МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**О Т Ч Е Т**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №13**

**«*B+ДЕРЕВЬЯ*»**

**по курсу:**

***"ФУНКЦИОНАЛЬНОЕ И ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ "***

|  |  |
| --- | --- |
| Выполнила: | студент группы10701118  Воробей И.А. |
| Проверил: | Доцент  Ковальков А. Т. |

Минск 2019**Лабораторная работа № 13**

**«B+ДЕРЕВЬЯ»**

***Цель работы:*** приобретение практических навыков составления и отладки программ, работающих с B+деревьями.

***Постановка задачи:***

Создать внешнюю базу данных на диске следующего формата:

student( Номер\_зачетки, Фамилия, Средний\_балл).

В этой БД по каждому из полей сформировать В+дерево.

Требуется:

а) вывести фамилии студентов в порядке возрастания номеров зачетной книжки;

б) вывести фамилии студентов в алфавитном порядке;

в) вывести фамилии студентов в порядке убывания среднего балла.

В меню предусмотреть операции создания внешней БД, извлечения из БД нужной информации в соответствии с заданием, дополнения БД новыми термами, удаления термов из БД, просмотра содержимого отдельных В+деревьев.

Выбрать из базы данных фамилии (вывод на экран по одной в строке), заканчивающихся на заданную букву.

***Код программы:***

% 5. Выбрать из базы данных фамилии (вывод на экран по одной в строке), заканчивающиеся на заданную букву

%C:\Users\Asus\AppData\Local\Temp

%floss

domains

db\_selector = data\_file

student=stdinfo(integer,string,integer)

predicates

nondeterm lastsym(string,string)

nondeterm repeat

nondeterm main\_menu

nondeterm process(integer) %(i)

print\_first

nondeterm looktree(integer) %(i)

addNew

open(bt\_selector,bt\_selector,bt\_selector) %(0,0,0)

close(bt\_selector,bt\_selector,bt\_selector) %(i,i,i)

nondeterm delete\_term(student) %(i)

nondeterm createDB(string) %(i)

nondeterm openDB(string) %(i)

nondeterm show\_bt(bt\_selector) %(i)

nondeterm up\_number(bt\_selector) %(i)

nondeterm down\_average(bt\_selector) %(i)

nondeterm search(char,ref) %(i,i)

clauses

lastsym(S,C):-frontstr(1,S,C,R), R="",!.

lastsym(S,C):-frontstr(1,S,\_,Q), lastsym(Q,C).

repeat.

repeat:-repeat.

main\_menu:-repeat,

write(">>--------------------------------------<<"),nl,

write(" 1:> Создать БД"),nl,

write(" 2:> Открыть БД"),nl,

write(" 3:> Просмотреть список"),nl,

write(" 4:> Добавить запись"),nl,

write(" 5:> Удалить зпись"),nl,

write(" 6:> Просмотр содержимого В+ дерева"),nl,

write(" 7:> Вывод фамилий студентов в порядке возрастания номеров зачетной книжки"),nl,

write(" 8:> Вывод фамилий студентов в алфавитном порядке"),nl,

write(" 9:> Вывод фамилий студентов в порядке убывания среднего балла"),nl,

write(" 10:>Выбрать из базы данных фамилии, начинающихся на заданную букву"),nl,

write(" 0:> Выход"),nl,

write(">>--------------------------------------<<"),nl,

write("---> "),readint(Choice),Choice<11,Choice>=0,process(Choice),Choice=0,!.

%выход из программы

process(0):-db\_close(data\_file),

write("Press any key to exit."),

readChar(\_),

exit.

%создание БД

process(1):-write("Введите имя создающейся БД > "),

readln(Name),

concat(Name,".dba",File), %соединение с расширением dba

createDB(File). %создание БД

%открыть БД

process(2):-db\_close(data\_file), %закрытие текущей БД

write("Введите имя БД > "),

readln(Name),

concat(Name,".dba",File), %соединение с расширением dba

openDB(File). %открытие БД

%печать списка

process(3):-write("-------------->Группа 10701118<--------------\n"),

write(" Номер зачетки ; Фамилия ; Средния балл \n"),

print\_first.

