МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники (ФИТР)

Кафедра программного обеспечения вычислительной техники

и автоматизированных систем

**Отчет по лабораторной работе № 12**

по дисциплине: ”***Функциональное и логическое программирование***”

на тему: “ **ВНЕШНИЕ  БАЗЫ  ДАННЫХ.**””

Выполнил**:** студент гр. 10701218

Ясень Р.С.

Принял**:** Ковалькова И.А.

Минск – 2019**Цель работы**: приобретение практических навыков составления и отладки программ, работающих с внешней БД.

**Общее задание:**

Создать внешнюю базу данных на диске, имеющую два связанных списка (две цепочки chain) с именами, например, gr215 и gr315. Структура термов внешней БД:

                         student( Фамилия, Имя, Дата\_рождения, Пол).

Извлечь из первой цепочки фамилии и имена студентов указанного пола и из обеих цепочек – фамилии и имена студентов, родившихся в заданном месяце.

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 цепочек внешней БД.

**Дополнительное задание(Вариант 2):**

Выбрать из всей БД фамилии (вывод на экран по одной в строке), в которых есть заданная буква (независимо от регистра).

**Код** **программы:**

*domains*

*db\_selector = database1.*

*student = s(string, string, string, integer, integer, integer, string).*

*predicates*

*nondeterm res.*

*nondeterm menu.*

*nondeterm process(integer).*

*nondeterm control\_open\_file(string).*

*nondeterm view\_database.*

*nondeterm view\_database2(ref).*

*nondeterm view\_database3.*

*nondeterm add(integer).*

*nondeterm delete(integer).*

*nondeterm show(integer).*

*nondeterm show2(string).*

*nondeterm show3(integer).*

*nondeterm letter.*

*nondeterm check(string,string,char).*

*nondeterm rav(string,char,char).*

*clauses*

*res:- menu.*

*%меню программы*

*menu:- nl, write ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"), nl,*

*write ("0. Выход"), nl,*

*write ("1. Cоздать базу данных"), nl,*

*write ("2. Просмотреть базу данных"), nl,*

*write ("3. Добавление элементов в базу данных"), nl,*

*write ("4. Удаление базы данных"), nl,*

*write ("5. Удаление элемента из базы данных"), nl,*

*write ("6. Основное задание"), nl,*

*write ("7. Индивидуальное задание - поиск фамилиий, которые содержат заданную букву"), nl,*

*write ("8. Просмотр цепочки"), nl,nl,*

*write ("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"), nl,*

*write ("Введите пункт меню"), nl, readint (X), process(X), !.*

*process(0).*

*process(1):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), db\_create(database1, FileAddress, in\_file),*

*db\_close(database1), menu.*

*process(2):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), control\_open\_file(FileAddress),*

*db\_open(database1, FileAddress, in\_file), view\_database, db\_close(database1), menu.*

*process(3):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), control\_open\_file(FileAddress),*

*db\_open(database1, FileAddress, in\_file), add(1), db\_close(database1), menu.*

*process(4):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), control\_open\_file(FileAddress),*

*db\_delete(FileAddress, in\_file), nl, write("База данных удалена"), nl, menu.*

*process(5):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), control\_open\_file(FileAddress), db\_open(database1, FileAddress, in\_file),*

*db\_statistics(database1, CountTerms, \_, \_, \_), delete(CountTerms), db\_close(database1), menu.*

*process(6):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), control\_open\_file(FileAddress),*

*db\_open(database1, FileAddress, in\_file), nl, write("Работа с одной цепочкой(введи 1) / Работа с двумя цепочками(введи 2)"), nl, readint(Chain), nl, show(Chain).*

*process(7):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), control\_open\_file(FileAddress),*

*db\_open(database1, FileAddress, in\_file), nl, letter, db\_close(database1), menu.*

*process(8):- nl, write("Введите имя базы данных"), nl, readln(FileAddress), control\_open\_file(FileAddress),*

*db\_open(database1, FileAddress, in\_file), view\_database3, db\_close(database1), menu.*

*% работа с цепочками данных*

*show(0):- db\_close(database1), menu.*

*show(1):- write("Выберите пол м/ж"), nl, readln(S), nl, show2(S),*

*write("Работа с одной цепочкой(введи 1) / работа с двумя цепочками(введи 2) или 0 для выхода"), nl, readint(Chain), show(Chain).*

