**Описание структур данных**

Хранение данных программы осуществляется в форме линейного однонаправленного списка. Каждый элемент списка представляет собой структуру (см. рис. 3), состоящую из полей с данными, а также указателя на соседний (следующий) элемент списка.

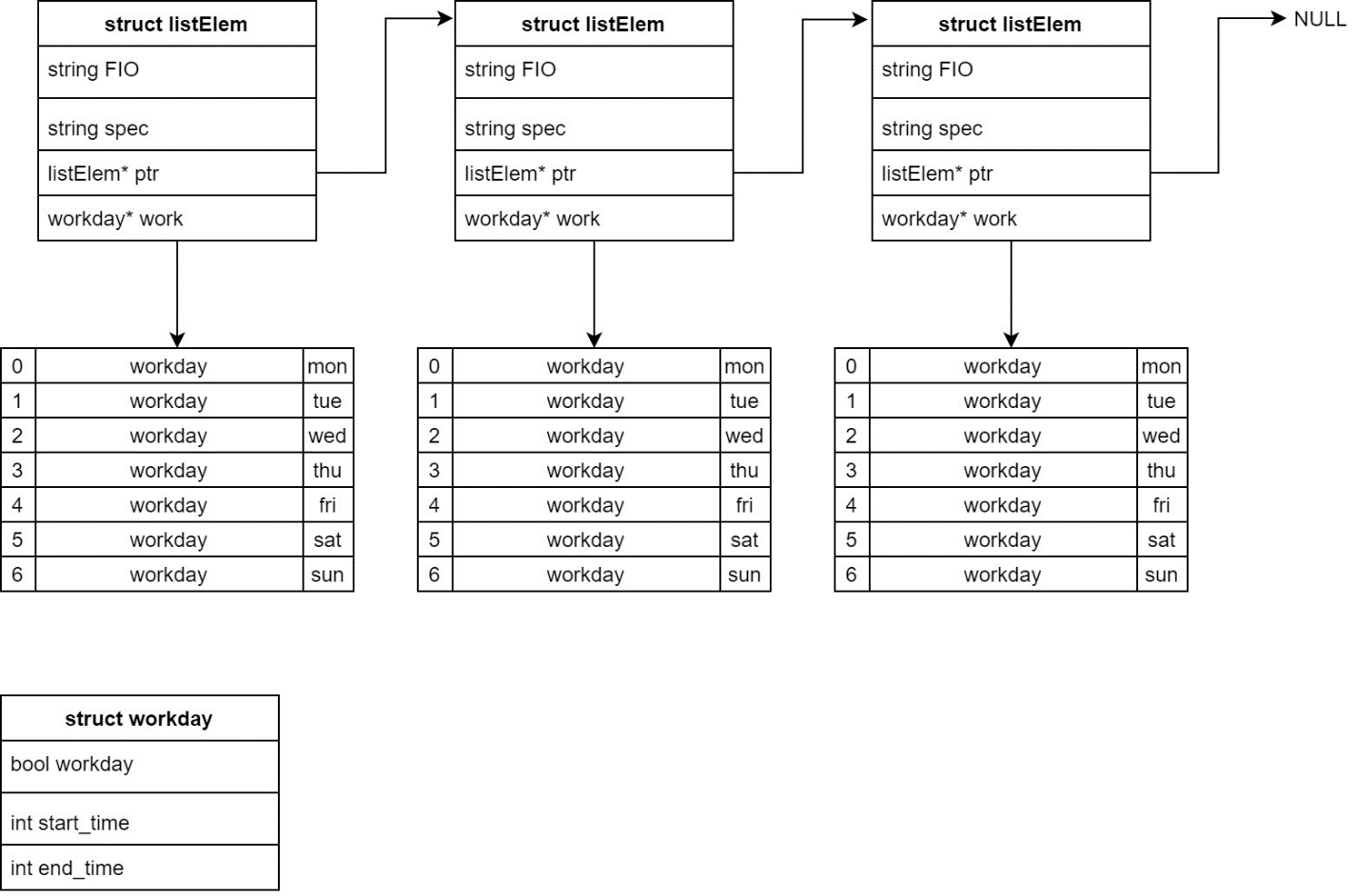


Рис. 3 - Структура хранения данных

Структуры

Врач listElem представлен в памяти программы в виде структуры с следующими полями:

string FIO, spec - ФИО и специализация

workday\* work - Указатель на массив рабочих дней

listElem\* ptr=NULL - Указатель на следующий элемент

Структура для хранения данных по рабочему дню workday содержит поля:

bool workday - Рабочий ли день

int start\_time, end\_time - Время начала и конца рабочего дня

**Описание пользовательских функций**

Модуль functions

**listElem\* init**

Назначение: Функция инициализации 0-го узла списка.

Параметры: **string FIO, string spec, workday\* work**

Принимает ФИО врача, специальность и массив рабочих дней.

Возвращаемое значение: listElem\* - указатель на начальный элемент списка

**add\_new\_item**

Назначение: Функция добавления нового узла в список

Параметры: **listElem\* root, string FIO, string spec, workday\* work**

Принимает указатель на начальный элемент списка, ФИО врача, специальность и массив рабочих дней.