%добавить запись

process(4):-addNew,

write("Информация добавлена "),nl.

%удалить запись

process(5):-write("Введите данные для удаления\n"),

write("Фамилия :> "),readln(F),

delete\_term(stdinfo(\_,F,\_));

write("Удалено!\n").

%просмотр одной части дерева

process(6):-write("Выберите дерево (1,2,3) :> "),

readint(Ch),

looktree(Ch).

%порядок возрастания зачеток

process(7):-bt\_open(data\_file,"CardNumber",Bt\_Num), %открытие ветви номеров зачеток

key\_first(data\_file,Bt\_Num,\_), %установка указателя на первый элемент

up\_number(Bt\_Num),

bt\_close(data\_file, Bt\_Num). %закрытие ветви

%по алфавиту

process(8):-bt\_open(data\_file,"Fam",Bt\_Fam), %открытие ветви фамилий

key\_first(data\_file,Bt\_Fam,\_), %установка указателя на первый элемент

up\_number(Bt\_Fam),

bt\_close(data\_file, Bt\_Fam). %закрытие ветви

%убывание среднего балла

process(9):-bt\_open(data\_file,"Average",Bt\_Av), %открытие ветви оценок

key\_last(data\_file,Bt\_Av,\_), %установка указателя на последний элемент

down\_average(Bt\_Av),

bt\_close(data\_file, Bt\_Av).

%вывод студентов чья фамилия начинается на опред.букву

process(10):-write("Введите букву :> "),

readchar(Char),write(Char),nl,

db\_chains(data\_file,Chain), %поочередно свзяываем Сhain с іменамі цепочкі базы данных

chain\_first(data\_file,Chain,Ref), %установка на 1ую ветвь

search(Char,Ref).

%открытие цепей дерева

open(Bt\_Num,Bt\_Fam,Bt\_Av):-bt\_open(data\_file,"CardNumber",Bt\_Num),%открытие цепи зачетки

bt\_open(data\_file,"Fam",Bt\_Fam), %открытие цепи фамилий

bt\_open(data\_file,"Average",Bt\_Av). %открытие цепи среднего балла

%закрытие цепей дерева

close(Bt\_Num,Bt\_Fam,Bt\_Av):-bt\_close(data\_file,Bt\_Num),

bt\_close(data\_file,Bt\_Fam),

bt\_close(data\_file,Bt\_Av).

%создание Базы

createDB(File):-existfile(File),

write("Файл с таким именем уже существует!\n"),

readchar(\_),!.

createDB(File):-db\_create(data\_file,File,in\_file),

bt\_create(data\_file,"CardNumber",Bt\_Num,10,2),

bt\_create(data\_file,"Fam",Bt\_Fam,10,2),

bt\_create(data\_file,"Average",Bt\_Av,10,2),

close(Bt\_Num,Bt\_Fam,Bt\_Av),

write("Новая база данных создана и открыта для записи\n"),

readchar(\_).

%открытие Базы

openDB(File):-existfile(File),

db\_open(data\_file,File,in\_file),

write("База открыта.\n"),

readchar(\_),!.

openDB(\_):-write("Файл не существует!\n"),

readchar(\_).

%печать списка

print\_first:-chain\_terms(data\_file,gr10701118,student,stdinfo(N,F,Av),\_),%последовательный просмотр цепочки

write(N," ",F," ",Av),nl,fail;true.

%просмотр определнной ветки

looktree(1):-bt\_open(data\_file,"CardNumber",Bt\_Num), %открытие ветки

show\_bt(Bt\_num), %отображение ветки

bt\_close(data\_file,Bt\_num). %закрытие ветки

looktree(2):-bt\_open(data\_file,"Fam",Bt\_fam),

show\_bt(Bt\_fam),

bt\_close(data\_file,Bt\_fam).

looktree(3):-bt\_open(data\_file,"Average",Bt\_av),

show\_bt(Bt\_av),

bt\_close(data\_file,Bt\_av).

