

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский политехнический
университет Петра Великого»

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

Направление: 02.03.01 Математика и компьютерные науки

Отчет о выполнении курсовой работы
«Реализация экспертной системы для выбора спортивной
секции для ребенка»
Дисциплина «Теория графов»

Выполнил студент группы

№5130201/30001

Мелещенко С.И.

Проверил

Востров А. В.

Санкт-Петербург, 2025

Содержание

Введение	3
1 Предметная область	4
2 Математическое описание	6
2.1 Экспертная система	6
2.2 Продукционная модель	7
2.3 Бинарное дерево решений	7
3 Реализация экспертной системы	12
3.1 deffunction member	12
3.2 deffunction ask-question	12
3.3 deffunction YesOrNo	13
3.4 defrule print	13
3.5 Основные правила системы	14
4 Результаты работы программы	16
Заключение	18
Список литературы	19
Приложение А	20

Введение

Данный отчет представляет собой описание выполнения курсовой работы по теме «реализация экспертной системы для выбора спортивной секции для ребенка». Работа выполнена в рамках изучения дисциплины «Теория графов».

Цель.

Реализация экспертной системы на языке CLIPS(«C Language Integrated Production System») для поиска ответа на вопрос «На какую секцию можно отдать ребенка?».

Задачи.

1. Определение предметной области.

а. Предметная область должна быть согласована с преподавателем.

2. Построение продукционной модели.

а. Модель должна быть представлена в виде бинарного дерева (минимум 4 яруса, не считая корня, и 30 узлов).

б. Построенное бинарное дерево должно быть согласовано с преподавателем.

3. Реализация экспертной системы в среде CLIPS.

а. Защита от некорректного пользовательского ввода.

Программа была реализована с помощью CLIPS в блокноте и запущена через CLIPSIDE.

1 Предметная область

В качестве предметной области экспертной системы была выбрана тема выбора спортивной секции для ребенка. Экспертная система разработана для помощи родителям в выборе оптимальной спортивной секции. Были составлены основные вопросы по этой предметной области основывающиеся на возрасте, здоровье, физических и психологических особенностях ребенка. Ответ на каждый вопрос шаг за шагом ведет к рекомендуемой секции.

Критерии.

1. Медицинские показания.

1. Наличие хронических заболеваний/противопоказаний.

Если "да" переход к уточнению характера ограничений.

Если "нет" переход к возрастным критериям.

Проблемы с опорно-двигательным аппаратом - плавание.

Дыхательные проблемы - йога.

Полный запрет активной деятельности - ЛФК под присмотром врача.

2. Возрастные критерии.

Ребенку меньше 3 лет - игровые развивающие секции.

Ребенку 3–7 лет - активные, но не травмоопасные виды.

Ребенку больше 7 лет - разделение по интересам, виды спорта, не требующие занятий с самого детства, единоборства.

3. Физические данные.

Природная гибкость/растяжка - художественная гимнастика, акробатика, фигурное катание, спортивная гимнастика, карате, тхэквондо, синхронное плавание, конькобежный спорт.

Худощавое телосложение - художественная гимнастика.

Высокий рост - баскетбол, волейбол.

Физическая развитость - тяжелая атлетика, спортивная гимнастика, акробатика, пауэрлифтинг, гиревой спорт.

Хорошая реакция - большой теннис, хоккей.

Развитая телесная координация - фехтование, танцы.

Хорошее чувство ритма - балльные танцы, танцы.

4. Психологические особенности.

Интерес к командным играм - футбол, хоккей, баскетбол.

Любовь к индивидуальным занятиям - большой теннис, плавание, легкая атлетика.

Активный ребенок - легкая атлетика, танцы, хоккей.

Спокойный ребенок - шахматы, конный спорт, керлинг, гольф, настольный теннис, го.

Не боится высоты - скалолазание.

5. Интересы и предпочтения.

Боевые искусства - карате, тхэквондо, дзюдо.

Традиционные стили в боевых искусствах - айкидо, кэндо, кобудо, ушу.

Интерес к работе с оружием в боевых искусствах - кэндо, кобудо, ушу.

Любовь к животным - конный спорт.

Любовь к воде - плавание.

Зимние виды спорта - фигурное катание.

2 Математическое описание

2.1 Экспертная система

Экспертная система — это информационная система, назначение которой частично или полностью заменить эксперта в той или иной предметной области. Экспертные системы не отвергают и не заменяют традиционного подхода к разработке программ, ориентированного на решение формализованных задач, но дополняют их, тем самым значительно расширяя возможности.

Экспертная система работает в двух основных режимах:

- 1) в режиме приобретения знаний;
- 2) в режиме решения задачи (называемом также режимом консультаций, или режимом использования экспертной системы).