*show(2):- write("Выберите месяц"), nl, readint(Month), nl, show3(Month),*

*write("Работа с одной цепочкой(введи 1) / работа с двумя цепочками(введи 2) или 0 для выхода"), nl, readint(Chain), show(Chain).*

*show(\_):- write("Ошибка"), nl, nl, write("Работа с одной цепочкой(введи 1) / работа с двумя цепочками(введи 2) или 0 для выхода"), nl,*

*readint(Chain), nl, show(Chain).*

*show2(S):- chain\_terms(database1, "gr215", student, s(F, I, O, Day, Month, Year, S), \_),*

*write("Фамилия: ", F), nl, write("Имя: ", I), nl, write("Отчество: ", O), nl,*

*write("Дата рождения: ", Day,".", Month, ".", Year), nl, fail.*

*show2(\_).*

*show3(Month):- db\_chains(database1, Chain), chain\_terms(database1, Chain, student, s(F, I, O, Day, Month, Year, \_), \_),*

*write("Фамилия: ", F), nl, write("Имя: ", I), nl, write("Отчество: ", O), nl,*

*write("Дата рождения: ", Day,".", Month, ".", Year), nl, nl, fail.*

*show3(\_).*

*% проверка существования файла*

*control\_open\_file(FileAddress):- existfile(FileAddress).*

*control\_open\_file(FileAddress):- not(existfile(FileAddress)), nl, write("Ошибка. база данных не найдена"), nl, menu.*

*% просмотр базы данных*

*view\_database:- db\_statistics(database1, CountTerms, \_, \_, \_), nl, write("Количество элементов базы данных - ", CountTerms),*

*nl, nl, db\_chains(database1, Chain), write(Chain), nl, chain\_first(database1, Chain, Ref), view\_database2(Ref).*

*view\_database.*

*view\_database2(Ref):- ref\_term(database1, student, Ref, s(F, I, O, Day, Month, Year, \_)), nl,*

*write("Фамилия: ", F), nl, write("Имя: ", I), nl, write("Отчество: ", O), nl,*

*write("Дата рождения: ", Day,".", Month, ".", Year), nl, nl, fail.*

*view\_database2(Ref):- chain\_next(database1, Ref, Next), view\_database2(Next).*

*view\_database3:- write("Введите название цепочки"),readln(Chain), nl, chain\_first(database1, Chain, Ref), view\_database2(Ref).*

*view\_database3.*

*add(0).*

*add(1):- nl, write("Фамилия: "), readln(F), write("Имя: "), readln(I), write("Отчество: "), readln(O),*

*write("Дата рождения"), nl, write("День: "), readint(Day), write("Месяц: "), readint(Month), write("Год: "), readint(Year), write("Пол: м/ж "), readln(S), nl, nl,*

*write("Цепочка "), readln(Chain), nl, chain\_insertz(database1, Chain, student, s(F, I, O, Day, Month, Year, S), \_),*

*write("Ввести данные 0 - нет 1- да"), nl, readint(Ask), add(Ask).*

*delete(0):- write("База данных пустая"), nl, menu.*

*delete(\_):- db\_chains(database1, Chain), nl, write("Фамилия: "), readln(F), nl,*

*chain\_terms(database1, Chain, student, s(F, \_, \_, \_, \_, \_, \_), Ref), term\_delete(database1, Chain, Ref), fail.*

*delete(\_).*

*letter:- write("Введите букву - "),readchar(Letter1),write(Letter1),nl,upper\_lower(Letter1,Letter),db\_chains(database1, Chain),*

*chain\_terms(database1,Chain,student,s(F,\_,\_,\_,\_,\_,\_),\_),NF = F,check(F,NF,Letter).*

*letter.*

*rav(F,Ch,Let):- Ch=Let, write(F), nl, fail, menu.*

*rav(\_,C,L):-C<>L.*

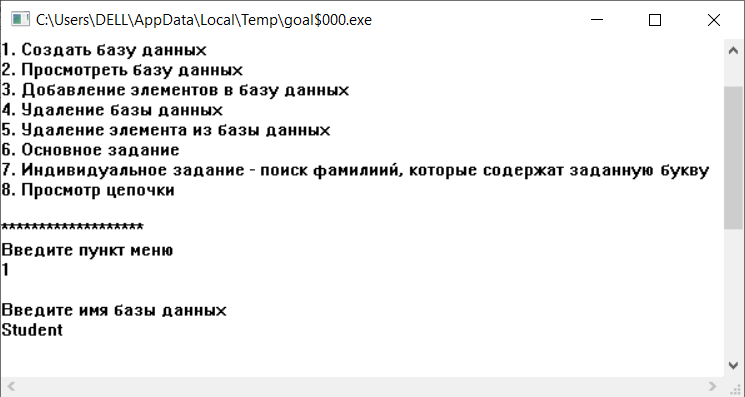
*check(\_,NF,\_):-str\_len(NF,N),N<0, write("Буква не встречается ни в одной из фамилий"),!, nl, fail, menu.*

