МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
федеральное государственное бюджетное образовательное учреждение высшего образования
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им.В.П.АСТАФЬЕВА
(КГПУ им.В.П.Астафьева)

Институт/факультет Математики, физики и информатики
(полное наименование института/факультета/филиала)

Выпускающая(ие) кафедра(ы) Базовая кафедра Информатики и информационных технологий в образовании
(полное наименование кафедры)

Петькина Екатерина Дмитриевна

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

Тема **Обучение будущих учителей информатики логическому программированию на основе ментально-телесного подхода**

Направление подготовки 44.04.01 Педагогическое образование
(код и наименование направления)

Магистерская программа IT-технологии в образовании
(наименование программы)

ДОПУСКАЮ К ЗАЩИТЕ

Заведующий кафедрой
д.п.н., профессор Пак Н.И.
(ученая степень, ученое звание, фамилия, инициалы)

(дата, подпись)

Руководитель магистерской программы
д.п.н., профессор Пак Н.И.
(ученая степень, ученое звание, фамилия, инициалы)

(дата, подпись)

Научный руководитель
К.п.н., доцент Степанова Т.А.
(ученая степень, ученое звание, фамилия, инициалы)

(дата, подпись)

Обучающийся Петькина Е.Д.
(фамилия, инициалы)

(дата, подпись)

Красноярск 2018

Реферат

Магистерская диссертация состоит из 80 страниц, 18 рисунков, 1 таблицы, 51 источника литературы и 4 приложений.

Краткая характеристика работы

Объект исследования: процесс обучения логическому программированию.

Предмет исследования: обучение логическому программированию на основе ментально-телесного подхода.

Цель исследования: на основе ментально-телесного подхода разработать средства обучения логическому программированию, повышающие уровень развития логического стиля мышления.

В нашей работе использовались следующие **методы исследования**:

1. теоретические, а именно: изучение и анализ педагогической, психологической и предметной литературы по теме исследования, анализ теоретических и эмпирических данных;
2. эмпирические, такие как наблюдение, тестирование, педагогический эксперимент.

Научная новизна данной работы описана в следующих пунктах:

– обоснована возможность и необходимость использования ментально-телесного подхода к обучению логическому программированию в педвузе.

– уточнено понятие логического стиля мышления, построена его структурная модель, определены уровни сформированности.

– предложена диагностика уровня сформированности логического стиля мышления.

– Предложены критерии для оценки правильности составления ментальных карт;

– определены средства развития логического стиля мышления – ментальные карты, кинестетические тренажеры.

– спроектированы принципиально новые средства обучения логическому программированию – кинестетические тренажеры и ментальные карты по темам «Списки» языка программирования Prolog.

Теоретическая значимость настоящей работы заключается в уточнении понятия логического стиля мышления, построении его структурной модели и определения уровней сформированности логического стиля мышления.

Практическая значимость работы состоит в разработке системы тестов для диагностики уровня сформированности логического стиля мышления и принципиально новых средств обучения логическому программированию - ментальных карт и кинестетического тренажера по теме «Списки» на языке программирования Prolog.

Для проверки эффективности ментально-телесного подхода в обучении будущих учителей информатики и диагностики уровня сформированности логического мышления, на базе отделения информатики Института Математики, Физики и Информатики Красноярского государственного педагогического университета им. В.П. Астафьева в 2018 году среди студентов 3 курса в рамках профильного исследования была проведена частичная апробация.

Полученные результаты позволяют утверждать, что применение разработанных нами средств обучения на основе ментально-телесного подхода способствует повышению уровня сформированности логического стиля мышления и эффективно влияет на уровень понимания логического программирования на языке Prolog, что говорит о справедливости выдвинутой нами гипотезы.

По результатам исследования имеются выступления на конференциях, семинарах и 8 публикаций в журналах.

Summary

Master's thesis consist of 80 pages, 18 pictures, 1 table, 51 sources and 4 addendums.

Master's thesis brief description

Subject of research: logical programming educative process.