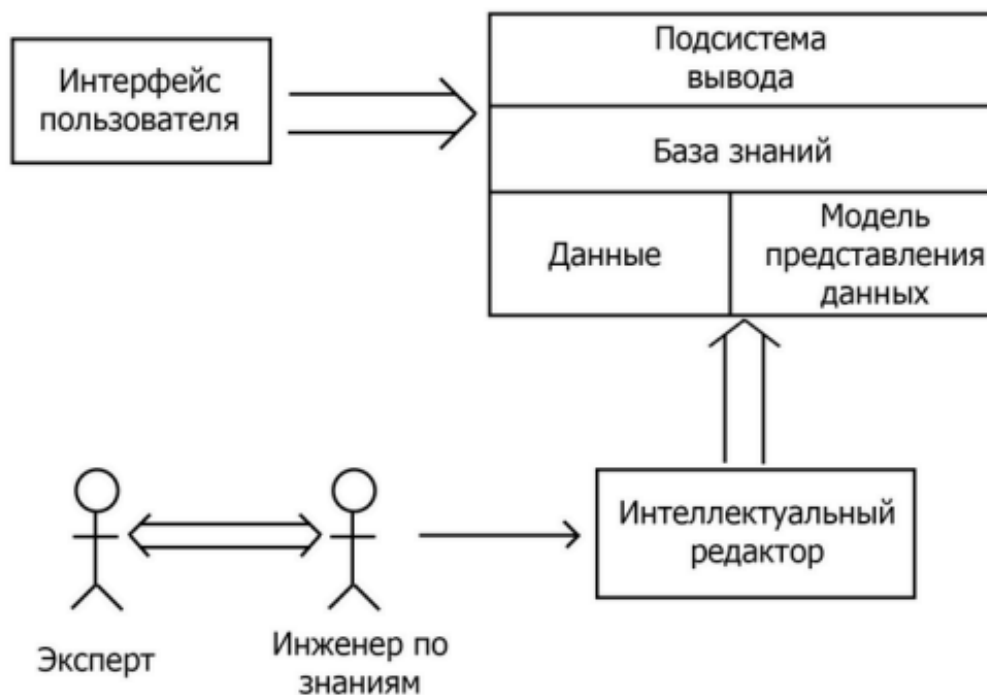


Рис. 1. Структура экспертной системы

Знания — это правила, законы, закономерности получены в результате профессиональной деятельности в пределах предметной области.

База знаний — база данных содержащая правила вывода и информацию о человеческом опыте и знаниях в некоторой предметной области.

Другими словами, это набор таких закономерностей, которые устанавливают связи между вводимой и выводимой информацией.

Данные — это совокупность фактов и идей представленных в формализованном виде.

2.2 Продукционная модель

Продукционная модель знания — модель, основанная на правилах, позволяет представить знание в виде предложений типа «Если (условие), то (действие), иначе (действие)».

Продукция состоит из двух частей: условие — антецедент, действие — консеквент.

Условия можно сочетать с помощью логических функций AND, OR.

В базе данных продукционной системы хранятся правила, истинность которых установлена к заранее при решении определенной задачи. Правило срабатывает, если при сопоставлении фактов, содержащихся в базе данных с антецедентом правила, которое подвергается проверке, имеет место совпадение. Результат работы правила заносится в базу данных.

2.3 Бинарное дерево решений

Бинарное дерево - это иерархическая структура данных, в которой каждый узел имеет не более двух дочерних узлов. Узлы обычно называются правыми и левыми потомками. При этом каждый из потомков, в свою очередь тоже является узлом, который может иметь двух потомков. Если у узла нет потомков, такой узел называют листом.

Перечень вопросов.

1. Есть ли у ребенка хронические заболевания/противопоказания к занятиям спорта по противопоказаниям от врача (специальная медицинская группа)?

2. Есть ли проблемы с опорно-двигательным аппаратом?

3. Есть ли дыхательные проблемы (пр. астма)?

4. Есть ли полный запрет активной деятельности?

5. Ребенку меньше 3-х лет?

6. Ребенку больше 7-и лет?

