

## Лабораторная работа №4

### Вариант №11

Выполнил: Котолевский М.Н. 19 группа.

#### Использование списков в Прологе

##### Цель работы:

Напишите программу, которая принимает 2 списка: 1- ФИО студентов и 2-набранные ими баллы. Нужно отсортировать по баллам и сформировать новый список успевающих студентов, в котором будут ФИО и баллы.

##### Ход работы:

Составим программу, которая удовлетворяет требованиям цели работы.

```
build_pairs([], [], []).
```

```
build_pairs([Name|NameTail], [Score|ScoreTail], [(Name, Score)|PairTail]) :-
```

```
    build_pairs(NameTail, ScoreTail, PairTail).
```

```
filter_success([], []).
```

```
filter_success([(Name, Score)|Tail], [(Name, Score)|SuccessTail]) :-
```

```
    Score >= 60,
```

```
    filter_success(Tail, SuccessTail).
```

```
filter_success([(Name, Score)|Tail], SuccessTail) :-
```

```
    Score < 60,
```

```
    filter_success(Tail, SuccessTail).
```

```
build_success_list([], []).
```

```
build_success_list([(Name, Score)|Tail], [Success|SuccessTail]) :-
```

```
    atom_concat(Name, ' - ', NameWithDash),
```

```
    atom_concat(NameWithDash, Score, Success),
```

```
    build_success_list(Tail, SuccessTail).
```

```
success_students(Names, Scores, SuccessList) :-
```

```
    build_pairs(Names, Scores, Pairs),
```

```
    filter_success(Pairs, SuccessPairs),
```

```
    sort(2, @>=, SuccessPairs, SortedPairs),
```

```
build_success_list(SortedPairs, SuccessList).
```

`build_pairs` принимает три пустых списка и рекурсивно строит список кортежей, состоящий из пар имени и оценки студента.

`filter_success` принимает список кортежей и рекурсивно фильтрует только те записи, где оценка студента равна или больше 60 баллов. Это создает список успевающих студентов.

`build_success_list` принимает список кортежей успешных студентов и рекурсивно создает новый список, в котором каждый элемент представляет собой строку в формате "Имя - Оценка".

`success_students` принимает имена и оценки студентов и использует предыдущие предикаты для создания и сортировки списка успешных студентов.

## Результаты работы:

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?-
Warning: c:/users/user/desktop/4_11/p_4.pl:9:
Warning: Singleton variables: [Name]
% c:/Users/user/Desktop/4_11/p_4.pl compiled 0.00 sec, 8 clauses
?- success_students(["Ivanov", "Petrov", "Sidorov", "Smirnov"], [75, 50, 80, 65],
SuccessList).
SuccessList = ['Sidorov - 80', 'Ivanov - 75', 'Smirnov - 65']
```

```
SWI-Prolog (AMD64, Multi-threaded, version 9.0.4)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- success_students(["Ivanov", "Petrov", "Sidorov", "Smirnov"], [50, 50, 50, 50],
SuccessList).
Warning: c:/users/user/desktop/4_11/p_4.pl:9:
Warning: Singleton variables: [Name]
% c:/Users/user/Desktop/4_11/p_4.pl compiled 0.00 sec, 8 clauses
?- success_students(["Ivanov", "Petrov", "Sidorov", "Smirnov"], [50, 50, 50, 50],
SuccessList).
SuccessList = [].
```

## Запросы:

```
success_students(["Ivanov", "Petrov", "Sidorov", "Smirnov"], [75, 50, 80, 65], SuccessList).
```

```
success_students(["Ivanov", "Petrov", "Sidorov", "Smirnov"], [50, 50, 50, 50], SuccessList).
```

**Вывод:**

Была разработана программа, которая принимает 2 списка: 1- ФИО студентов и 2-набранные ими баллы. Отсортировав по баллам, сформирован новый список успевающих студентов, в котором находятся ФИО и баллы.