Subject of research: logical programming education based on mental-embodied cognition approach.

Goal of research: develop logical programming teaching techniques, based on mental- embodied cognition approach, which increase development level of logical thinking style.

In this master's thesis used following **methods of research**:

1. Theoretical, exactly: analyze and study pedagogical, psychological and thematic literature about subject of research, analyze of theoretical and empirical data:

2. Empirical, such as observation, testing, pedagogical experiment.

Scientific novelty of this master's thesis described in following point:

– reasoned possibility and necessary of using mental- embodied cognition approach to logical programming education in pedagogical university.

– clarified the concept of logical thinking style, built its structure model, defined levels of formedness.

– proposed diagnostic for logical thinking style level of formedness.

– proposed criteria for evaluate correction of making mental cards.

– defined means of development for logical thinking style, such as mental cards and kinesthetic trainers.

– developed principally new logical programming teaching techniques – kinesthetic trainers and mental cards for subject “Lists” programming language “Prolog”.

Theoretical significance of this master’s thesis concludes in clarifying the concept of logical thinking style, building its structure model, defining levels of formedness of logical thinking style.

Practical significance of this master’s thesis composes in developing test system for diagnostic logical thinking style level of formedness and principally new logical programming teaching techniques – mental cards and programming language “Prolog” kinesthetic trainers for subject “Lists”.

For efficiency checking of mental- embodied cognition in future informatics teachers education and diagnostic logical thinking style level of formedness, partially approbation was held in 2018 among third year education students of informatics department of Institute of Mathematics, Physics and Informatics of Krasnoyarsk State Pedagogical University named by V.P. Astafev, in terms of field-specific research.

Obtained results allows to claim, that using of developed logical programming teaching techniques, based on mental- embodied cognition approach, contribute to enhance logical thinking style level of formedness and effectively influence on level of understanding “Prolog” language logical programming, which says about the truth of the master’s thesis hypothesis.

About research results there are presentations on conferences and seminars, and 8 publications in magazines and conferences materials.

Оглавление

Введение 7

Глава 1. Теоретические основы ментально-телесного подхода к обучению логическому программированию 13

1.1.Особенности изучения логического программирования и языка Prolog в педвузе 13

1.2.Сущность ментально-телесного подхода к обучению программированию 23

1.3.Логическое мышление как необходимое условие успешного изучения языка Prolog 33

Выводы по первой главе 38

Глава 2. Условия реализации ментально-телесного подхода к обучению логическому программированию 39

2.1.Диагностика уровня сформированности логического мышления 39

2.2.Разработка ментальных карт и кинестетических тренажеров по теме «Списки» на языке программирования Prolog курса «Языки и методы программирования» 50

2.3.Экспериментальная проверка эффективности реализации ментально-телесного подхода 56

Выводы по второй главе 60

Заключение 61

Библиография 63

Приложения 70

Приложение 1. Материалы для диагностики уровня сформированности логического стиля мышления 70

Приложение 2. Ментальные карты по теме "Язык логического программирования Prolog" 78

Приложение 3. Кинестетический тренажер по теме "Списки в Prolog" 82

Приложение 4. Ментальные карты студентов 83

Введение

Современные образовательные стандарты предъявляют высокие требования к предметной подготовке учителя. Учитель информатики должен владеть всеми современными технологиями программирования, поэтому в курс «Языки и методы программирования» в педагогическом вузе включено изучение не только императивного программирования, которое изучается в школе, но и логического, функционального, объектно-ориентированного, параллельного и др. современных технологий программирования.

Изучение логического программирования вызывает определенные сложности у студентов в силу своих особенностей. Поэтому поиск новых подходов к разработке методических систем обучения логическому программированию является актуальной проблемой.

Ментальный подход к обучению предполагает смещение целеполагания учебного процесса в сторону развития когнитивных способностей обучаемых, в частности, если рассматривать обучение программированию – то основной целью будет являться развитие алгоритмического стиля мышления. В рамках ментального подхода к обучению программированию предполагается использование методики ментальных карт как средства развития алгоритмического мышления.