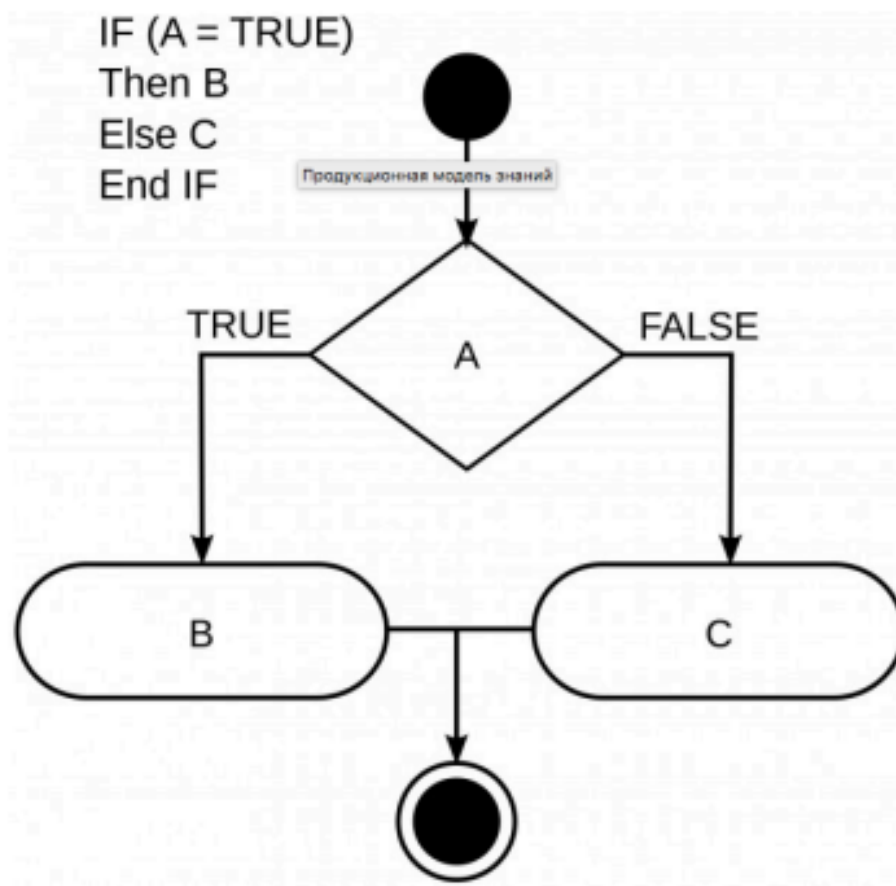


Рис. 2. Структура правил

7. Есть ли интерес к командным играм?
8. Есть ли генетически заложенный высокий рост?
9. Интересуют ли боевые искусства?
10. Интересуют ли традиционные/философские стили?
11. Есть ли у ребенка природная хорошая растяжка?
12. Хочет ли ребенок работать с оружием?
13. Комфортно ли ребенку заниматься деятельностью без физической подвижности?
14. Физически развитый ли ребенок?
15. Хорошо ли у ребенка развита телесная координация?
16. Активный ли ребенок?
17. Любит ли лошадей?
18. Любит ли плавать?

19. Любит ли бегать?
20. Хорошо ли находит язык с другими детьми?
21. Есть ли природная гибкость, растяжка?
22. Худощавое ли телосложение (узкие плечи, видны ребра при глубоком вдохе, тонкие кости, мало мышц)?
23. Физически развитый ли ребенок?
24. Любит ли ребенок плавать?
25. Интересуется ли зимними видами спорта? Или катается на коньках?

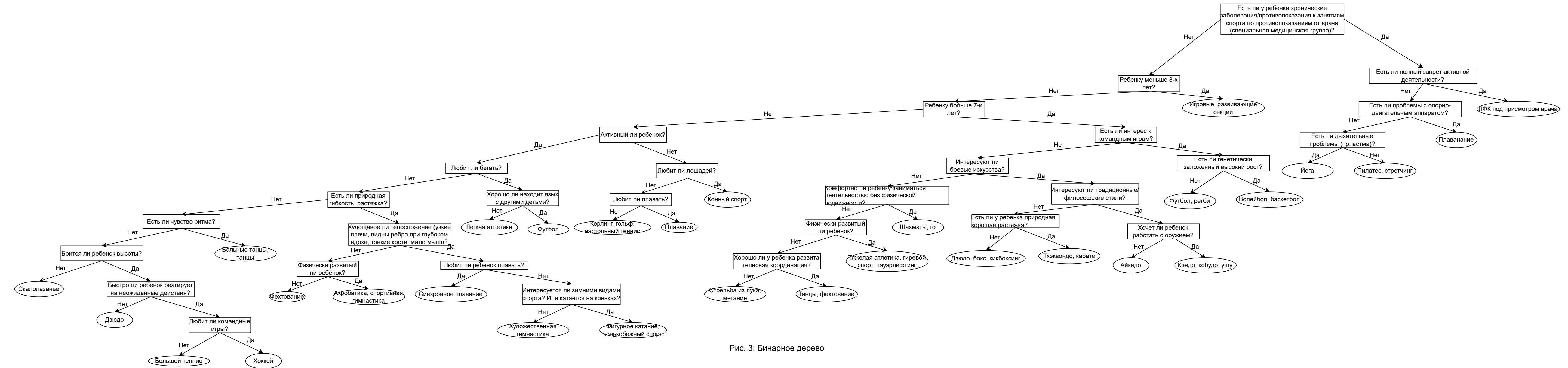
26. Есть ли чувство ритма?
27. Любит ли находить способы обойти/перелезть через преграды?
28. Быстро ли ребенок реагирует на неожиданные действия?
29. Любит ли командные игры?

Перечень возможных результатов.