*check(F,NF,Letter):-str\_len(NF,N),N>0,frontchar(NF,Char1,F1), upper\_lower(Char1,Char),rav(F,Char,Letter), Char<>Letter, check(F,F1,Letter).*

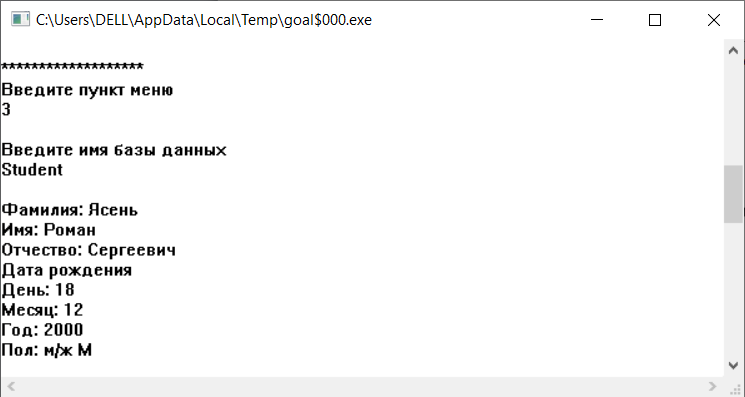
*goal*

*res.*

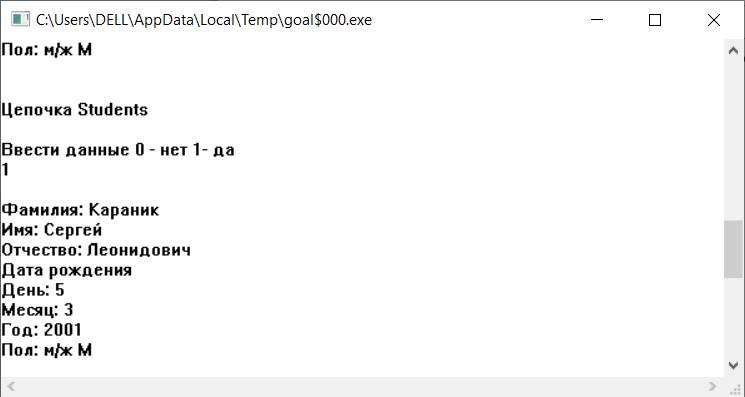
**Результат выполнения программ:**



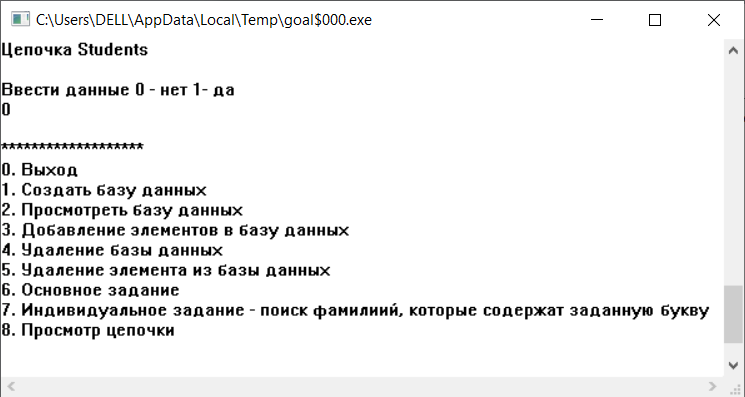
**№1.**



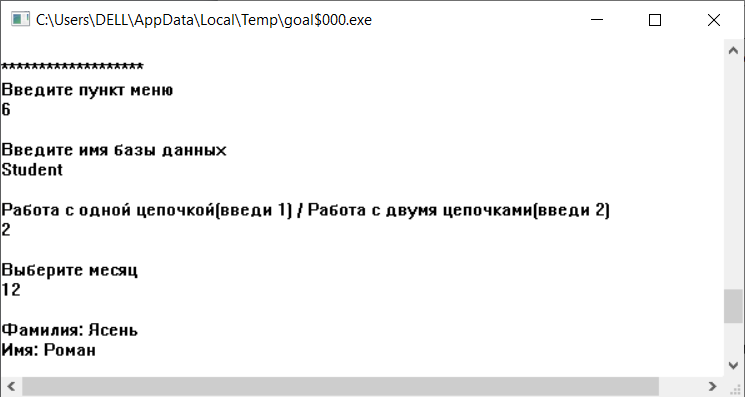
**№2.**



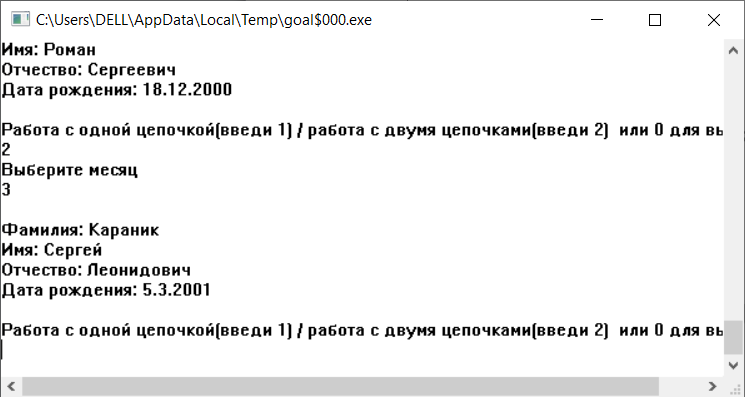
**№3.**



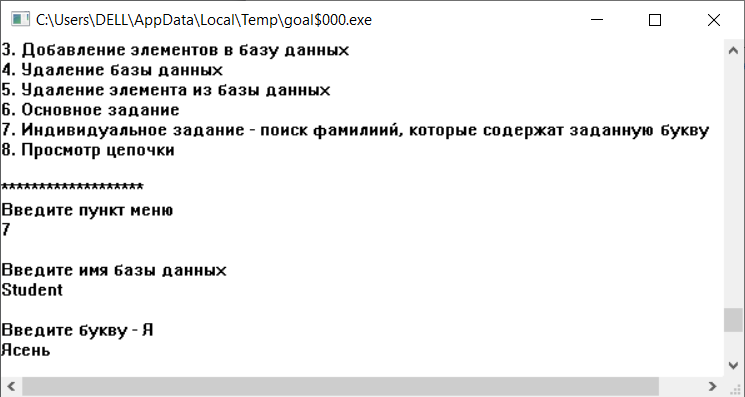
**№4.**



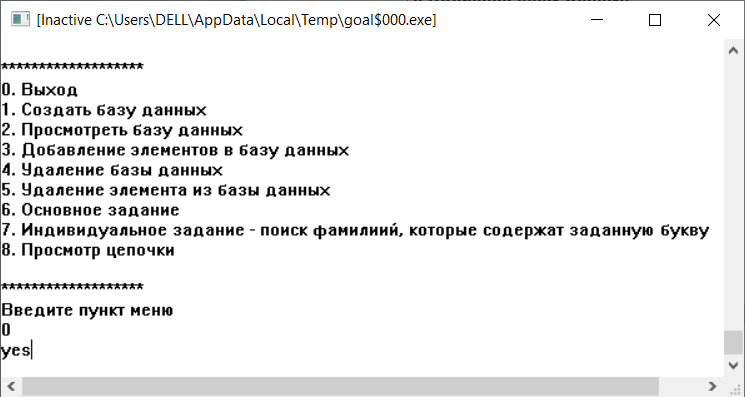
**№5.**



**№6.**



**№7.**



**№8.**

**Контрольные вопросы:**

1. Отличия внешней базы данных от динамической базы данных.

*Внешняя база данных удобна, если требуется хранить в ней много информации, в этом случае её можно удобно записать в файл. Поэтому внешняя БД обычно создается в случае, если объем данных больше свободной части ОЗУ или предполагается значительное расширение БД.*

1. Какие объекты данных могут быть использованы в качестве термов в динамической и внешней базах данных?

*Во внешней БД термы могут храниться не только в виде структуры, похожей на факт, но и в виде списков, сложных структур, чисел и т.д., то есть простые и составные объкты.*

1. Где и как производится объявление внешней базы данных в программе ?