Согласно положениям такого относительно нового течения в психологии как телесный подход, немаловажную роль в формировании мышления вообще, и, алгоритмического мышления в частности играют тактильные, моторные ощущения, кинестетические каналы восприятия.

Однако в настоящее время практически не существует средств обучения, ориентированных на кинестетические каналы восприятия.

Актуальность и практический аспект данной проблемы связаны с необходимостью поиска новых эффективных подходов к обучению. Одним из перспективных видится ментально-телесный подход.

Из всего вышесказанного возникают **противоречия**:

– между требованиями ФГОС к уровню предметной подготовки будущих учителей информатики и недостаточным количеством средств и методов обучения современным парадигмам и технологиям программирования;

– между наличием методических систем обучения студентов логическому программированию и слабой проработкой вопросов их использования с учетом когнитивных способностей обучающихся;

– между возможностями ментальных и телесных технологий обучения и недостаточной соответствующей методической базой.

Выделенные нами противоречия определяют следующую **проблему:** какие средства обучения использовать при обучении логическому программированию в педвузе, чтобы сформировать необходимый для будущего учителя информатики уровень логического стиля мышления, способствующий высокому качеству обучения логическому программированию.

Объект исследования: процесс обучения логическому программированию.

Предмет исследования: обучение логическому программированию на основе ментально-телесного подхода.

Цель исследования: на основе ментально-телесного подхода разработать средства обучения логическому программированию, повышающие уровень развития логического стиля мышления.

Гипотеза:

Сформировать необходимый уровень логического стиля мышления будущих учителей информатики, обеспечивающий высокое качество обучения логическому программированию возможно, если:

– будет уточнено понятие логического стиля мышления, предложены способы его диагностики;

– обучение будет происходить с опорой на когнитивные особенности познания, учитывая процессуальную структуру мыслительных операций, основанную на информационной модели памяти;

– использовать средства обучения ЛП, отражающие мыслительный процесс решения задачи, нацеленные не только на аудиальные и визуальные, но и на кинестетические каналы восприятии - ментальные карты и кинестетические тренажеры.

В ходе исследования были поставлены следующие **задачи**:

1. Рассмотреть теоретические основы ментального подхода к обучению программированию и телесного подхода к теории мышления.

2. Выявить особенности обучения логическому программированию в педвузе.

3. Уточнить понятие логического стиля мышления, построить структурную модель, предложить средства его развития и определить способы его диагностики.

4. Разработать ментальные карты и кинестетические тренажеры по теме «Списки» языка программирования Prolog курса «Языки и методы программирования» и провести апробацию разработанных материалов.

Для решения поставленных задач использовались следующие **методы исследования**:

3. теоретические, а именно: изучение и анализ педагогической, психологической и предметной литературы по теме исследования, анализ теоретических и эмпирических данных;

4. эмпирические, такие как наблюдение, тестирование, педагогический эксперимент.

Методологическим обоснованием исследования явились работы исследователей в области психологии А.Н.Леонтьева, С.Л.Рубинштейна**,** У.Найсера, Р.Амтхауера, А.Л.Алюшина, Е.Н.Князевой, А.Ш. Тхостова, Р. Бир, Р. Брукс, Т. ван Гелдер, Э. Кларк, Ж. Лакофф, П. Маес, Э. Прем, Э. Телен, Ф. Варела; методики обучения программированию А.П.Газейкиной,

П.Б.Хорева, Г.С.Иванова, Н.И.Пака, Т.А.Степановой; в области теоретического программирования Ц.Ин, Д.Соломон, В.М.Зеленин, Н.И.Пак, В.И.Якушевич; в области методики ментальных карт Тони Бюзана и Жозефа Новака.

Теоретическая значимость настоящей работы заключается в уточнении понятия логического стиля мышления, построении его структурной модели и определения уровней сформированности логического стиля мышления.