1. Скалолазание.
2. Дзюдо.
3. Большой теннис.
4. Хоккей.
5. Бальные танцы, танцы.
6. Фехтование.
7. Акробатика, спортивная гимнастика.
8. Синхронное плавание.
9. Легкая атлетика.
10. Футбол.
11. Художественная гимнастика.
12. Керлинг, гольф, настольный теннис.
13. Плавание.
14. Фигурное катание, конькобежный спорт.
15. Конный спорт.
16. Стрельба из лука, метание.
17. Танцы, фехтование.

18. Тяжелая атлетика, гиревой спорт, пауэрлифтинг.
19. Шахматы, го.
20. Дзюдо, бокс, кикбоксинг.
21. Тхэквондо, карате.
22. Айкидо.
23. Кэндо, кобудо, ушу.
24. Футбол, регби.
25. Волейбол, баскетбол.
26. Игровые, развивающие секции.
27. ЛФК под присмотром врача.
28. Пилатес, стретчинг.
29. Йога.
30. Плавание.

Всего 29 узлов, 30 листьев и 10 ярусов. Ниже приведена схема бинарного дерева решений для моей экспертной системы.



3 Реализация экспертной системы

Была реализована экспертная система посредством CLIPS.

Перед запуском программы необходимо ввести (clear).

После нужно загрузить файл (load "main.clp").

И запустить код с помощью (run).

Полный код программы приведен в приложении 1.

3.1 deffunction member

Эта функция проверяет наличие элемента в списке, предварительно приводя строки к нижнему регистру.

Вход: ?item - элемент для поиска (может быть любого типа: строка, число и т.д.), \$?list - список значений, в котором производится поиск (переменное число аргументов).

Выход: Логическое значение TRUE/FALSE.

Сначала lexemer проверяет, является ли элемент строкой, если да, то lowercase приводит строку к нижнему регистру. Затем используем member\$ - встроенную функцию CLIPS для поиска элемента в списке.

Листинг 1. deffunction member

```
1 (deffunction member (?item $?list)
2   (if (lexemer ?item)
3     then (bind ?item (lowercase ?item)))
4   (member$ ?item ?list))
```

3.2 deffunction ask-question

Функция задает пользователю вопрос и проверяет ответ на соответствие допустимым значениям ("да" , "нет")

Вход: ?question - строка с вопросом для пользователя, \$?allowed-values - список допустимых ответов (переменное число аргументов).

Вывод: ответ пользователя, приведенный к нижнему регистру ("да" , "нет").

Сначала выводим вопрос пользователю и считываем ответ с клавиатуры. Если ответ - это строка, которую мы приводим к нижнему регистру. Пока ответ не входит в допустимые значения, повторяем вопрос. Затем снова читаем ответ, если это строка, то приводим к нижнему регистру и возвращаем валидный ответ.

Листинг 2. `deffunction ask-question`

```
1 (deffunction ask-question (?question $?allowed-values)
2   (printout t ?question)
3   (bind ?answer (read))
4   (if (lexemep ?answer)
5       then (bind ?answer (lowercase ?answer)))
6   (while (not (member ?answer ?allowed-values)) do
7     (printout t ?question)
8     (bind ?answer (read))
9     (if (lexemep ?answer)
10        then (bind ?answer (lowercase ?answer))))
```

3.3 `deffunction YesOrNo`

Функция задает пользователю вопрос и проверяет ответ на соответствие допустимым значениям ("да" , "нет")

Вход: `?question` - строка с вопросом.

Вывод: `TRUE`, `FALSE`.

Задаем вопрос с вариантами "да" , "нет". Если ответ да, то возвращаем `TRUE`. Иначе возвращаем `FALSE`.

Листинг 3. `deffunction YesOrNo`

```
1 (deffunction YesOrNo (?question)
2   (bind ?response (ask-question ?question да нет))
3   (if (eq ?response да)
4       then TRUE
5       else FALSE))
```

3.4 `defrule print`

Выводит рекомендации для пользователя.

Вход: `repair ?item`.

Выход: вывод в консоль.

salience 0 - низкий приоритет, выполняется последним. Правило repair активируется, когда в рабочей памяти появляется факт вида (repair <значение>). Printout выводит данные в стандартный поток вывода, format - функция форматированного вывода.

t - вывод в консоль

"%s%n%n%n":

%s - вставка строкового значения (?item).

%n - перенос строки (3 раза).

?item - значение из факта repair

Листинг 4. defrule print

```
1 (defrule print
2   (declare (salience 0))
3   (repair ?item) =>
4   (printout t crlf)
5   (format t "%s%n%n%n" ?item))
```

3.5 Основные правила системы

Рассмотрим на примере defrule ageless3.

Вход: (health no), (not (ageless3?)), (not (repair?)), информация о том проверялся ли возраст, нет ли предыдущих ограничений и что нет проблем со здоровьем.

Выход: (assert (repair "...")) или (assert (ageless3 no))), то есть вывод текста с рекомендуемой сессией или запоминание выбора.

