

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Севастопольский государственный университет»

Кафедра ИС

РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА
по дисциплине «Методы и системы искусственного интеллекта»
вариант № 3

Выполнил: ст. гр. ИС/б-20-1-о
Галенин А.К.

Проверил: Сметанина Т.И.
Забаштанский А.К.

Севастополь
2024

Вариант № 3

Разработать на языке Пролог реляционную базу данных (не меньше 5 фактов), соответствующую приведенной таблице, составить 5 правил, удовлетворяющих условиям:

1) проверить, верно ли, что в базе данных есть хотя бы один студент, размер стипендии у которого больше 2 000, но меньше 4 000;

2) проверить, верно ли, что в базе данных есть хотя бы один студент, размер стипендии у которого больше 4 000 или меньше 2 000;

3) проверить, верно ли, что в базе данных есть хотя бы один студент, у которого порядковый номер и номер курса совпадают (если есть, то вывести его фамилию);

4) вывести фамилии всех студентов, обучающихся на курсе, заданном пользователем;

5) проверить, есть ли в базе данных два студента, обучающихся в одной группе и имеющих одинаковый размер стипендии.

К каждому правилу реализовать характерные запросы (не менее двух).

Таблица 1 – Сведения о стипендиях студентов

№	ФИО студента	Номер группы	Курс	Средний балл	Размер стипендии
---	--------------	--------------	------	--------------	------------------

ХОД РАБОТЫ

1. В соответствии с таблицей сведений о стипендиях студентов, были составлены необходимые факты и правила, удовлетворяющие условию задания. Полный код программы на языке SWI Prolog представлен в приложении А.

2. После чего были протестированы правила получившейся пролог-программы:

Сперва было протестировано правило `more_and_less`, при помощи которого можно проверить, есть ли в базе данных хотя бы один студент, у которого размер стипендии больше 2000, но меньше 4000. Результаты тестирования представлены на рисунках 1 и 2.

```
Вывести фамилии студентов, у которых стипендия от 2000 до 4000? {да/нет}
|: да.
Ильина А.Н. ПОрЛ/Б-21-4-о -> 3500
Введите любой символ|: █
```

Рисунок 1 – Проверка корректности работы правила `more_and_less`

```
Вывести фамилии студентов, у которых стипендия от 2000 до 4000? {да/нет}
|: нет.
В базе данных есть хотя бы один студент, размер стипендии у которого больше 2000, но меньше 4000
Введите любой символ|: █
```

Рисунок 2 – Проверка корректности работы правила `more_and_less`

Затем было протестировано правило `more_or_less`, которое позволяет определить, есть ли в базе данных хотя бы один студент, размер стипендии у которого больше 4000 или меньше 2000. Результаты тестирования представлены на рисунках 3 и 4.

```
Вывести фамилии студентов, у которых стипендия или меньше 2000 или больше 4000? {да/нет}
|: да.
Галенин А.К. ИС/Б-20-1-о -> 1048
Введите любой символ|: .
Ильина А.Н. ПОрЛ/Б-21-4-о -> 1500
Введите любой символ|: .
Петров П.П. Абвгдж-1 -> 4001
Введите любой символ|: █
```

Рисунок 3 – Проверка корректности работы правила `more_or_less`

```
Вывести фамилии студентов, у которых стипендия или меньше 2000 или больше 4000? {да/нет}
|: нет.
В базе данных есть хотя бы один студент, размер стипендии у которого большее 4000 или меньше 2000
Введите любой символ|: █
```

Рисунок 4 – Проверка корректности работы правила `more_or_less`

После чего была проверена корректность работы правила `student_with_same_course_and_id`, которое позволяет проверить, есть ли в базе данных студент у которого порядковый номер совпадает с номером курса. Результаты тестирования представлены на рисунках 5 и 6

```
Вывести фамилии студентов, у которых совпадает порядковый номер и номер курса? {да/нет}  
|: да.  
Петров П.П.  
Введите любой символ|: .  
Иванов И.И.  
Введите любой символ|: █
```