Практическая значимость работы состоит в разработке системы тестов для диагностики уровня сформированности логического стиля мышления и принципиально новых средств обучения логическому программированию - ментальных карт по теме «Списки» и кинестетического тренажера по теме «Списки» в языке программирования Prolog.

Научная новизна данной работы описана в следующих пунктах:

– обоснована возможность и необходимость использования ментально-телесного подхода к обучению логическому программированию в педвузе;

– уточнено понятие логического стиля мышления, построена его структурная модель, определены уровни сформированности;

– предложена диагностика уровня сформированности логического стиля мышления;

– предложены критерии для оценки правильности составления ментальных карт;

– определены средства развития логического стиля мышления – ментальные карты, кинестетические тренажеры;

– спроектированы принципиально новые средства обучения логическому программированию – кинестетические тренажеры и ментальные карты по темам «Списки».

На защиту выносятся следующие **положения**:

1. Структурная модель уточненного понятия логического стиля мышления позволяет определить способы диагностики уровеня его сформированности и средства его развития.

2. Опора на ментально-телесный подход позволяет разработать средства обучения логическому программированию, учитывающие когнитивные особенности студентов

3. Использование кинестетических тренажеров и ментальных карт в обучении студентов способствует формированию их логического стиля мышления в процессе обучения логическому программированию.

Материалы исследования обсуждались на заседаниях кафедры информатики и вычислительной техники КГПУ им. В.П. Астафьева, на научно-исследовательских семинарах-вебинарах «Информационные технологии и открытое образование». Также были представлены на XIX Международном научно-практическом форуме студентов, аспирантов и молодых ученых «Молодежь и наука XXI века: Актуальные проблемы информатики и информационных технологий в образовании» (Красноярск, 22 мая 2017 г.), на XVIII Международном научно-практическом форуме студентов, аспирантов и молодых ученых, посвященному 85-летию КГПУ им. В.П. Астафьева. «Молодежь и наука XXI века: Актуальные проблемы информатики и информационных технологий в образовании» (Красноярск, 23 мая 2017 г.), на международной научно - практической конференции «Проблемы и приоритеты развития науки в XXI веке» (Смоленск, 30 декабря 2017), на международной научно-практической конференции «Внедрение результатов инновационных разработок: проблемы и перспективы» (Стерлитамак, 8 января 2018), на международной научно - практической конференции «Роль и место информационных технологий в современной науке» (Магнитогорск, 28 декабря 2017), на международной научно-практической конференции «Вопросы образования и науки» (Тамбов, 30 ноября 2017), на международной научно - практической конференции «Современные условия взаимодействия науки и техники» (Омск, 13 декабря

2017), на международной научно - практической конференции «Традиционная и инновационная наука: история, современное состояние, перспективы» (Пермь, 10 января 2018), на международной научно - практической конференции «Планирование и интерпретация результатов научно-технических исследований» (Тюмень, 11 января 2018).

По результатам исследования имеется 8 публикаций.

Глава 1. Теоретические основы ментально-телесного подхода к обучению логическому программированию

1.1.Особенности изучения логического программирования и языка Prolog в педвузе

Современные образовательные стандарты предъявляют высокие требования к предметной подготовке учителя. Учитель информатики должен владеть всеми современными технологиями программирования, поэтому в курс «Языки и методы программирования» в педагогическом вузе включено изучение всех популярных парадигм программирования, а именно не только структурного программирования, которое изучается в школе, но и логического, функционального, объектно-ориентированного, параллельного и др. современных технологий программирования.

Как обычно принято в педвузах обучение программированию будущего учителя информатики включает в себя освоение студентами сначала структурного программирования, обычно это язык Pascal, затем объектно-ориентированного, например такие языка как Java и C++, и только потом логического программирования, а именно языка Prolog. Данный вывод мы можем сделать исходя из анализа различных педагогических статей и методической литературы [7, электронный ресурс].

