

Practice 2 - Report / Практична робота 2 - Звіт

Завдання:

Заняття 2

12.09.2023

Завдання 1. Напишіть дві функції для створення списків за даними, введеними з клавіатури.

Перша функція повертає вказівник на голову списку.

Друга – не повертає значення, а формує голову списку за допомогою параметру.

Напишіть функцію виведення на екран елементів списку.

Напишіть головну функцію та протестуйте в ній всі функції.

```
pnnode formLIFO_1 ( ... )          void formLIFO_2 ( ... )
{                                  {
    ...                            ...
}
```

Завдання 2. Поміркуйте, як створити список так, щоб елементи в списку були розміщені в тому ж порядку, що й при введенні з клавіатури, а не в зворотному.

Завдання 3.

Описати структуру *Футболіст* з полями :

Прізвище

Амплуа (тип перелічення)

Вік

Кількість ігор

Кількість голів

Створити **список** структур.

Реалізувати функції:

визначити кращого форварда і вивести інформацію про нього;

вивести відомості про футболістів, які зіграли менше 5-ти ігор.

Скрипт:

```
1 // Завдання 1: Функції для створення списку за даними, введеними з клавіатури.
2 // Функція formLIFO_1 повертає вказівник на голову списку.
3 // Функція formLIFO_2 формує голову списку за допомогою параметру.
4 // Функція printList виводить елементи списку на екран.
5 // Головна функція main протестує всі функції.
6
7 #include <stdio.h>
8 #include <stdlib.h>
9 #include <string.h>
10
11 // Структура для елементів списку
12 struct pnnode {
13     char *name;
14     int goals;
15     struct pnnode *next;
16 };
17
18 // Функція formLIFO_1: створює список за даними, введеними з клавіатури.
19 // Повертає вказівник на голову списку.
20 pnnode *formLIFO_1() {
21     pnnode *head = NULL;
22     pnnode *newNode = NULL;
23     char name[50];
24     int goals;
25
26     while (1) {
27         printf("Enter player name: ");
28         fgets(name, sizeof(name), stdin);
29         if (name[0] == '\n') break;
30         printf("Enter goals scored: ");
31         scanf("%d", &goals);
32         newNode = (pnnode *) malloc(sizeof(pnnode));
33         newNode->name = strdup(name);
34         newNode->goals = goals;
35         newNode->next = head;
36         head = newNode;
37     }
38     return head;
39 }
40
41 // Функція formLIFO_2: формує голову списку за допомогою параметру.
42 void formLIFO_2(pnnode *head) {
43     // Тут можна додати код для формування списку за допомогою параметру.
44 }
45
46 // Функція printList: виводить елементи списку на екран.
47 void printList(pnnode *head) {
48     pnnode *current = head;
49     while (current != NULL) {
50         printf("Player: %s, Goals: %d\n", current->name, current->goals);
51         current = current->next;
52     }
53 }
54
55 // Головна функція main
56 int main() {
57     pnnode *head = formLIFO_1();
58     formLIFO_2(head);
59     printList(head);
60     return 0;
61 }
```

Пояснення:

```
1  #include <iostream>
2  using namespace std;
```

Бібліотека вводу виводу даних в консоль. Простір імен std.

```
4  struct Footballist {
5      string surname = "";
6      string role = "";
7      short age = 0;
8      short games_count = 0;
9      int goal_count = 0;
10 };
```

Запис структури футболістів.

```
11 Footballist SetFootballist() {
12     Footballist footballist;
13     cout << "\nFootballist's surname: ";
14     cin >> (&footballist)->surname;
15     cout << "Footballist's role: ";
16     cin >> (&footballist)->role;
17     cout << "Footballist's age: ";
18     cin >> (&footballist)->age;
19     cout << "Footballist's games count: ";
20     cin >> (&footballist)->games_count;
21     cout << "Footballist's goal count: ";
22     cin >> (&footballist)->goal_count;
23     return footballist;
24 }
```

Функція, яка зчитує інформацію з консолі про нового футболіста, і повертає його.

```
26 struct tnode {
27     Footballist footballist;
28     tnode* next = NULL;
29 };
30 typedef tnode* pnode;
```

Запис списку футболістів.

```
34 pnode formLIFO_1(int n) {
35     pnode head = NULL, cur;
36     Footballist footballist;
37     while (n > 0) {
38         footballist = SetFootballist();
39         n--;
40         cur = new tnode;
41         cur->footballist = footballist;
42         cur->next = head;
43         head = cur;
44     }
45     return head;
46 }
```

```
47 void formLIFO_2(int n, pnode * p) {
48     pnode head = NULL, cur;
49     Footballist footballist;
50     while (n > 0) {
51         footballist = SetFootballist();
52         n--;
53         cur = new tnode;
54         cur->footballist = footballist;
55         cur->next = head;
56         head = cur;
57     }
58     *p = head;
59 }
```

Дві функції для створення списків футболістів за даними, введеними з клавіатури. Створення списків в них здійснюється за принципом LIFO - останній зайшов перший вийшов.