%вывод ветві

show\_bt(Bt):-

key\_current(data\_file,Bt,Key,\_),

write(Key),nl,fail.

show\_bt(Bt):-

key\_next(data\_file,Bt,\_),

show\_bt(Bt).

show\_bt(\_).

%сортировка по возрастанию/алфавиту

up\_number(Bt):-

key\_current(data\_file,Bt,\_,Ref),%возвращает ключ с указателем на текущий элемент

ref\_term(data\_file,student,Ref,stdinfo(Num,Fam,Av)), %связывает Ref c stdinfo

write(Num," ",Fam," ",Av),nl,fail.

up\_number(Bt):-

key\_next(data\_file,Bt,\_),%перевод указателя на следующий элемент

up\_number(Bt).

up\_number(\_).

%сортировка по убыванию (функционал аналогичен сортировке по возрастанию)

down\_average(Bt):-

key\_current(data\_file,Bt,\_,Ref),

ref\_term(data\_file,student,Ref,stdinfo(Num,Fam,Av)),

write(Num," ",Fam," ",Av),nl,fail.

down\_average(Bt):-

key\_prev(data\_file,Bt,\_),

down\_average(Bt).

down\_average(\_).

%добавление элемента

addNew:-write("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"),

write("Номер зачетки > "),readint(N),

write("Фамилия студента > "),readln(F),

write("Средний балл > "),readint(Av),

chain\_insertz(data\_file,gr10701118,student,stdinfo(N,F,Av),Ref), %запись элемента

open(Bt\_Num,Bt\_Fam,Bt\_Av),

str\_int(Name,N),key\_insert(data\_file,Bt\_Num,Name,Ref), %добавление в дерево

key\_insert(data\_file,Bt\_Fam,F,Ref), %добавление в дерево

str\_int(Aver,Av),key\_insert(data\_file,Bt\_Av,Aver,Ref), %добавление в дерево

close(Bt\_Num,Bt\_Fam,Bt\_Av).

%удаление элемента из дерева

delete\_term(stdinfo(N,F,Av)):-chain\_terms(data\_file,gr10701118,student,stdinfo(N,F,Av),Ref), %последовательный проход по дереву

open(Bt\_Num,Bt\_Fam,Bt\_Av),%открытие опред. цепи

str\_int(Name,N),%получаем номер от имени

key\_delete(data\_file,Bt\_Num,Name,Ref), %удаляем из цепи зачеток

key\_delete(data\_file,Bt\_Fam,F,Ref), %удаляем из цепи фамилий

str\_int(Aver,Av),%номер среднего балла

key\_delete(data\_file,Bt\_Av,Aver,Ref), %удаляем из цепи среднего балла

close(Bt\_Num,Bt\_Fam,Bt\_Av), %закрытие цепи

term\_delete(data\_file,gr10701118,Ref),fail. %удалчем из базы

%поиск совпадений первой буквы фамилии и введенной буквы

search(Char,Ref):-ref\_term(data\_file,student,Ref,stdinfo(N,F,Av)), %связывает Ref c stdinfo

%frontchar(F,NeedChar,\_), %выделяем первый символ фамилии

lastsym(F,NeedChar), %выделяем первый символ фамилии

str\_char(C1,Char),

NeedChar = C1,

writef("Номер зачетки: %d\nФамилия: %s\nСредний балл: %d\n",N,F,Av),fail.

