БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет ЗВиДО

Специальность ПОИТ

Лабораторная работа № 1

по дисциплине «Основы алгоритмизации и программирования»

часть 2

Вариант № 7

Выполнила: Н. Г. Карпеко

Договор № 696 от 16.05.2017г.

Минск 2018

Тема задания

Разработка, отладка и выполнение программы с использованием подпрограмм с различными типами параметров, типов запись, множество, файлы в среде Турбо/Борланд Паскаль

1. Задача

Дан список тарифов I-net провайдера.

Надо организовать калькулятор тарифов. Предусмотреть возможность добавлять, удалять и корректировать записи из списка тарифов, просматривать весь список.

Информацию в динамический список загружать из типизированного файла.

1. Программа

**program** CalcTarif;

**uses** crt; //Для использования readkey и clrscr

**type**

tarif = **record** //элемент списка описываем через тип "запись"

tar: string[20]; //Название тарифа

inter\_traf: integer; //интернет-трафик

min: integer; //количество минут

plata: integer; //абонентская плата

**end**;

adr = ^zveno; //указатель на элемент списка тарифов

zveno = **record**

infa: tarif;

next: adr; //ссылка на следующий элемент списка

**end**;

//===== standart ================

**procedure** standart;

**begin**

window(0, 0, 24, 79);

textbackground(White);

textcolor(blue);

clrscr;

gotoxy(1, 3);

**end**;

**procedure** menu;

**forward**;

**procedure** enter;

**begin**

writeln;

writeln(' Чтобы вернуться в меню, нажмите Ввод (Enter). ');

readln;

readkey;

**end**;

//=== Bставкa нового тарифа в список =========

**procedure** add(**var** spis: adr; tar: string[10]; inter\_traf, min, plata: integer);

**var**

tmp: adr; // дополнительный указатель

**begin**

standart;

**new**(tmp);

write(' Введите имя добавляемого тарифа: '); read(tar);

tmp^.infa.tar := tar;

write(' К-во интернет-трафика (2/4/8/16/32 Гб): '); read(inter\_traf);

tmp^.infa.inter\_traf := inter\_traf;

write(' Количество бесплатных минут в месяц(50/100/150): '); read(min);

tmp^.infa.min := min;

write(' Абонентская плата (5/10/20/30 руб): '); read(plata);

tmp^.infa.plata := plata;

**if** spis = nil **then** // если список пуст - первый элемент нулевой (1й вариант)

**begin**

**new**(spis); //создаём его первый элемент

tmp := spis; // Ставим указатель на 1й элемент (новый элемент = голова)

**end**

**else** //если список не пуст (2й вариант)

**begin**

tmp := spis; // указатель ставим на голову

**while** tmp^.next <> nil **do** // пока указатель не покажет в пустоту

tmp := tmp^.next; //ставим указатель tmp на последний элемент списка

**new**(tmp^.next); //создаём следующий элемент

tmp := tmp^.next; //переносим tmp на новый элемент

**end**;

tmp^.infa.tar := tar; //заносим значение – наименование тарифа

tmp^.infa.inter\_traf := inter\_traf; //– количество интернет-трафика

tmp^.infa.min := min; // – количество минут

tmp^.infa. plata := plata; // – абонентская плата

tmp^.next := nil; //зануляем указатель

enter;

**end**;

//========= Просмотр списка ============

**procedure** print(spis: adr);

**begin**

standart;

**if** spis = nil **then**

**begin**

write(' Список тарифных планов пуст.'); readln; **exit**;

**end**;

**while** spis <> nil **do** //пока указатель spis не будет указывать в пустоту

**begin** //вывод текущего элемента списка

writeln(' Параметры тарифного плана:');

writeln(' Название тарифа: ', spis^.infa.tar);

writeln(' Количество интернет-трафика: ', spis^.infa.inter\_traf);

writeln(' Количество минут в месяц: ', spis^.infa.min);

writeln(' Абонентская плата: ', spis^.infa. plata);

writeln;

spis := spis^.next; //Переход к следующему элементу

**end**;

enter;

**end**;

//========== удаление списка ==============

**procedure** del\_spisok(**var** spis: adr);

**var**

tmp: adr;

c: char;

**begin**

standart;

writeln(' Удалить весь список Y/N?'); readln(c);

**if** c **in** ['Y', 'y'] **then**

**begin**

**while** spis <> nil **do**

**begin**

tmp := spis;

spis := spis^.next;

dispose(tmp);