Перша функція повертає вказівник на голову списку.

Друга – не повертає значення, а формує голову списку.

```

63  pnode formFIFO_1(int n) {
64      pnode head = NULL;
65      pnode * cur = &head;
66      Footballist footballist;
67      while (n > 0) {
68          footballist = SetFootballist();
69          n--;
70          *cur = new tnode;
71          (*cur)->footballist = footballist;
72          cur = &(*cur)->next;
73      }
74      return head;
75  }

76  void formFIFO_2(int n, pnode * p) {
77      pnode head = NULL;
78      pnode* cur = &head;
79      Footballist footballist;
80      while (n > 0) {
81          footballist = SetFootballist();
82          n--;
83          *cur = new tnode;
84          (*cur)->footballist = footballist;
85          cur = &(*cur)->next;
86      }
87      *p = head;
88  }

```

Дві функції для створення списків футболістів за даними, введеними з клавіатури. Створення списків в них здійснюється за принципом FIFO - перший зайшов перший вийшов.

Перша функція повертає вказівник на голову списку.

Друга – не повертає значення, а формує голову списку.

```

92  void free(pnode p) {
93      if (p != NULL) {
94          free(p->next);
95          delete p;
96      }
97  }

```

Функція для очищення пам'яті від даного списку.

```

101 void PrintFootballist(pnode footballist) {
102     cout << "surname: " << footballist->footballist.surname
103         << " role: "
104         << footballist->footballist.role << " age: "
105         << footballist->footballist.age << " games count: "
106         << footballist->footballist.games_count <<
107         " goal count: " << footballist->footballist.goal_count << endl;
108 }
109 void PrintAll(pnode footballist) {
110     PrintFootballist(footballist);
111     if (footballist->next != NULL) PrintAll(footballist->next);
112 }

```

Функції для виведення даних про футболістів у консоль.

```

116 pnode BestForward(pnode footballist, pnode bestForward) {
117     if (footballist->footballist.goal_count / footballist->footballist.games_count >
118         bestForward->footballist.goal_count / bestForward->footballist.games_count)
119         bestForward = footballist;
120     if (footballist->next == NULL) return bestForward;
121     else return BestForward(footballist->next, bestForward);
122 }
123 bool isLessThanFiveGames(pnode footballist) {
124     return footballist->footballist.games_count < 5;
125 }
126 void LessThanFiveGames(pnode footballist) {
127     if (isLessThanFiveGames(footballist))
128         PrintFootballist(footballist);
129     if (footballist->next != NULL) LessThanFiveGames(footballist->next);
130 }

```

Функція пошуку кращого форварда та функція виведення футболістів що зіграли менше п'яти разів у консоль. В практичній роботі номер 1 ці функції опрацьовували динамічні масиви футболістів. Тут функції опрацьовують списки футболістів. Складність виконання даних алгоритмів при цьому залишається незмінною.

```

134 int main()
135 {
136     int n = 0;
137     pnode footballist = NULL;
138     int command = 0;
139     while (true) {
140         cout << "0 for help" << endl;
141         cin >> command;
142         if (command == 0) cout << "0 - help\n1 - new list\n2 - continue with existing\n3 - exit" << endl;
143         else if (command == 3) {
144             break;
145         }
146         else {
147             if (command == 1) {
148                 cout << "Enter count: ";
149                 cin >> n;
150                 if (n < 1) {
151                     cout << "n must be bigger than 0" << endl;
152                     continue;
153                 }
154                 cout << "1 - LIFO_1; 2 - LIFO_2; 3 - FIFO_1; 4 - FIFO_2" << endl;
155                 cin >> command;
156                 if (command == 1) footballist = formLIFO_1(n);
157                 else if (command == 2) formLIFO_2(n, &footballist);
158                 else if (command == 3) footballist = formFIFO_1(n);
159                 else if (command == 4) formFIFO_2(n, &footballist);
160                 else cout << "Unknown command" << endl;
161             }
162             else if (command == 2) {
163                 int i = 0;
164                 cout << "0 for help" << endl;
165                 while (true) {
166                     cin >> command;
167                     if (command == 0) cout << "0 - help\n1 - find best forward\n2 - show all less than 5 games footballist\n3 - print all\n4 - go back" << endl;
168                     else if (command == 1) PrintFootballist(BestForward(footballist, footballist));
169                     else if (command == 2) LessThanFiveGames(footballist);
170                     else if (command == 3) PrintAll(footballist);
171                     else if (command == 4) break;
172                     else cout << "Unknown command" << endl;
173                 }
174             }
175             else cout << "Unknown command" << endl;
176         }
177     }
178     free(footballist);
179 }
180

```

Головна функція. Ця функція починає роботу програми та надає користувачу можливість викликати функції даного коду. Тобто користувач може створювати новий список 2-ма способами і працювати з ним.