Язык программирования предназначен для написания компьютерных программ, которые представляют собой набор правил, позволяющих компьютеру выполнить тот или иной вычислительный процесс, организовать управление различными объектами, и т. п.

Прежде всего, языки программирования нужно классифицировать по уровням. В современной классификации языков программирования всё чаще используют деление на два уровня:

– *Языки низкого уровня.* Такими считаются языки, близкие к машинному коду и ориентированные на конкретные команды процессора, то есть машинно-ориентированные языки.

– *Языки высокого уровня.* Такими языками считаются машинно-независимые языки, разработанные для удобства пользования человеком.

Написанная на любом языке программирования программа представляет собой некий текст, не понятный компьютеру. Чтобы компьютер мог выполнить команды, которые содержит эта программа, ее необходимо перевести на язык понятный компьютеру – двоичный код. Такой перевод называется трансляцией.

Языки программирования по способу трансляции можно разделить на:

– компиляторы,

– интерпретаторы.

В компиляторах перевод всего текста программы в код осуществляется сразу, и затем создается исполняемый файл, который можно неоднократно запускать [11, с. 47].

Интерпретаторы при запуске программы каждую строчку данной программы последовательно переводят в код и выполняют; затем интерпретаторы переводят в код и выполняют другую строчку, и так далее.

Далее классифицировать языки программирования можно по парадигме программирования. Парадигма – есть совокупность идей и понятий, определяющих подход к программированию. Большинство авторов даёт чёткое разделение языков программирования на процедурные (императивные) и непроцедурные (декларативные). Также еще выделяют параллельное программирование. Класс непроцедурных языков содержит в себе два подкласса: функциональные и логические языки. Часто отдельным классом выделяют объектно-ориентированные языки программирования [22, с.43-46].

Процедурное (Императивное) программирование – Один из самых первых придуманных видов языков программирования и потому самый близкий к понятному человеку языку. Также императивное программирование часто называют директивным, из-за того что алгоритм

выполнения действий программы записывается четкой цепочкой алгоритма. И эта цепочка команд называется директивой.

Наиболее популярными императивными языками являются один из первых созданных – язык Pascal, в основном созданный в целях обучения программированию, в силу простоты своих алгоритмов, и язык С который близок в основном профессиональным программистам.

Язык программирования Pascal является чисто процедурным и достаточно простым языком. Главными особенностями данного языка можно выделить такие, как наличие средств структурного программирования и строгая типизация языка.

Объектно-ориентированное программирование (ООП) по большей степени относится к процедурному программированию, но часто выделяется в отдельный класс благодаря новым дополнениям. К этим дополнениям относятся принцип инкапсуляции данных и методов, принцип наследования, полиморфизм и абстракция данных.

Объектно-ориентированное программирование появилось вследствие того, что программисты стали задаваться вопросом, что же главнее в программе – алгоритм программы или все же данные которые обрабатывает данный алгоритм. В связи с развитием языков программирования и обрабатываемых ими данных, появилась потребность к хранению данных вместе с алгоритмами для их обработки, чтобы избежать ошибок из-за разнообразности данных. И здесь появилось понятие объекта в программировании.

Примерами объектно-ориентированных языков можно назвать такие, как Delphi, C++, Java и др.

И так, Delphi — структурированный, объектно-ориентированный язык программирования, обладающий строгой статической типизацией переменных. Основная область использования данного языка — написание прикладного программного обеспечения.

C++ — компилируемый и также строго типизированный язык программирования. C++ предоставляет все четыре возможности ООП, то есть все принципы, которые выделяют данный вид программирования от других — абстракцию, инкапсуляцию, наследование и полиморфизм.

Технология *параллельного программирования* может быть реализована в каждой из парадигм. В отличие от программирования последовательных вычислений, концептуальную основу которого составляет понятие алгоритма, реализуемого по шагам строго последовательно во времени, в параллельном программировании программа порождает совокупность параллельно протекающих процессов обработки информации, полностью независимых или связанных между собой статическими или динамическими пространственно-временными, причинно-следственными или информационными отношениями.