Возвращаемое значение: -

**listprint**

Назначение: Функция поэлементного вывода на экран всего списка.

Параметры: **listElem lst**

Принимает указатель на начальный элемент списка.

Возвращаемое значение: -

**Get\_new\_item\_from\_keyboard**

Назначение: Функция для ввода нового элемента с клавиатуры

Параметры: **listElem\*\* root**

Принимает указатель на указатель на начальный элемент списка.

Возвращаемое значение: -

**save**

Назначение: Функция для сохранения списка в файл

Параметры: **listElem\* root**

Принимает указатель на начальный элемент списка.

Возвращаемое значение: -

**read**

Назначение: Функция для чтения списка из файла

Параметры: **listElem\*\* root**

Принимает указатель на указатель на начальный элемент списка.

Возвращаемое значение: -

**Алгоритм программы**

Модуль functions

#include "Header.h"

listElem\* init(string FIO, string spec, workday\* work)

listElem\* elem

elem = new listElem

elem->FIO = FIO

elem->work = work

elem->spec = spec

elem->ptr = NULL

вернуть elem

void add\_new\_item(listElem\* root, string FIO, string spec, workday\* work)

listElem \*temp,\*p

temp = new listElem

Пока (root->ptr != NULL)

root = root->ptr

p = root->ptr

root->ptr = temp

temp->FIO = FIO

temp->work = work

temp->spec = spec

temp->ptr = p

listElem\* delete\_item(listElem\* elem, listElem\* root)

listElem\* temp

Если (elem != root)

temp = root

Пока (temp->ptr != elem)

temp = temp->ptr

temp->ptr = elem->ptr

Иначе

temp = root->ptr

root = root->ptr

delete elem

вернуть temp

void listprint(listElem\* root)

listElem\* p

p = root

Пока (p != NULL)

вывод << p->FIO << " " << p->spec << endl

Цикл по i от 0 до 7

вывод << "В день недели " << i << " "

Если (p->work[i].workday)

вывод << "рабочий день от " << p->work[i].start\_time << " до " << p->work[i].end\_time << endl

Иначе

вывод << "выходной" << endl

p = p->ptr

void Get\_new\_item\_from\_keyboard(listElem\*\* root)

string FIO, spec

workday \*work

ввод.clear()

ввод.ignore(numeric\_limits<streamsize>::max(), '\n')

вывод << endl << "ФИО врача:"

getline(ввод, FIO)

вывод << endl << "Специальность:"

getline(ввод, spec)

work = new workday[7]

Цикл по i от 0 до 7

вывод << "Во сколько начинает работать врач в " << i << "-й день недели? ('-1' - если выходной):" << endl

ввод >> work[i].start\_time

Если (work[i].start\_time == -1)

work[i].workday = false

Иначе

work[i].workday = true

вывод << "Во сколько заканчивает работать врач в " << i << "-й день недели?" << endl

ввод >> work[i].end\_time

Если (\*root == NULL)

\*root=init(FIO, spec, work)

Иначе

add\_new\_item(\*root, FIO, spec, work)

void save(listElem\* root)

string output\_file\_name

вывод << "output filename:"

ввод >> output\_file\_name

ofstream out(output\_file\_name)

Если (!out)

вывод << "Can't open file"

вернуть

listElem\* p

p = root

Пока (p != NULL)

Вывод в файл out<< p->FIO << endl << p->spec << endl

Цикл по i от 0 до 7

Если (p->work[i].workday)

Вывод в файл out<< p->work[i].start\_time << endl << p->work[i].end\_time << endl

Иначе

Вывод в файл out<< "-1" << endl

p = p->ptr

out.close()

void read(listElem\*\* root)

string input\_file\_name,FIO,spec

workday\* work

int starttime

int i = 0, k

вывод << "input filename:"

ввод >> input\_file\_name

Если stream fs(input\_file\_name)

Если (!fs)

вывод << "Can't open file"

вернуть

Пока (fs.good())

Если (getline(fs, FIO))

Если (getline(fs, spec))

work = new workday[7]

Цикл по i от 0 до 7

fs >> work[i].start\_time

Если (work[i].start\_time != -1)

work[i].workday = true

fs >> work[i].end\_time

Иначе

work[i].workday = false

Если (\*root == NULL)

\*root = init(FIO, spec, work)

Иначе

add\_new\_item(\*root, FIO, spec, work)

getline(fs, input\_file\_name)

fs.close()

main

listElem\* root=NULL

char ch

SetConsoleCP(1251)

SetConsoleOutputCP(1251)

Пока (true)

вывод << "1 - Добавить новый элемент с клавиатуры"<<endl << "2 - Вывод на экран" << endl << "3 - загрузить из файла" << endl <<"4 - сохранить в файл"<<endl

Если (!ввод.get(ch)) Прервать

Переключатель по (ch):

В случае '1':

Get\_new\_item\_from\_keyboard(&root)

Прервать

В случае '2':

listprint(root)

Прервать

В случае '3':

read(&root)

Прервать

В случае '4':

save(root)

Прервать

ввод.clear()

ввод.ignore(numeric\_limits<streamsize>::max(), '\n')

вернуть 0