**end**;

writeln(' Список удален.');

**end**

**else**

writeln(' Список не был удален.');

enter;

**end**;

// ========= удаление элемента =======

**procedure** del\_elem(**var** spis: adr; tmp: adr);

**var**

tmp1: adr;

**begin**

**if** (spis = nil) **or** (tmp = nil) **then exit**; //если список пуст или указатель никуда не указывает, то выходим (1й вариант)

**if** tmp = spis **then** //если удаляемый элемент - вершина списка (голова) (2й вариант)

**begin**

spis := tmp^.next; // переносим вершину и

dispose(tmp); // освобождаем память из элемента

**end**

**else** // неск-ко эл-ов и надо найти нужный (3й вариант)

**begin**

tmp1 := spis; //ставим допольнительный указатель на вершину списка

**while** tmp1^.next <> tmp **do** //доходим до элемента стоящего перед удаляемым

tmp1 := tmp1^.next; //передвигаем указатель

tmp1^.next := tmp^.next; //указатель элемента переносим на следующий элемент за удаляемым

dispose(tmp); //удаляем элемент

**end**;

**end**;

// ====== Функция поиска элемента ===============

**function** search\_elem(spis: adr; tar: string[20]): adr;

**begin**

standart;

**if** spis <> nil **then** //если список не пуст

**while** (spis <> nil) **and** (tar <> spis^.infa.tar) **do** // пока spis (Указатель) не укажет в "пустоту" или пока не нашли нужный элемент

spis := spis^.next; // перемещаем указатель на следующий элемент

search\_elem := spis; //ф-ция возвращает указатель на найденный элемент или nil, если его нет

enter;

**end**;

//=== удалениe элемента из списка по значению =========

**procedure** delete\_el\_znach(**var** spis: adr; tar: string[20]);

**var**

tmp: adr;// дополнительный указатель

**begin**

standart;

**if** spis = nil **then** //если список пуст, то выводим сообщение и выход (1й вариант)

**begin**

write(' Список тарифных планов пуст.'); readln ;**exit**;

**end**;

write(' Введите наименование удаляемого элемента: '); readln(tar); writeln;

tmp := search\_elem(spis, tar); //указываем на удаляемый элемент (вызов функции поиска элемента) (2й вариант)

**if** tmp = nil **then** //если элемент не найден, то - сообщение и выход (3й вариант)

**begin**

write(' Искомый элемент ', tar, ' отсутствует в списке'); readln; **exit**;

**end**

**else**

**begin**

del\_elem(spis, tmp); // удаление элемент (вызов процедуры удаления элемента)

writeln(' Элемент ', tar, ' удален');

**end**;

enter;

**end**;

// ========= подбор тарифа ====================

**procedure** podbor(spis: adr; tar: string[20]; inter\_traf, min, plata: integer);

**begin**

standart;

**if** spis = nil **then** write(' Список тарифных планов отсутствует.');

writeln(' Введите характеристики тарифного плана: ');

writeln(' Количество интернет-трафика (2/4/8/16/32 Гб): '); readln(inter\_traf);

writeln(' Количество минут в месяц (50/100/150): '); readln(min);

**while** (spis <> nil) **and** (spis^.infa.inter\_traf <> inter\_traf) **and** (spis^.infa.min <> min) **do**

spis := spis^.next;

**if** spis <> nil **then begin**;

writeln;

writeln(' Вам подходит тарифный план: ', spis^.infa.tar);

writeln(' Параметры тарифного плана: ');

writeln(' Название тарифа: ', spis^.infa.tar);

writeln(' Количество интернет-трафика: ', spis^.infa.inter\_traf);

writeln(' Количество минут в месяц: ', spis^.infa.min);

writeln(' Абонентская плата: ', spis^.infa. plata);

**end**

**else** writeln(' Подходящего тарифного плана нет.');

enter;

**end**;

//==== запись списка в файл ====================

**procedure** save(**var** spis: adr; f: **file of** tarif);

**var**

tek\_elem: adr; //указатель текущего элемента

n: word;

name: string[11];

**begin**

standart;

n := 0; // - Количество записанных элементов

**if** spis = nil **then** //1й вар-т

**begin**

writeln(' Список пуст'); writeln;

writeln(' Чтобы вернуться в меню, нажмите Ввод (Enter). ');

readln;