Параллельное программирование представляется достаточно сложным для изучения, но оно также включает в себя все черты более традиционного, последовательного программирования [25, электронный ресурс].

Примером первого языка параллельного программирования является Язык Occam. Данный язык был создан в 80х годах и предназначен для программирования транспьютеров — многопроцессорных систем распределенной обработки данных.

Главным отличием процедурного и непроцедурного программирования является то, что программа непроцедурных языков программирования четко декларирует, какую цель необходимо достигнуть, а процедурное программирование дает предписание как это сделать.

Непроцедурные (Декларативные) языки программирования – это языки объявлений и построения структур. Алгоритмические действия не прописываются программистами в языках данного типа программирования, а строятся самой программой. В непроцедурном программировании производится построение какой-либо структуры или системы, то есть декларируются какие-то свойства создаваемого объекта, благодаря которым,

программа должна будет достичь цели. Эти языки получили широкое применение в системах автоматизированного проектирования (САПР), в так называемых CAD-пакетах, в моделировании, системах искусственного интеллекта. Наиболее существенными классами декларативных языков являются функциональные языки, такие как Lisp, и логические языки, такие как Prolog, которые мы рассмотрим далее.

Функциональное программирование – это такой тип программирования, в котором трактуется процесс вычисления как вычисление математических функций. Данный вид программирования не предполагает хранения состояний программы и как следствие изменяемость этих состояний, и главным образом, руководствуется вычислением функций от заданных ему данных и от результатов вычисления других функций. На практике отличие математической функции от понятия «функции» в императивном программировании заключается в том, что императивные функции могут опираться не только на аргументы, но и на состояние внешних по отношению к функции переменных, а также иметь побочные эффекты и менять состояние внешних переменных.

Самым популярным представителем функциональных языков программирования является Lisp. Лямбда-исчисление является основой данного языка и формализмом для представления функций и способов их комбинирования [9, с. 115].

Так же можно отнести к наиболее известным языкам функционального программирования язык Haskell – превосходный язык для обучения и экспериментов со сложными функциональными типами данных и с нестрогой семантикой; Это один из активно развивающихся функциональных языков.

Логическое программирование – это вид программирования, которое основывается на аппарате математической логики и использовании принципов резолюций, то есть автоматического доказательства теорем.

Также используется механизм логического вывода информации, использующий заданные правила и факты, основанные на формальном исчислении.

Логическое программирование на ряду с функциональным, очень сильно отличаются от основных идей создания языков программирования. Логическое программирование не строится на основных принципах архитектуры фон Неймана и привычных наборах операций, основанных на последовательности преобразований, а на некой абстрактной модели. Данный тип программирования основывается главным образом на убеждении, что это компьютер должен выполнять операции, которые ему диктует человек, а не наоборот, как в стандартных языках программирования, где человек старается писать алгоритмы действий как можно ближе компьютерному языку. В идеальной задумке, предполагается, что человеку даже не требуется задавать никаких инструкций компьютеру для решения задачи, а лишь сформулировать сведения о задаче, в виде логических аксиом и гипотез, которых должно быть достаточно для решения поставленной задачи. Логическая программа, при достаточном множестве аксиом и предположений, которые ей даны, должна выполняться в виде логического утверждения, которое подлежит доказательству. Данное утверждение называется в логическом программировании целевым утверждением. Таким образом, выполнение логической программы заключается в попытке решить задачу, или доказать целевое утверждение, при использовании исходных предположений и аксиом [44, с.138].

Первые языки логического программирования были созданы в 50-60х годах на основе математического аппарата исчисления предикатов, и задумывались для использования в автоматическом доказательстве теорем. Позже эти языки дополнили некоторым множеством синтаксических функций и стали применяться на ряду с языками общего назначения. Первый такой экспериментальный интерпретатор был использован Русселом и Колмероэ в университете в 70х годах. Данному языку было дано имя Prolog

(“программирование на языке логики” – PROgramming in LOGic), и он оказал сильное влияние на разработку последующих систем.