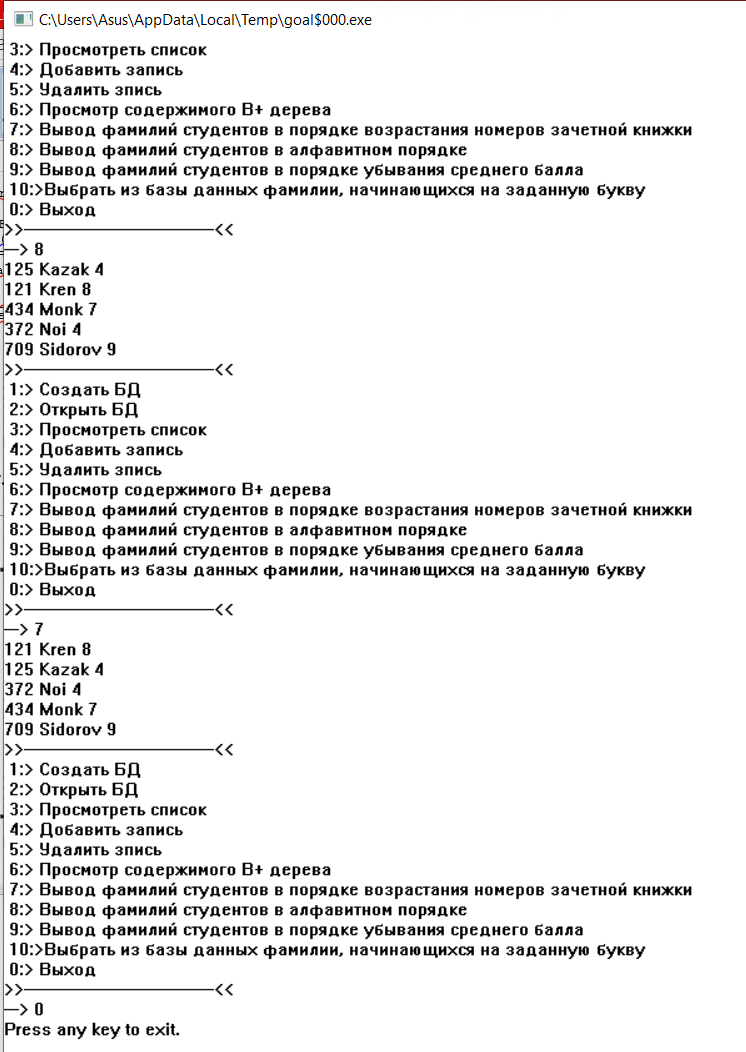
search(Char,Ref):-chain\_next(data\_file,Ref,Next),%переход на следующую запісь

search(Char,Next).

goal

main\_menu.

***Результаты работы программы:***



**Контрольные вопросы:**

1. *Что такое В+деревья, где и для чего они используются?*

В+дерево является структурой данных, которую можно применять для очень эффективного метода сортировки большого количества данных. В+деревья находятся во внешней базе данных.

1. *Отличие В+дерева от двоичного (бинарного) дерева.*

В+ дерево поддерживает структуру, упрощающую поиск и сортировку данных, в связи с чем эти операции, по сравнению с двоичными деревьями, выполняются быстрее.

1. *Механизм упорядочивания данных в двоичном дереве.*

Каждая вершина дерева имеет 2 вершины ниже себя, если вершина не является листом. Вершины содержат ссылочные номера данных для ключа, который устанавливается программистом. данном ДД упорядочены слева направо, то есть для каждой вершины все значения вершин левого поддерева меньше значений родительской вершины. А все значения правого поддерева – больше. В свою очередь оба поддерева упорядочены таким же образом. Преимущество упорядочивания состоит в том, что для поиска некоторого объекта в ДД всегда достаточно просмотреть не более одного поддерева

1. *Что представляет собой ключ В+дерева ?*

Ключ - информационное поле записи, по которому построено В+дерево. При вводе новой записи во внешнюю БД Пролог включает ключ и указатель, соответствующие этой записи, в В+дерево.

1. *Страница, длина и порядок В+дерева.*

В+дерево разбито на отдельные страницы или узлы. Каждый узел В+дерева является либо последним (лист), либо порождает два нижележащих узла. Каждый узел содержит группу ключей из заданного диапазона. По значению ключа можно быстро проверить, не находится ли искомый ключ внутри диапазона конкретного узла.

