Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

Кафедра прикладной математики и кибернетики

**РАСЧЕТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ**

по дисциплине «Функциональное и логическое программирование»

Вариант 0

Выполнил:

Клапатун А.В.

Проверил:

Галкина М.Ю.

Новосибирск 2020г

Условие задачи.

Написать на языке SWI-PROLOG программу для работы с базой данных по заданию. Начальная база данных должна храниться в файле. В программе должно присутствовать меню из 5 пунктов, реализующих следующие возможности:

1. Просмотр содержимого динамической базы данных (при запуске программы динамическая база данных должна заполняться из файла базы данных только один раз и только в этом пункте);
2. Добавления записи в динамическую базу данных (за один вход в этот пункт должна быть возможность добавления нескольких записей);
3. Удаления записи из динамической базы данных (за один вход в этот пункт должна быть возможность удаления нескольких записей);
4. Выполнения запроса к динамической базе данных по заданию;
5. Выход из программы с сохранением содержимого динамической базы данных в исходный файл базы данных (сохранение должно быть только один раз и только в этом пункте).

**Задача по варианту:**

Создайте базу данных об игрушках: название, стоимость. Получите названия всех наиболее дорогих игрушек (цены которых отличаются от самой дорогой не более, чем на 100 рублей).

Исходный текст программы:

|  |  |
| --- | --- |
| % Автор: Клапатун Антон  % Дата: 12.12.2019  :-dynamic toy/2.  %...............................................................................  goal:- repeat,  writeln("Выберите один из следующих пунктов меню:"),  writeln("1 - Просмотреть содержимое динамической памяти;"),  writeln("2 - Добавление записей;"),  writeln("3 - Удаление записей;"),  writeln("4 - Получение названия всех наиболее дорогих игрушек;"),  writeln("5 - Выход (с сохранением)."),  read(Choice),  Choice < 6,  proccess(Choice),  Choice=5,!.  %...............................................................................  proccess(1):- write("Введите имя базы данных: "),  %read(NameBD),  consult("BD\_Toys.txt"),  findall((Name,Cost), toy(Name,Cost), ToysList),  write\_bd(ToysList).  proccess(2):- nl,writeln("Добавление элемента:"),  repeat,  write("Название игрушки"),  read(Name),  write("Стоимость"),  read(Cost),  assertz(toy(Name,Cost)),  nl,writeln("Хотите добавить еще запись? (y - да; n - нет): "),  read(Answ),nl,  check\_answer(Answ),!.  proccess(3):- nl,writeln("Удаление записей."),nl,  repeat,  writeln("Введите название игрушки: "),  read(Name),  remove\_record(Name),  findall((Name,Cost), toy(Name,Cost), ToysList),  write\_bd(ToysList),  nl,writeln("Хотите удалить еще что-то? (y - да; n - нет): "),  read(Answ),nl,  check\_answer(Answ),!.  proccess(4):- nl,writeln("Наиболее дорогие игрушки:"),nl,  most\_expensive\_toy.  proccess(5):- findall((Name,Cost), toy(Name,Cost), ToysList),  open("BD\_Toys.txt", write, F),  set\_output(F),  write\_in\_file(ToysList),  close(F).  %...............................................................................  %1  write\_bd([]):- !.  write\_bd([H|T]):- writeln(H),  write\_bd(T),!. | %...............................................................................  %2  %Add  check\_answer(y):- fail.  check\_answer(n):- write("Готово"),nl.  %...............................................................................  %3  %Delete  remove\_record(Name):- findall((\_,C),toy(Name,C),ListT),  write\_elem(ListT).  write\_elem([]):- nl,writeln("Sorry, but I can't to find it"),nl.  write\_elem([(N,C)|[]]):- retract(toy(N,C)).  write\_elem([(N,C)|T]):- retract(toy(N,C)),  write\_elem(T),!.  %...............................................................................  %4  %Наиболее дорогие игрушки  most\_expensive\_toy:- findall(C, toy(\_,C), ToysListN),  msort(ToysListN, ToysListN2),  reverse(ToysListN2,ToysListN3),  nth0(0,ToysListN3,Max),  condition(ToysListN3, Max, \_, \_, ToysList),  write\_bd(ToysList),nl.  condition([], ToysList, \_, ToysList):- !.  condition([H|T], Max, ToysListBefore, ToysListAfter, ToysList):- (Max-100)=< H,  findall((Name,H), toy(Name,H), TmpList),  append(ToysListBefore, TmpList, ToysListAfter),  condition(T, Max, ToysListAfter, \_, ToysList),!.  condition([\_|\_], \_,ToysList,\_,ToysList):- !.  %...............................................................................  %5  write\_in\_file([]):- !.  write\_in\_file([(Name,Cost)|T]):- write('toy("'),write(Name),  write('",'),write(Cost),  write(').'),nl,  write\_in\_file(T),!. |

Результаты работы программы:

|  |  |
| --- | --- |
|  |  |