Prolog не является единственным языком логического программирования. Существуют и другие языки логического программирования, но не столь популярные, как Prolog.

Рассмотрим некоторые из них.

Язык логического программирования KL0 (от англ. “kernel-language version 0” – ядро-язык версии 0) – язык, в основу которого положено расширение языка логического программирования Prolog. Среди его особенностей, не свойственных языку Prolog можно выделить следующие:

– более гибкая структура управления,
– многопроцессовость,
– операции с побочным эффектом,
– машинно-ориентированные операции,
– невозможность управления базами данных и таблицами имен.

Язык логического программирования ShapeUp - ещё один язык логического программирования, в основу которого положен язык Prolog, расширенный средствами сопоставления строк.

В ShapeUp образцы строк рассматриваются так же, как и термы языка Prolog, и их сопоставление возложено на процесс унификации. Таким образом, программы на ShapeUp значительно проще, чем аналогичные программы на Prolog, их легче писать и понимать. Сокращается значительно и размер программ [44, с. 113].

В языке логического программирования Prolog программа состоит из предикатов, представляющими собой функции, вырабатывающие логические значения — любой предикат содержит вычисления, которые могут быть либо истинными, либо ложными. При этом результаты вычисления предикат возвращает, только если вычисления истинны. Предикаты состоят из правил (предложений), описывающих вычисления и соединенных между собой логическими операторами И/ИЛИ, при этом

логическому И соответствует оператор запятая, а ИЛИ — оператор точка. Программы на языке Prolog могут содержать также переменные (их имена обязательно должны начинаться с большой буквы — этим они отличаются от других объектов), однако одним из базовых в логическом и функциональном программировании является принцип однократного присваивания, в соответствии с которым переменная может получить значение лишь один раз, все остальные попытки присваивания будут работать как проверка на равенство. Следствием принципа однократного присваивания является отсутствия циклических конструкций в Prolog — вместо них везде применяется рекурсия, что не снижает скорость работы программы, т.к. хвостовая рекурсия также эффективна, как и цикл [50, с. 47].

Принцип однократного присваивания не позволит изменить значение элемента массива, поэтому базовой структурой данных в логических языках программирования является связный список. При этом изменение элемента списка может реализовываться через удаление старого узла и добавление нового. При обработке списков на языке программирования Prolog учитывается их рекурсивная природа — любой список из одного или более элемента, можно разделить на первый элемент и другой список методом разделения на голову и хвост, особым видом списка здесь является пустой список. Для обозначения пустого списка используется конструкция из двух квадратных скобок — []. Основной операцией по работе со списками является разделение списка на голову (первый элемент) и хвост (новый список из остальных элементов) [28, с. 37].

Структура программы языка Prolog:

– секция domains (описание типов объектов, используемых в предикатах);

– секция predicates (определение типов используемых предикатов)

– секция clauses (перечисление фактов, правил и утверждений);

– секция goals (задание внутренней цели решаемой задачи).

Основные особенности языка Prolog:

– для представления знаний в данном виде логического программирования используются фразы Хорна;

– программа языка описывает логическую модель предметной области в виде фактов относительно свойств предметной области и отношений между этими свойствами, также правила вывода новых свойств и отношений из тех, что уже заданны;

– использование терма как единообразной структуры данных;

– отсутствие операторов присваивания, ветвлений, безусловных переходов и указателей.

– для объяснения смысла программы применяются три семантические модели: декларативная, процедурная модели и модель в виде абстрактной машины.[49, с. 20-38]

Рекурсивный вызов цели в Prolog в последовательности действий и в реализации подобен тому же вызову в обычных рекурсивных языках. Различие возникает при реализации возврата. В обычных языках, если вычисление не может быть продолжено (например, все ветви в операторе case ложны), возникает ошибка выполнения. В Prolog вычисление просто возвращается к последнему выбору и делается попытка продолжить вычисления по новому пути.