Длина *В+дерева -* длина ключа, в пределах которой будет производиться упорядочивание ключей при построении дерева.

Порядок *В+дерева -* порядок дерева, определяющий, сколько ключей запоминается в каждом узле В+дерева. Для баз данных среднего размера рекомендуется Пор=4.

1. *Как в программе создается В+дерево ?*

В+дерево **создается** с помощью предиката

bt\_create(ИмяБД, ИмяВ+дер, ПерВ+дер, ДлКл, Пор)(db\_selector, string, bt\_selector, integer, integer): (i,i,o,i,i)

1. *Какие стандартные предикаты используются для открытия, закрытия и удаления В+дерева ?*

Стандаотные предикаты для открытия, закрытия и удаления В+дерева*:*

bt\_open(ИмяБД, ИмяВ+дер, ПерВ+дер)(db\_selector,string,bt\_selector):(i,i,o)

bt\_close(ИмяБД, ПерВ+дер)(db\_selector, bt\_selector):(i,i)

bt\_delete(ИмяБД, ИмяВ+дер)(db\_selector,bt\_selector):(i,i)

1. *Как можно вставить новый ключ в В+дерево ?*

Вставка нового ключа в В+дерево осуществляется предикатом

key\_insert(ИмяБД,ПерВ+дер,Ключ,Ссылка)(db\_selector, bt\_selector, string, ref):(i,i,i,i)

1. *Как можно удалить ключ из В+дерева?*

Удаление ключа из В+дерева осуществляется предикатом

key\_delete(ИмяБД, ПерВ+дер, Ключ, Ссылка)(db\_selector, bt\_selector, string, ref):(i,i,i,i)

1. *Какие стандартные предикаты используются для установки указателя ключа В+дерева на первый, последний, текущий ключ?*

Стандартные предикаты для установки указателя ключа В+дерева на первый, последний, текущий ключ соответственно

key\_current(ИмяБД,ПерВ+дер,Ключ,Ссылка)(db\_selector,bt\_selector,string,ref):(i,i,o,o)

key\_first(ИмяБД, ПерВ+дер,Ссылка) (db\_selector, bt\_selector, ref):(i,i,o)

key\_last(ИмяБД, ПерВ+дер, Ссылка) (db\_selector, bt\_selector, ref):(i,i,o)

1. *Какие стандартные предикаты используются для перемещения указателя на ключ В+дерева вперед и назад?*

Стандартные предикаты для перемещения указателя на ключ В+дерева вперед и назад

key\_next(ИмяБД, ПерВ+дер, Ссылка) (db\_selector, bt\_selector, ref):(i,i,o)

key\_prev(ИмяБД, ПерВ+дер, Ссылка) (db\_selector, bt\_selector, ref):(i,i,o)

1. *Какой тип данных должны иметь ключи В+дерева?*

Ключи В+дерева должны иметь иметь тип данных string или быть приведены к нему.

1. *Как можно определить имена В+деревьев внешней базы данных?*

Определение имен В+деревьев внешней базы данных осуществляется с помощью предиката

db\_btrees(ИмяБД, Имя БДер)(db\_selector, string):(i,o)

1. *Как можно получить статистическую информацию о В+дереве ?*

bt\_statistics(СелекторБД,СелекторДерева,ЧислоКлючей,ЧислоСтраниц,Глубина,ДлинаКлюча,Порядок,РазмерСтраницы)

(db\_selector,bt\_selector,real,real,integer,integer,integer,integer) - (i,i,o,o,o,o,o,o)

Предикат позволяет получить статистику по дереву с селектором СелекторДерева, которое находится в БД с логическим именем СелекторБД.

1. *В каком порядке разместятся в В+дереве следующие значения ключа: 1, 2, 5, 12,43?*

В порядке возрастания (так и разместятся).