(health no) проверяет наличие факта, что у ребенка нет проблем со здоровьем. (not (ageless3?)) проверяет отсутствие факта о проверке возраста, (not (repair?)) проверяет отсутствие уже готовых рекомендаций (repair).

(if (YesOrNo "Ребенку меньше 3-х лет? Напишите да/нет: ") вызывает функцию YesOrNo, которая выводит вопрос пользователю. Она принимает ответ "да" или "нет". Возвращает TRUE для "да" FALSE для "нет".

Если ответ "да" (assert (repair "Рекомендуемые секции: Игровые развивающие секции.")) Создает факт с рекомендацией.

Если ответ "нет" (assert (ageless3 no)) Фиксирует, что возраст прове-

рялся и ребенок старше 3 лет.

Позволяет активировать следующие правила в цепочке

Листинг 5. defrule ageless3

```
1 (defrule ageless3 ""  
2 (health no)  
3 (not (ageless3?))  
4 (not (repair?)) =>  
5 (if (YesOrNo "Ребенку меньше 3-х лет? Напишите да/нет: ")  
6 then  
7 (assert (repair "Рекомендуемые секции: Игровые развивающие секци  
и. "))  
8 else  
9 (assert (ageless3 no))))
```

4 Результаты работы программы

При запуске программы выводится первый вопрос (см. [Рис. 4]).

```
CLIPS> (reset)
CLIPS> (run)
Есть ли у ребенка хронические заболевания/противопоказания? Напишите да/нет: |
```

Рис. 4. Начальный экран

Далее пользователь ответами да или нет приближает себя к определенному листу дерева (см. [Рис. 5] [Рис. 6], [Рис. 7]),.

```
CLIPS> (reset)
CLIPS> (run)
Есть ли у ребенка хронические заболевания/противопоказания? Напишите да/нет: да
Есть ли полный запрет активной деятельности? Напишите да/нет: нет
Есть ли проблемы с опорно-двигательным аппаратом? Напишите да/нет: нет
Есть ли дыхательные проблемы (пр. астма)? Напишите да/нет: да
Есть ли дыхательные проблемы (пр. астма)? Напишите да/нет: да
Рекомендуемые секции: Йога
```

Рис. 5. Работа экспертной системы пример 1

```
CLIPS> (reset)
CLIPS> (run)
Есть ли у ребенка хронические заболевания/противопоказания? Напишите да/нет: нет
Ребенку меньше 3-х лет? Напишите да/нет: нет
Ребенку больше 7-и лет? Напишите да/нет: да
Есть ли интерес к командным играм? Напишите да/нет: нет
Интересуют ли боевые искусства? Напишите да/нет: нет
Комфортно ли ребенку заниматься деятельностью без физической подвижности? Напишите да/нет: да
Рекомендуемые секции: Шахматы, го.
```

Рис. 6. Работа экспертной системы пример 2


```

CLIPS> (reset)
CLIPS> (run)
Есть ли у ребенка хронические заболевания/противопоказания? Напишите да/нет: нет
Ребенку меньше 3-х лет? Напишите да/нет: нет
Ребенку больше 7-и лет? Напишите да/нет: нет
Активный ли ребенок? Напишите да/нет: да
Любит ли бегать? Напишите да/нет: нет
Есть ли природная гибкость, растяжка? Напишите да/нет: да
Худощавое ли телосложение (узкие плечи, видны ребра при глубоком вдохе, тонкие кости, мало мышц? Напишите да/нет: нет
Физически развитый ли ребенок? Напишите да/нет: да

Рекомендуемые секции: Акробатика, спортивная гимнастика.

```

Рис. 7. Работа экспертной системы пример 3

При некорректном вводе (не "да"или не "нет") вопрос повторяется (см. [Рис. 8]).

```

Ребенку меньше 3-х лет? Напишите да/нет: казуууаупупку
Ребенку меньше 3-х лет? Напишите да/нет: ----
Ребенку меньше 3-х лет? Напишите да/нет: 585444p4
Ребенку меньше 3-х лет? Напишите да/нет: да

```

Рис. 8. Некорректный ввод

При достижении листа дерева пользователь получает результат - рекомендуемую секцию (см. [Рис. 9]).

```

Есть ли противопоказания по активной деятельности? Напишите да/нет: да
Рекомендуемые секции: Стретчинг, пилатес.

```

Рис. 9. Вывод результата

Заключение

В ходе выполнения работы была выбрана предметная область, построена продукционная модель, разработана экспертная система для подбора спортивной секции ребенку на основе его индивидуальных характеристик и медицинских показаний. Система реализована в среде CLIPS, что позволило организовать эффективный механизм логического вывода рекомендаций.

Система охватывает 5 основных аспектов: медицинские показания (4 критерия), физическое развитие (11 критериев), психологические особенности (4 критерия), возрастные ограничения (3 категории), личные предпочтения (7 критериев).