Рисунок 5 – Проверка корректности работы правила
`student_with_same_course_and_id`

```
Вывести фамилии студентов, у которых совпадает порядковый номер и номер курса? {да/нет}  
|: нет.  
В базе данных есть хотя бы один студент, у которого порядковый номер и номер курса совпадают  
Введите любой символ|: █
```

Рисунок 6 – Проверка корректности работы правила
`student_with_same_course_and_id`

Затем было протестировано правило `find_students_by_course`, позволяющее вывести всех студентов одного курса, введенного пользователем системы. Результаты тестирования представлены на рисунках 7 и 8.

```
Введите номер курса, список студентов которого необходимо вывести  
|: 4.  
Галенин А.К.  
Введите любой символ|: .  
Озеров В.В.  
Введите любой символ|: .  
Петров П.П.  
Введите любой символ|: .  
Васечкин. В.В.  
Введите любой символ|: .  
Больше нет студентов данного курса  
Введите любой символ|: █
```

Рисунок 7 – Проверка корректности работы правила `find_students_by_course`

```
Введите номер курса, список студентов которого необходимо вывести  
|: 1.  
Больше нет студентов данного курса  
Введите любой символ|: █
```

Рисунок 8 – Проверка корректности работы правила `find_students_by_course`

После чего было протестировано правило `matching_students_of_same_group_with_same_studentship`, которое необходимо, чтобы проверить, есть ли в базе данных 2 студента, которые обучаются в одной группе и имеют одинаковый размер стипендии. Результаты тестирования представлены на рисунках 9 и 10.

```
Вывести фамилии студентов из одной группы и имеющих одинаковый размер стипендии? {да/нет}  
|: да.  
Галенин А.К. Озеров В.В. -> 7777  
Введите любой символ|: .  
Озеров В.В. Галенин А.К. -> 7777  
Введите любой символ|: █
```

Рисунок 9 – Проверка корректности работы правила `matching_students_of_same_group_with_same_studentship`

```
Вывести фамилии студентов из одной группы и имеющих одинаковый размер стипендии? {да/нет}  
|: нет.  
В базе данных есть два студента, обучающихся в одной группе и имеющих одинаковый размер стипендии  
Введите любой символ|: █
```

Рисунок 10 – Проверка корректности работы правила `matching_students_of_same_group_with_same_studentship`

ВЫВОДЫ

В ходе выполнения расчётно-графической работы были закреплены навыки организации динамических баз данных средствами языка Пролог. Была разработана средствами языка Пролог база данных, включающая в себя

сведения о стипендиях студентов, также были разработаны факты и пять правил, которые позволили выполнить задание по варианту.

Все правила потом были протестированы, с использованием различных фактов, и в результате тестирования ошибок обнаружено не было.

ПРИЛОЖЕНИЕ А

Код программы

Файл students.pl

```
:- dynamic student/6.

student(1, "Галенин А.К.", "ИС/Б-20-1-о", 4, 3, 7777).
student(2, "Ильина А.Н.", "ПОРЛ/Б-21-4-о", 3, 5, 1500).
student(3, "Озеров В.В.", "ИС/Б-20-1-о", 4, 3, 7777).
student(4, "Петров П.П.", "Абвгдеж-1", 4, 4, 4001).
student(5, "Иванов И.И.", "Абвгдеж-2", 5, 5, 3999).
student(6, "Васечкин. В.В.", "ИС/Б-20-1-о", 4, 2, 2489).

menu:-
    tty_clear,

write('#####'
#####'),nl,
    write('1. Студенты со стипендией > 2000 и < 4000
'),nl,
    write('2. Студенты со стипендией > 4000 или < 2000
'),nl,
    write('3. Студенты, у которых совпадает номер курса и порядковый номер
'),nl,
    write('4. Студенты, обучающиеся на заданном курсе
'),nl,
    write('5. Студенты, обучающиеся в одной группе и имеющие одинаковый
размер стипендии'),nl,
    write('6. Выход
'),nl,
    write('{Для выбора пункта меню, необходимо написать соответствующую
цифру пункта, С ТОЧКОЙ в конце}'),nl,

write('#####'
#####'),nl,
    read(Item), menu_item(Item).

menu_item(1):-
    more_and_less,
    menu,!.

menu_item(2):-
    more_or_less,
    menu,!.

menu_item(3):-
    student_with_same_course_and_id,
    menu,!.

menu_item(4):-
    find_students_by_course,
    menu,!.