*В разделе «domains» под доменом db\_selector объявляются имена баз данных, нужных для программы, разделённые знаком «;».*

1. Где может располагаться внешняя база данных?

*В файле, в оперативной памяти и в EMS.*

1. Соглашения об именовании стандартных предикатов управления внешними базами данных.

*Предикаты управления базами данных в целом начинаются с комбинации «db\_», предикаты управления цепочками начинаются с комбинации «chain\_», предикаты управления термами начинаются с комбинации «term\_».*

1. Что такое цепочка внешней базы данных?

*Структура базы данных, хранящая информацию. В ней можно добавлять и удалять элементы. Каждая цепочка имеет свое имя (типа symbol) и может быть обработана как отдельная группа данных.*

1. Где впервые при работе с программой, использующей внешнюю базу данных, встречается имя цепочки?

*При создании первого элемента цепочки.*

1. Что произойдет, если при вводе данных неправильно набрано имя цепочки?

*Будет создана новая цепочка с набранным именем.*

1. Что содержится во внешней базе данных, созданной предикатом db\_create?

*Во внешней базе данных никакой информации не будет, она будет пустая.*

1. Как можно определить имена цепочек внешней базы данных?

*С помощью предиката db\_chains(ИмяБД, Цепь)(db\_selector, symbol):(i,o)*

1. Какие стандартные предикаты используются для занесения термов во внешнюю базу данных?

* *chain\_inserta(ИмяБД, Цепь, Дом, Терм, Указ)(db\_selector, symbol, <domain>, <term>, ref):(i,i,i,i,o),*
* *chain\_insertz(ИмяБД, Цепь, Дом, Терм, Указ)(db\_selector, symbol, <domain>, <term>, ref):(i,i,i,i,o),*
* *chain\_insertafter(ИмяБД, Цепь, Дом, Указ, Терм, НовУказ) (db\_selector, symbol, <domain>, ref, <term>, ref):(i,i,i,i,i,o)*

1. Какие стандартные предикаты используются для движения по термам внешней базы данных?

* *chain\_first(ИмяБД, Цепь, Указ\_на\_первый\_терм)(db\_selector, symbol, ref):(i,i,o)*
* *chain\_last(ИмяБД, Цепь, Ука\_на\_последний\_терм)(db\_selector, symbol, ref):(i,i,o)*
* *chain\_next(ИмяБД,Цепь,Указ\_на\_следующий\_терм)(db\_selector, symbol, ref):(i,i,o)*
* *chain\_prev (ИмяБД,Цепь,Указ\_на\_предыдущий\_терм)(db\_selector, symbol,ref):(i,i,o)*

1. Какие стандартные предикаты используются для извлечения термов из внешней базы данных, чем они отличаются?

* *ref\_term(ИмяБД, Дом, Указ, Терм) (db\_selector,<domain>,ref,<term>)::(i,i,i,o) – извлекает терм по указателю вне зависимости от его содержимого*
* *chain\_terms( ИмяБД, Цепь, Дом, Терм, Указ)(db\_selector, symbol, <domain>, <term>, ref):(i,i,i,o,o) (i,i,i,i,o) – могут извлекаться термы только с нужными свойствами*.

1. Что такое указатель терма, его тип.

*Структура данных, с помощью которой можно перемещаться по базе данных и извлекать из неё нужные значения. Задаётся типом ref.*

1. Объяснить, как с помощью правила rd(Ref), используемого в программе, приведенной в разделе 2, извлекаются термы из одной цепочки (из всех цепочек).

*Получаем указатель на первый элемент заданной цепочки. Затем выполняется предикат rd(Ref). По полученному указателю получаем содержимое элемента и выводим его на экран. Затем переходим к следующему элементу цепочки, получаем указатель на него и с этим указателем снова начинаем выполнение данного предиката. Для извлечения термов из всех цепочек можно работать с каждой цепочкой по очереди.*

1. Как можно открыть внешнюю базу данных, помеченную как ошибочная (invalid)?

*С помощью команды db\_openinvalid(ИмяБД, ИмяФайла, Место)(db\_selector, string, symbol):(i,i,i).*

1. Будет ли отличаться скопированная внешняя база данных с помощью предиката db\_copy от оригинала?

*Содержимое отличаться не будет, а вот местонахождение скопированной внешней базы данных может отличаться.*

**Вывод:** приобрели практические навыки в составлении и отладки программ, работающих с внешней БД.