Бинарное дерево состоит из 29 узлов, 30 листьев и 10-и ярусов. Всего на работу было потрачено около 4х часов.

Недостатком данной программы является ограниченность моей экспертной системы. Определен конечный набор возможных рекомендаций, что может не охватывать все возможные варианты выбора спортивных секций. База спортивных секций ограничена 30-ю видами спорта.

Преимуществом данной системы является то, что она обеспечивает персонализированные рекомендации, учитывая индивидуальные особенности ребенка. Учитываются индивидуальные особенности ребенка (Физические данные, здоровье, темперамент), что гарантированно помогает подобрать наиболее подходящий вид спорта из имеющихся.

Дополнительно может быть расширено дерево добавлением новых секций и критериев, реализован ввод данных о ребенке через анкету.

Список использованной литературы

- [1] Т.А. Гаврилова, А.П. Частиков. «Разработка экспертных систем. Среда CLIPS». (дата обращения: 13.04.2025)
- [2] CLIPS. A Tool for Building Expert Systems.
// URL: <https://www.clipsrules.net/> (дата обращения: 12.04.2025)
- [3] Пример экспертной системы.
// URL: <https://habr.com/en/sandbox/32843/> (дата обращения: 13.04.2025)
- [4] Бинарные деревья.
// URL: <https://habr.com/en/articles/835706/> (дата обращения: 14.04.2025)

Приложение А

Листинг 6. Полный код программы

```
1 (deffunction member (?item $?list)
2   (if (lexeme? ?item)
3     ; Проверяем, является ли элемент строкой
4     then (bind ?item (lowercase ?item)))
5     ; Если да, приводим к нижнему регистру
6     (member$ ?item ?list))
7   ; Проверяем наличие элемента в списке
8
9 (deffunction ask-question (?question $?allowed-values)
10  (printout t ?question)
11  ; Выводим вопрос на экран
12  (bind ?answer (read))
13  ; Читаем ответ пользователя
14  (if (lexeme? ?answer)
15    ; Если ответ — строка
16    then (bind ?answer (lowercase ?answer)))
17    ; Приводим к нижнему регистру
18  (while (not (member ?answer ?allowed-values)) do
19    ; Пока ответ недопустим
20    (printout t ?question)
21    ; Повторяем вопрос
22    (bind ?answer (read))
23    ; Снова читаем ответ
24    (if (lexeme? ?answer)
25      ; Если снова строка
26      then (bind ?answer (lowercase ?answer))))
27    ; Приводим к нижнему регистру
28    ?answer)
29  ; Возвращаем валидный ответ
30
31 (deffunction YesOrNo (?question)
32  (bind ?response (ask-question ?question да нет))
33  ; Задаем вопрос с вариантами
34  (if (eq ?response да)
35    ; Если ответ "да"
36    then TRUE
37    ; Возвращаем TRUE
38    else FALSE))
39  ; Иначе FALSE
40
41 (defrule print
42  (declare (salience 0))
43  ; Низкий приоритет выполнения
44  (repair ?item) =>
```

```

45 ; Если есть факт repair
46 (printout t crlf)
47 ; Переход на новую строку
48 (format t "%s%n%n%n" ?item))
49 ; Форматированный вывод сообщения
50
51 ; 0
52 (defrule zero-question ""
53 (not (health?)) ; Если здоровье не проверялось
54 (not (repair?)) ; И нет предыдущих ограничений
55 =>
56 (if (YesOrNo "Есть ли у ребенка хронические заболевания/
57 противопоказания? Напишите да/нет: ")
58 then
59 (assert (health right))
60 else
61 (assert (health no))))
62
63 ; +
64 (defrule musculoskeletal ""
65 (health right)
66 (not (musculoskeletal?))
67 (not (repair?)) =>
68 (if (YesOrNo "Есть ли полный запрет активной
69 деятельности? Напишите да/нет: ")
70 then
71 (assert (repair "Рекомендуемые секции: ЛФК под
72 присмотром врача."))
73 else
74 (assert (musculoskeletal no))))
75
76
77 ; +—
78 (defrule breath ""
79 (musculoskeletal no)
80 (not (breath?))
81 (not (repair?)) =>
82 (if (YesOrNo "Есть ли проблемы с опорно-двигательным
83 аппаратом? Напишите да/нет: ")
84 then
85 (assert (repair "Рекомендуемая секция: Плавание."))
86 else
87 (assert (breath no))))
88
89
90 ; +—
91 (defrule activeact ""
92 (breath no)

