Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Новгородский государственный университет имени Ярослава Мудрого»

Кафедра информационных технологий и систем

**ИСПОЛЬЗОВАНИЕ ОТСЕЧЕНИЙ**

Лабораторная работа по дисциплине:

“Функциональное и логическое программирование”

Выполнил студент группы 8091:

\_\_\_\_\_\_\_\_\_\_ / Васильев И.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Принял преподаватель:

\_\_\_\_\_\_ / Михайлов Д.В.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**Великий Новгород**

**2021**

# Цель и задачи

Целью работы является изучение правил использования отсечения в двух случаях: для подтверждения правильности выбранного решения и для прекращения процесса порождения и проверки возможных решений.

Задача №1  
Создать меню, обеспечить вызов и выполнение всех заданий из Лабораторной работы № 6. Каждое задание должно выполняться в отдельном окне. При выполнении работы в среде SWI-Prolog (независимо от версии) рекомендуется использовать средства создания графического интерфейса (см. ниже, а также прилагаемые примеры из предыдущей лабораторной работы). Реализовать процедуру ввода списка. При использовании среды визуальной разработки SWI-Prolog изучить принципы расстановки отсечений в предложениях, которые генерируются экспертом кода и обрабатывают события от пунктов меню.

Задача №2  
Написать для игры «Крестики-нолики» процедуру «Следующий ход», которая для заданного положения на доске находила бы наилучший ход, обеспечивающий либо предотвращение проигрыша, либо выигрыш, либо наилучший прогнозируемый результат.

Задача №3

Написать программу, реализующую выигрышную стратегию для игры «Крестики-нолики»

на доске 3 × 3.

# Решение

# Для решения поставленных задач была использована реализация языка Prolog –SWI-Prolog.

# Задача №1

Создать меню, обеспечить вызов и выполнение всех заданий из Лабораторной работы № 6. Каждое задание должно выполняться в отдельном окне. При выполнении работы в среде Visual Prolog (независимо от версии) рекомендуется использовать средства создания графического интерфейса (см. ниже, а также прилагаемые примеры из предыдущей лабораторной работы). Реализовать процедуру ввода списка. При использовании среды визуальной разработки Visual Prolog изучить принципы расстановки отсечений в предложениях, которые генерируются экспертом кода и обрабатывают события от пунктов меню.

Программный код представлен в архиве в файле “lab71.pl”

Пример работы программы:

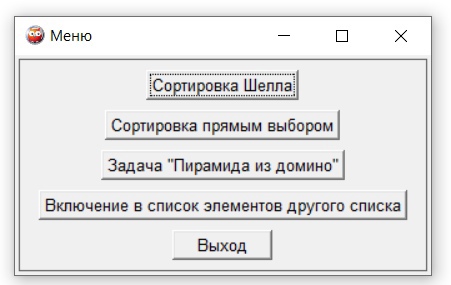


Рисунок 1 Визуальное окно главного меню

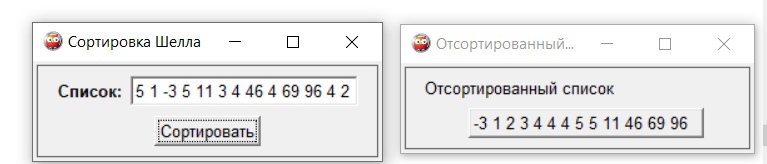


Рисунок 2 Визуальное окно задачи сортировки Шелла

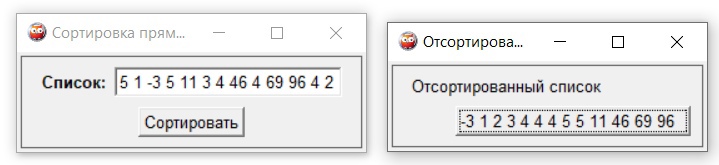


Рисунок 3 Визуальное окно задачи сортировки прямым выбором

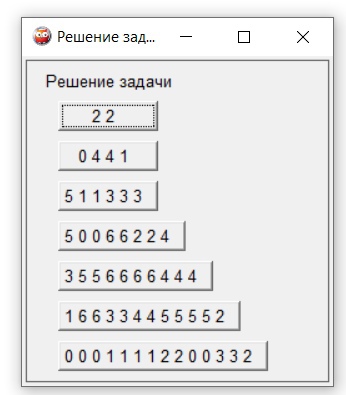


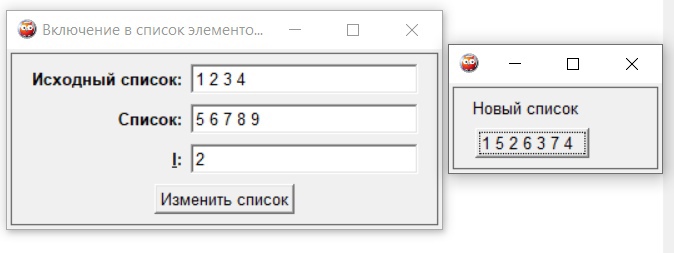
Рисунок 4 Визуальное окно задачи пирамида из домино  