**exit**;

**end**

**else** //2й вар-т

**begin**

writeln(' Идет сохранение... ');

write(' Введите имя файла для сохранения данных (до 11 символов): '); readln(name);

assign(f, name);

rewrite(f); //открываем файл для записи

tek\_elem := spis;

**while** tek\_elem <> nil **do**

**begin**// - включаем весь список

write(f, tek\_elem^.infa); // - записываем текущее значение

tek\_elem := tek\_elem^.next; // - перемещаем указатель по списку

n := n + 1; // наращиваем количество записанных элементов

**end**;

close(f); //закрываем файл

writeln;

writeln(' Сохранение завершено. ');

writeln(' Имя файла: ', name); //выводим сообщение о сохранении

writeln(' Количество записанных элементов: ', n);

**end**;

enter;

**end**;

//==== Загрузка списка из файла =======================

**procedure** load(**var** spis: adr; **var** f: **file of** tarif);

**var**

vsp: adr;

name: string[11];

**begin**

standart;

clrscr;

write(' Введите имя файла c данными: '); readln(name);

assign(f, name); //подключились к файлу

**if** FileExists(name) **then** //exists ( FileExists() - прцедура проверки наличия файла)

**begin**

reset(f); //открываем файл

**new**(vsp); //создаем новый элемент списка

vsp^.next := nil; //зануляем указатель

spis := vsp;

**while not** Eof(f) **do** //считываем все данные до конца файла

**begin**

spis := spis^.next;

read(f, spis^.infa); //читаем

//write(spis^.infa); // вывод данных на экран

spis^.next := nil;

**end**;

**end**

**Else**

**begin**

rewrite(f);

**end**;

enter;

**end**;

// ======= МЕНЮ ===============

**procedure** menu;

**var**

f: **file of** tarif;

spis: adr;

tar: string[20]; //Название тарифа

inter\_traf: integer; //интернет-трафик

min: integer; //количество минут

plata: integer; //абонентская плата

key: char;

**begin**

**repeat**//выбор пункта меню

clrscr;

window(20, 8, 60, 25); textbackground(brown); textcolor(lightgray);

clrscr; gotoxy(10, 2);

Writeln(' МЕНЮ: '); gotoxy(3, 3);

Writeln('1. Bставкa нового тарифа в список.'); gotoxy(3, 4);

Writeln('2. Просмотр тарифов.'); gotoxy(3, 5);

Writeln('3. Удаление всего списка.'); gotoxy(3, 6);

Writeln('4. Удаление тарифного плана по наименованию.'); gotoxy(3, 7);

Writeln('5. Подбор тарифного плана.'); gotoxy(3, 8);

Writeln('6. Запись списка в файл.'); gotoxy(3, 9);

Writeln('7. Запись списка из файла.'); gotoxy(3, 10);

Writeln('ESC - выход.');

key := readkey;

**case** key **of** //выбор пункта меню

'1': add(spis, tar, inter\_traf, min, plata); // 1. Bставкa нового тарифа в список.

'2': print(spis); // 2. Просмотр тарифов.

'3': del\_spisok(spis); // 3. Удаление всего списка.

'4': delete\_el\_znach(spis, tar); // 5. Удаление тарифного плана по наименованию.

'5': podbor(spis, tar, inter\_traf, min, plata); // 8.Подбор тарифного плана.

'6': save(spis, f); // 6. Запись списка в файл.

'7': load(spis, f); // 7. Запись списка из файла.

**End**;

**until** key = #27; //- код клавиши Esc - выходим из цикла и из программы

**end**;

//===== ОСНОВНАЯ ПРОГРАММА ============================

**VAR**

f: **file of** tarif;

spis: adr;

tar: string[20];

inter\_traf: integer;

min: integer;

plata: integer;

key: char;

**BEGIN**

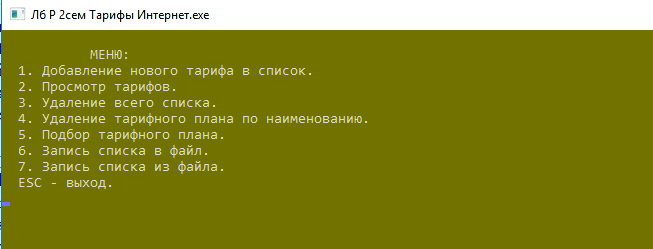
standart;

menu;