```

```

93 (not (activeact?))
94 (not (repair?)) =>
95 (if (YesOrNo "Есть ли дыхательные проблемы (пр. астма)?
96 Напишите да/нет: ")
97 then
98 (assert (repair "Рекомендуемые секции: Йога"))
99 else
100 (assert (repair "Рекомендуемая секция: Стретчинг, пилатес.")))
101
102
103
104 ; —
105 (defrule ageless3 ""
106 (health no)
107 (not (ageless3?))
108 (not (repair?)) =>
109 (if (YesOrNo "Ребенку меньше 3-х лет? Напишите да/нет: ")
110 then
111 (assert (repair "Рекомендуемые секции: Игровые развивающие секци
    и."))
112 else
113 (assert (ageless3 no))))
114
115 ; —
116 (defrule agemore7 ""
117 (ageless3 no)
118 (not (agemore7?))
119 (not (repair?)) =>
120 (if (YesOrNo "Ребенку больше 7-и лет? Напишите да/нет: ")
121 then
122 (assert (agemore7 right))
123 else
124 (assert (agemore7 no))))
125
126 ; —+
127 (defrule teamgames7 ""
128 (agemore7 right)
129 (not (teamgames7?))
130 (not (repair?)) =>
131 (if (YesOrNo "Есть ли интерес к командным играм?
132 Напишите да/нет: ")
133 then
134 (assert (teamgames7 right))
135 else
136 (assert (teamgames7 no))))
137
138 ; —++
139 (defrule hight ""

```

```

140 (teamgames7 right)
141 (not (hight?))
142 (not (repair?)) =>
143 (if (YesOrNo "Есть ли генетически заложенный высокий
144 рост? Напишите да/нет: ")
145 then
146 (assert (repair "Рекомендуемые секции: Волейбол, баскетбол."))
147 else
148 (assert (repair "Рекомендуемые секции: Футбол, регби.")))
149
150 ; —+—
151 (defrule fight ""
152 (teamgames7 no)
153 (not (fight?))
154 (not (repair?)) =>
155 (if (YesOrNo "Интересуют ли боевые искусства?
156 Напишите да/нет: ")
157 then
158 (assert (fight right))
159 else
160 (assert (fight no))))
161
162 ; —++
163 (defrule traditionalfight ""
164 (fight right)
165 (not (traditionalfight?))
166 (not (repair?)) =>
167 (if (YesOrNo "Интересуют ли традиционные/философские стили?
168 Напишите да/нет: ")
169 then
170 (assert (traditionalfight right))
171 else
172 (assert (traditionalfight no))))
173
174 ; —+++
175 (defrule weapon ""
176 (traditionalfight right)
177 (not (weapon?))
178 (not (repair?)) =>
179 (if (YesOrNo "Хочет ли ребенок работать с оружием?
180 Напишите да/нет: ")
181 then
182 (assert (repair "Рекомендуемые секции: Кэндо, кобудо, ушу."))
183 else
184 (assert (repair "Рекомендуемая секция: Айкидо.")))
185
186 ; —++—
187 (defrule career ""

```

```

188 (traditionalfight no)
189 (not (flexibility?))
190 (not (repair?)) =>
191 (if (YesOrNo "Есть ли у ребенка природная хорошая растяжка?
192 Напишите да/нет: ")
193 then
194 (assert (repair "Рекомендуемые секции: Тхэквондо, карате."))
195 else
196 (assert (repair "Рекомендуемые секции: Дзюдо, бокс,
197 кикбоксинг.")))
198
199
200
201 ; —+—
202 (defrule logic ""
203 (fight no)
204 (not (logic?))
205 (not (repair?)) =>
206 (if (YesOrNo "Комфортно ли ребенку заниматься деятельностью
207 без физической подвижности? Напишите да/нет: ")
208 then
209 (assert (repair "Рекомендуемые секции: Шахматы, го."))
210 else
211 (assert (logic no))))
212
213 ; —+—
214 (defrule strong ""
215 (logic no)
216 (not (strong?))
217 (not (repair?)) =>
218 (if (YesOrNo "Физически развитый ли ребенок? Напишите
219 да/нет: ")
220 then
221 (assert (repair "Рекомендуемые секции: Тяжелая атлетика,
222 гиревой спорт, пауэрлифтинг."))
223 else
224 (assert (strong no))))
225
226 ; —+—
227 (defrule feelbody ""
228 (strong no)
229 (not (feelbody?))
230 (not (repair?)) =>
231 (if (YesOrNo "Хорошо ли у ребенка развита телесная
232 координация?
233 Напишите да/нет: ")
234 then
235 (assert (repair "Рекомендуемые секции: Танцы, фехтование."))