Рисунок 5 Визуальное окно задачи включения в список элементов

Обоснование:  
Для решения задачи было создано главное окно «Меню» (menu()) рисунок 1, где реализованы 4 кнопки, которые ведут к окнам, которые решают поставленные задачи:  
  
1.«Сортировка Шелла» (shell\_win()) - сортировка списка методом Шелла рисунок 2;

2. «Сортировка прямым выбором» (selection\_win())- сортировка методом прямого выбора рисунок 3;

3. «Задача “Числа по периметру”» (domino\_result()) - решение головоломки «Числа по периметру» рисунок 4;

4. «Вставка подсписка в список» (insert\_win()) - вставка подсписка в список, начиная с i-го элемента рисунок 5.

В данном задании был использован механизм отсечения в конце и в начале вызовов обработчиков событий. Использование этого отсечения в начале правила гарантирует запоминание всех целей как неизменяемых, которые были установлены перед вхождением в это правило. Использование отсечения в конце подтверждает правильность выбора правила и прекращает дальнейший поиск.

# Задание 2

Написать для игры «Крестики-нолики» процедуру «Следующий ход», которая для заданного положения на доске находила бы наилучший ход, обеспечивающий либо предотвращение проигрыша, либо выигрыш, либо наилучший прогнозируемый результат.

Программный код представлен в архиве в файле “lab72.pl”

Пример работы программы:

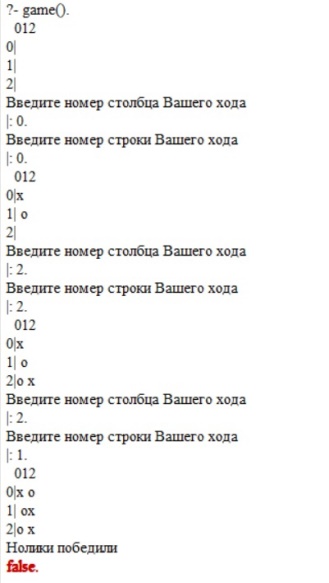


Рисунок 6 Пример работы программы - победа ноликов (компьютер)

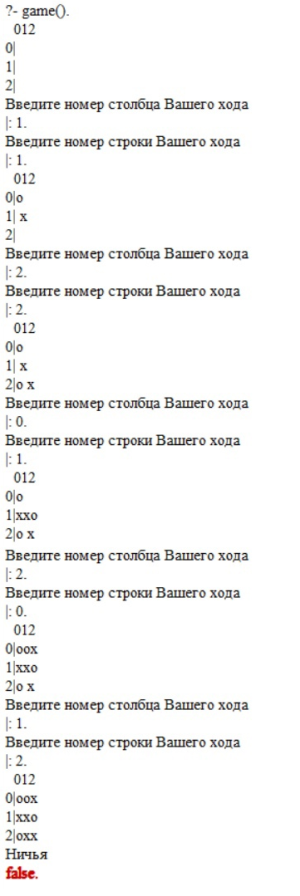


Рисунок 7 Пример работы программы - ничья

Обоснование:  
 Запуск программы осуществляется функцией game().

Так как правила проверяются сверху вниз, а сверху находятся более ценные ходы, для хода ноликов были написаны правила potential\_moves\_o(), с использованием отсечений.

Был применён следующий алгоритм:

* если игрок почему-то не занял центральную клетку – ставим туда нолик;
* если есть возможность победить, то выбираем выигрышную позицию;
* если у крестиков выигрышная ситуация, то предотвращаем её;
* если центральная ячейка к моменту вашего начального хода будет уже занята, то выбираем левую верхнюю ячейку;
* если игрок занял угловую позицию, то занимаем угловую позицию на -противоположной стороне
* иначе занимаем угловые позиции, если они заняты, любую свободную ячейку.

Отсечение в данной задачи используется для поиска первого нахождения правила для хода ноликов из приведённых, которое удовлетворяет условиям.