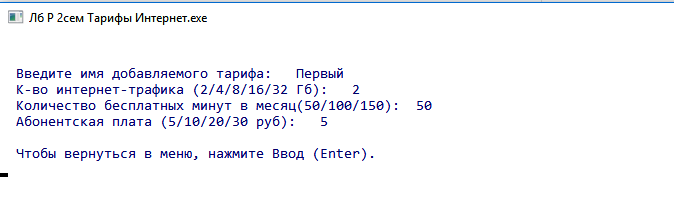
readln;

**END**.

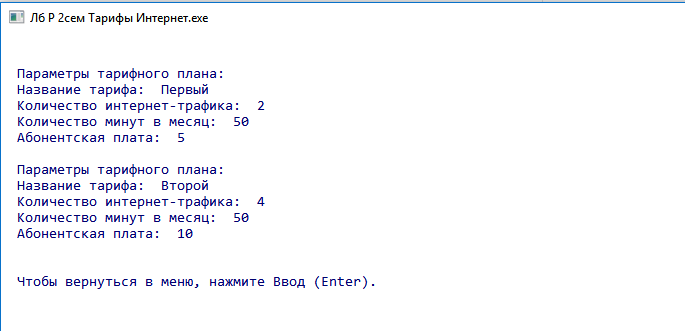
1. **Тестирование программы**

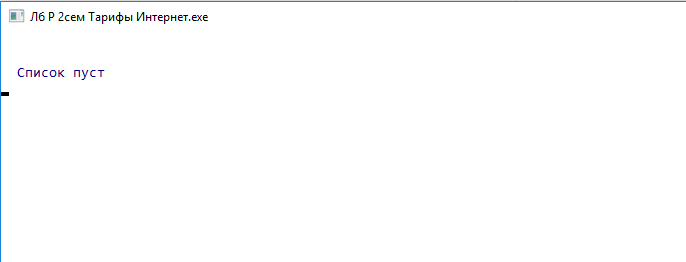


1. Добавление нового тарифа в список

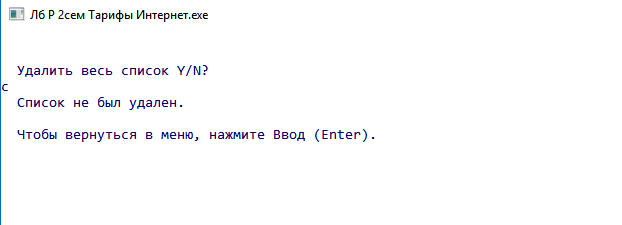


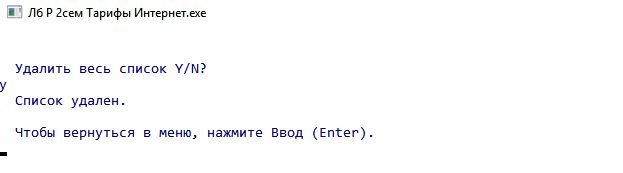
2. Просмотр тарифов



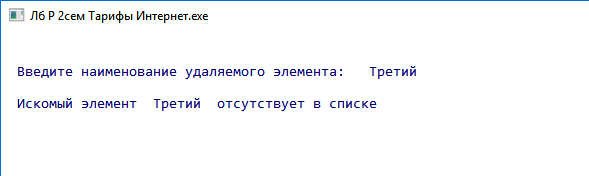


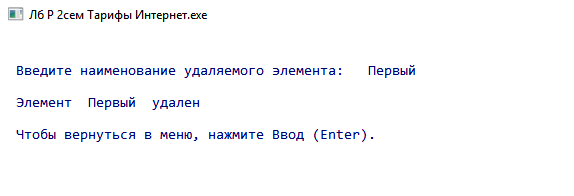
3. Удаление всего списка

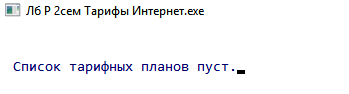




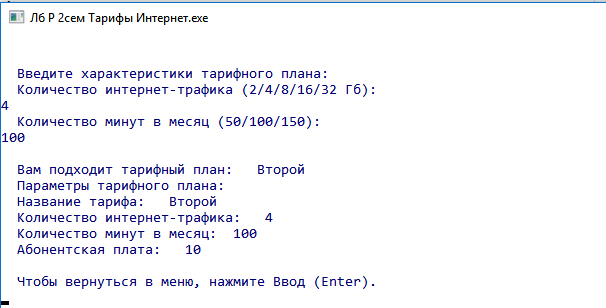
4. Удаление тарифного плана по наименованию







5. Подбор тарифного плана



6. Запись списка в файл

