

Języki i paradygmaty programowania 1 – studia stacjonarne 2025/26

Lab 12. Typ wyliczeniowy, Struktury danych cd.: kolejka (bufor FIFO), stos (bufor LIFO). Projekt nr 4.

1. Typ wyliczeniowy (`enum`) - rodzaj typu danych zawierający listę etykiet z przypisanymi wartościami typu `int`. Etykiety są dostępne globalnie i nie mogą zmieniać wartości. Typ wyliczeniowy pozwala na tworzenie zmiennych tego typu. Przykład (za Wikipedia):

```
enum typ_wyliczeniowy { PUNKT, PROSTA, TROJKAT, KWADRAT, KOLO = 360,  
WIEKSZE_KOLO };  
enum typ_wyliczeniowy eMojaZmienna = PROSTA;  
eMojaZmienna = PUNKT;  
  
/* ... */  
  
if (eInnaZmienna == TROJKAT)  
{  
    /* ... */  
}
```

2. Sprawdź jakie wartości przyjęły poszczególne etykiety typu `MY_INTERF_EN` z poniższego przykładu.

```
enum MY_INTERF_EN  
{  
    INTERF_PUSH,  
    INTERF_POP,  
    INTERF_FIND_LASTNAME,  
    INTERF_DISPLAY,  
    INTERF_COUNT,  
    INTERF_CLEAR,  
    INTERF_STOP,  
    INTERF_TOT  
};
```

3. Projekt nr 4.

Zaimplementuj stos (LIFO) i kolejkę (FIFO) w dwóch wariantach - za pomocą listy oraz za pomocą tablic (w sumie 4 projekty). Napisz menu oraz odpowiednie funkcje pozwalające na:

- dodawanie elementu;
- pobieranie elementu;
- wyszukanie elementu po określonym polu (np. nazwisku);
- wypisanie wszystkich elementów;
- sprawdzenie liczby elementów w strukturze;
- usunięcie całej struktury (wszystkich elementów) – zwolnienie pamięci;
- zakończenie programu - usunięcie wszystkich elementów;
- opcjonalnie: zapis struktury do pliku binarnego;
- opcjonalnie: odczyt struktury z pliku binarnego

Niech danymi kolejki/stosu będą informacje o studentach: imię, nazwisko, rok urodzenia. Do implementacji menu użyj typu wyliczeniowego zdefiniowanego w pkt. 2.

4. Fragment programu:

```
enum MY_INTERF_EN
{
    INTERF_PUSH,
    INTERF_POP,
    INTERF_CLEAR,
    INTERF_FIND_LASTNAME,
    INTERF_SAVE_TO_FILE,
    INTERF_READ_FROM_FILE,
    INTERF_STOP,
    INTERF_TOT
};

char strtab[7][20] =
{
    "0 - push",           //INTERF_PUSH
    "1 - pop",            //INTERF_POP
    "2 - clear",          //INTERF_CLEAR
    "3 - find lastname", //INTERF_FIND_LASTNAME
    "4 - save to file",  //INTERF_SAVE_TO_FILE
    "5 - read from file",//INTERF_READ_FROM_FILE,
    "6 - finish"          //INTERF_STOP
};

void menu()
{
    size_t it;
    for (it = 0; it < INTERF_TOT; ++it)
    {
        printf("%s\n", strtab[it]);
    }
}

int main()
{
    H = (MY_DATA*)NULL, T = (MY_DATA*)NULL;
    size_t op = 0;
    while (op >= INTERF_PUSH && op <= INTERF_STOP)
    {
        menu();
        scanf("%d", &op); getchar();
        switch (op)
        {
            case INTERF_PUSH: push();
                break;
            case INTERF_POP: pop();
                break;
            case INTERF_CLEAR: clear();
                break;
            case INTERF_FIND_LASTNAME: find_lastname();
                break;
            case INTERF_SAVE_TO_FILE: save_to_file();
                break;
            case INTERF_READ_FROM_FILE: read_from_file();
                break;
            case INTERF_STOP: clear();
                return 0;
            default:
                printf("nieuznawany kod operacji\n");
        };
    }
    return 0;
}
```