# Задание 3

Написать программу, реализующую выигрышную стратегию для игры «Крестики-нолики» на доске 3 × 3. Игровое поле и весь процесс игры должен отображаться на экране в графическом режиме.

Программный код представлен в архиве в файле “lab73.pl”

Пример работы программы:

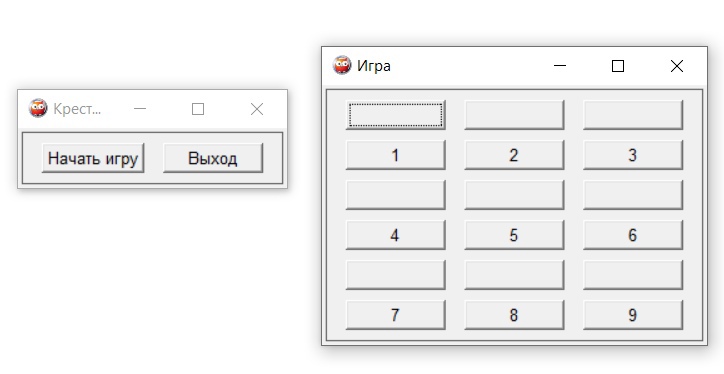


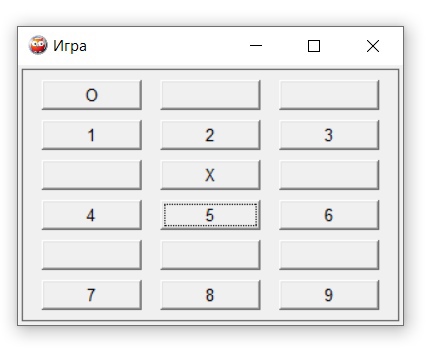
Рисунок 8 Визуальные окна меню и игрового поля  


Рисунок 9 Визуальное окно игры “Крестики-нолики”  
  
Игрок ходит крестиками, при нажатии на поле с цифрами, над ним появляется "Х"

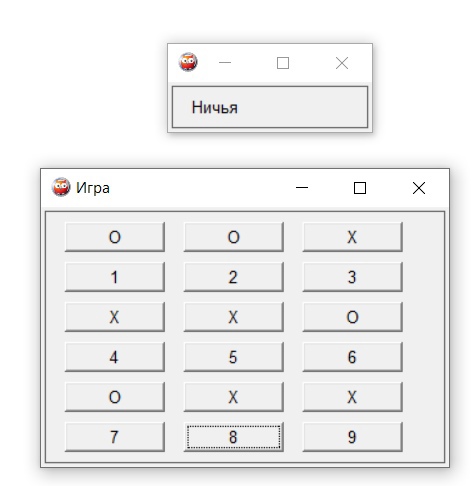


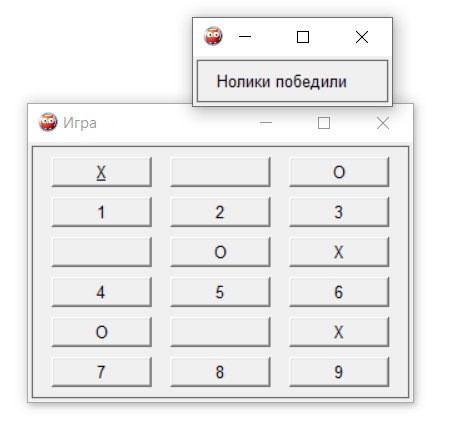
Рисунок 10 Пример игровой ничьи  


Рисунок 11 Пример победы ноликов (компьютер)

Обоснование:

Реализованная выигрышная стратегия:

Основная идея заключается в проверке 9-ти возможных вариантов и комбинаций (все линии по вертикали, горизонтали и диагонали). Если на одной из линий образуется два одинаковых символа, то бот ставит «0» для выигрыша или же предотвращения проигрыша. Если же данная ситуация не обнаружена, то бот ставит «0» в свободное поле, рядом с другим «0». Если же данный ход является первым для бота, то он ставит «0» в свободное место. Игра заканчивается либо победой пользователя, либо победой бота. Также может быть ничья, ситуация, когда всё поле уже заполнено, но ни один из игроков не победил.

Отсечение в данной задачи используется для поиска первого нахождения правила для хода ноликов из приведённых, которое удовлетворяет условиям.

# Выводы по проведенным экспериментам

В ходе выполнения задач лабораторной работы, я изучил работу механизма отсечения. Механизм отсечения позволяет экономить память и время на вычисления альтернативных вариантов.