 'zapisz'.

→ zapisz

Dane te można teraz użyć do odczytu potem.

Aby zakończyć działanie programu, wpisujemy 'zabij'.

-zabij

Jeżeli teraz włączymy program ponownie, możemy wpisać 'elka' aby załadować plik elka.txt, który wcześniej edytowaliśmy.

→ elka

Zobaczmy teraz, że gdy wpiszemy 'pokaz' nasze zmiany zaimplementowane wcześniej zostały zapisane prawidłowo.

→ elka