```

```

menu_item(5):-
    matching_students_of_same_group_with_same_studentship,
    menu,!.
```

menu_item(6):-!.

```

menu_item(_Item):-
    tty_clear,
    write('Нет такого пункта меню, выберите пункт в пределах от 1 до
6'),nl,
    menu.
```

more_and_less:-

```

    tty_clear,
    write('Вывести фамилии студентов, у которых стипендия от 2000 до 4000?
{да/нет}'),nl,
    read(Answer),
    (Answer = да ->
        student(_,FIO,Group,_,_,Studentship),
        Studentship > 2000, Studentship < 4000,
        write(FIO),write(' '),write(Group),write(' ') ->
    '),write(Studentship),nl,
    write('Введите любой символ'),get(temp),
    fail;
    Answer = нет ->
        student(_,FIO,Group,_,_,Studentship),
        (Studentship > 2000, Studentship < 4000),!,
        write('В базе данных есть хотя бы один студент, размер стипендии
у которого больше 2000, но меньше 4000'),nl,
        write('Введите любой символ'),get(temp)
    ).
```

more_or_less:-

```

    tty_clear,
    write('Вывести фамилии студентов, у которых стипендия или меньше 2000
или больше 4000? {да/нет}'),nl,
    read(Answer),
    (Answer = да ->
        student(_,FIO,Group,_,_,Studentship),
        (Studentship > 4000; Studentship < 2000),
        write(FIO),write(' '),write(Group),write(' ') ->
    '),write(Studentship),nl,
    write('Введите любой символ'),get(temp),
    fail;
    Answer = нет ->
        student(_,FIO,Group,_,_,Studentship),
        (Studentship > 4000; Studentship < 2000),!,
        write('В базе данных есть хотя бы один студент, размер стипендии
у которого большее 4000 или меньше 2000'),nl,
        write('Введите любой символ'),get(temp)
    ).
```

student_with_same_course_and_id:-

```

    tty_clear,
```



```

        write('Вывести фамилии студентов, у которых совпадает порядковый номер
и номер курса? {да/нет}'),nl,
        read(Answer),
        (Answer = да ->
            student(N,FIO,_,Course,_,_),
            N = Course,
            write(FIO),nl,
            write('Введите любой символ'),get(temp),
            fail;
        Answer = нет ->
            student(N,FIO,_,Course,_,_),
            N = Course,!,
            write('В базе данных есть хотя бы один студент, у которого
порядковый номер и номер курса совпадают'),nl,
            write('Введите любой символ'),get(temp)
        ).

```

```

find_students_by_course:-
    tty_clear,
    write('Введите номер курса, список студентов которого необходимо
вывести'),nl,read(User_Course),
    student(N, FIO, Group, Course, Average, Studentship),
    member(User_Course, [Course]),
    write(FIO),nl,write('Введите любой символ'),get(temp),fail,!;
    write('Больше нет студентов данного курса'),nl,write('Введите любой
символ'),get(temp).

```

```

matching_students_of_same_group_with_same_studentship:-
    tty_clear,
    write('Вывести фамилии студентов из одной группы и имеющих одинаковый
размер стипендии? {да/нет}'),nl,
    read(Answer),
    (Answer = да ->
        student(_,FIO,Group,_,_,Studentship),
        student(_,FIO_,Group,_,_,Studentship),
        FIO \= FIO_,
        write(FIO),write(' '),write(FIO_),write(' ') ->
    '),write(Studentship),nl,
        write('Введите любой символ'),get(temp);
    Answer = нет ->
        student(_,FIO,Group,_,_,Studentship),
        student(_,FIO_,Group,_,_,Studentship),
        FIO \= FIO_,!,
        write('В базе данных есть два студента, обучающихся в одной группе
и имеющих одинаковый размер стипендии'),nl,
        write('Введите любой символ'),get(temp)
    ).

```

```

:- menu.

```