```



```

236 else
237 (assert (repair "Рекомендуемые секции: Стрельба из лука,
238 метание.")))
239
240
241
242 ; —
243 (defrule activechild ""
244 (agemore7 no)
245 (not (activechild?))
246 (not (repair?)) =>
247 (if (YesOrNo "Активный ли ребенок? Напишите да/нет: ")
248 then
249 (assert (activechild right))
250 else
251 (assert (activechild no))))
252
253 ; —
254 (defrule hourse ""
255 (activechild no)
256 (not (hourse?))
257 (not (repair?)) =>
258 (if (YesOrNo "Любит ли лошадей? Напишите да/нет: ")
259 then
260 (assert (repair "Рекомендуемая секция: Конный спорт."))
261 else
262 (assert (hourse no))))
263
264 ; —
265 (defrule water ""
266 (hourse no)
267 (not (water?))
268 (not (repair?)) =>
269 (if (YesOrNo "Любит ли плавать? Напишите да/нет: ")
270 then
271 (assert (repair "Рекомендуемая секция: Плавание."))
272 else
273 (assert (repair "Рекомендуемые секции:
274 Керлинг, гольф, настольный теннис.")))
275
276
277 ; —+
278 (defrule run ""
279 (activechild right)
280 (not (run?))
281 (not (repair?)) =>
282 (if (YesOrNo "Любит ли бегать? Напишите да/нет: ")
283 then

```

```

284 (assert (run right))
285 else
286 (assert (run no)))
287
288 ; —++
289 (defrule communication ""
290 (run right)
291 (not (communication?))
292 (not (repair?)) =>
293 (if (YesOrNo "Хорошо ли находит язык с другими детьми?
294 Напишите да/нет: ")
295 then
296 (assert (repair "Рекомендуемая секция: Футбол."))
297 else
298 (assert (repair "Рекомендуемая секция: Легкая атлетика."))))
299
300 ; —+-
301 (defrule naturalflexibility ""
302 (run no)
303 (not (naturalflexibility?))
304 (not (repair?)) =>
305 (if (YesOrNo "Есть ли природная гибкость, растяжка?
306 Напишите да/нет: ")
307 then
308 (assert (naturalflexibility right))
309 else
310 (assert (naturalflexibility no))))
311
312 ; —+++
313 (defrule slim ""
314 (naturalflexibility right)
315 (not (slim?))
316 (not (repair?)) =>
317 (if (YesOrNo "Худощавое ли телосложение
318 (узкие плечи, видны ребра при глубоком вдохе, тонкие кости,
319 мало мышц?
320 Напишите да/нет: ")
321 then
322 (assert (slim right))
323 else
324 (assert (slim no))))
325
326 ; —+++
327 (defrule physpower ""
328 (slim no)
329 (not (physpower?))
330 (not (repair?)) =>
331 (if (YesOrNo "Физически развитый ли ребенок?

```

```

332 Напишите да/нет: ")
333 then
334 (assert (repair "Рекомендуемые секции:
335 Акробатика, спортивная гимнастика."))
336 else
337 (assert (repair "Рекомендуемая секция: Фехтование.")))
338
339 ; —+—+—+
340 (defrule swim ""
341 (slim right)
342 (not (swim?))
343 (not (repair?)) =>
344 (if (YesOrNo "Любит ли ребенок плавать? Напишите да/нет: ")
345 then
346 (assert (repair "Рекомендуемая секция: Синхронное плавание."))
347 else
348 (assert (swim no))))
349
350 ; —+—+—+
351 (defrule winter ""
352 (swim no)
353 (not (winter?))
354 (not (repair?)) =>
355 (if (YesOrNo "Интересуется ли зимними видами спорта?
356 Или катается на коньках? Напишите да/нет: ")
357 then
358 (assert (repair "Рекомендуемая секция: Фигурное катание,
359 конькобежный спорт."))
360 else
361 (assert (repair "Рекомендуемые секции: Художественная гимнастика
362 ."))))
363
364
365 ; —+—
366 (defrule rythm ""
367 (naturalflexibility no)
368 (not (rythm?))
369 (not (repair?)) =>
370 (if (YesOrNo "Есть ли чувство ритма? Напишите да/нет: ")
371 then
372 (assert (repair "Рекомендуемые секции: Бальные танцы, танцы."))
373 else
374 (assert (rythm no))))
375
376 ; —+—
377 (defrule barriers ""
378 (rythm no)

```

```

379 (not (barriers?))
380 (not (repair?)) =>
381 (if (YesOrNo "Боится ли ребенок высоты?
382 Напишите да/нет: ")
383 then
384 (assert (barriers no))
385 else
386 (assert (repair "Рекомендуемая секция: Скалолазание."))))
387
388 ; —+—
389 (defrule react ""
390 (barriers no)
391 (not (react?))
392 (not (repair?)) =>
393 (if (YesOrNo "Быстро ли ребенок реагирует
394 на неожиданные действия? Напишите да/нет: ")
395 then
396 (assert (react right))
397 else
398 (assert (repair "Рекомендуемая секция: Дзюдо."))))
399
400 ; —+—+
401 (defrule comgames ""
402 (react right)
403 (not (comgames?))
404 (not (repair?)) =>
405 (if (YesOrNo "Любит ли командные игры? Напишите да/нет: ")
406 then
407 (assert (repair "Рекомендуемая секция: Хоккей."))
408 else
409 (assert (repair "Рекомендуемая секция: Большой теннис."))))

```