Структуры данных, которыми оперируют логические программы, - термы – соответствуют общим структурам записей в обычных языках программирования. Prolog использует очень гибкую систему организации структур данных. Подобно языку Лисп, Prolog является бестиповым языком, не содержащим объявления данных [49, с. 38-43].

Другие особенности использования структур данных в языке Prolog связаны с природой логических переменных. Логические переменные соотносятся с объектами, а не с ячейками памяти. Если переменной сопоставлен конкретный объект, то эта переменная уже никогда не может ссылаться на другой объект. Иными словами, логическое программирование

не поддерживает механизм деструктивного присваивания, позволяющий изменять значение инициализированной переменной.

Основные сложности, возникающие у студентов при работе с Prolog:

– в связи с недостаточными знаниями по математической логике неправильное составление логических выражений;

– отсутствие прямого доступа к элементам списка;

– непривычный синтаксис языка;

– неграмотное составление цели;

– непонимание составления правил остановки рекурсии.

Студенты испытывают сложности при обучении логическому программированию, и на наш взгляд, основными причинами этому являются такие как: изменение стиля мышления с алгоритмического на логический стиль мышления, а также недостаточная теоретическая разработанность методики обучения логическому программированию. Для решения данных проблем в дальнейших параграфах мы выведем уточненное понятие логического стиля мышления и представим разработанные нами средства обучения на основе ментально-телесного подхода.

1.2.Сущность ментально-телесного подхода к обучению программированию

Изучение языка программирования, относящегося к другой парадигме, вызывает определенный ряд сложностей. Так как при переходе к программированию методами, которые относятся к другой парадигме, необходимо изменить не только подход к решению поставленной задачи, но и перестроить мыслительную деятельность относительно новой парадигмы. Каждая парадигма программирования предполагает формирование определенного стиля мышления – объектного, функционального, логического, параллельного [7, электронный ресурс].

Поскольку в школьном курсе информатики изучаются языки программирования, относящиеся к императивной парадигме, можно сказать, что у студентов в начале обучения сформировано алгоритмическое мышление в узком смысле, в том понимании, которое сложилось в тот период времени, когда преобладала парадигма императивного, структурного программирования.

Изучение логического программирования вызывает определенные сложности у студентов в силу своих особенностей. Поэтому поиск новых подходов к разработке методических систем обучения логическому программированию является актуальной проблемой. Одним из перспективных видится ментально-телесный подход. Это сплав двух подходов: ментального и телесного.

В настоящее время, существует ряд российских и западных исследований, посвященных разработке методических подходов к обучению программированию, среди которых можно выделить: дифференцированный, системный, деятельностный, когнитивный, проблемный, семиотический и, интересующие нас, ментальный и телесный подходы. Рассмотрим кратко основные подходы.

Дифференцированный подход к обучению программированию. Данный подход основывается на разделении обучающихся по группам, различным по содержанию образования, методам и формам обучения. При таком подходе учитываются индивидуальные особенности обучающихся, их уровень развития и интересы, на основе которых и происходит разделение на группы.

Согласно дифференцированному подходу для каждой группы учащихся разрабатываются задания подходящие их уровню развития [20, с 46-48].

Системный подход к обучению программированию. Этот подход основан на систематизации, объекты обучения рассматриваются как системы. По данному подходу обучение проходит на основе выявления связей между объектами, закономерностями и сведении всех деталей в единую картину.

Согласно положениям системного подхода применительно к процессу обучения программированию необходимо рассматривать учебную задачу совместно со средствами, методами и процессами программирования, как единое целое с внутренними взаимосвязями и изучение всех компонентов должно происходить параллельно.

Деятельностный подход к обучению программированию. Основной идеей данного подхода является утверждение о том, что между психикой человека и его деятельностью есть неразрывная связь, так как именно деятельность формирует психику человека. Суть деятельностного подхода в обучении состоит в направлении «всех педагогических мер на организацию интенсивной, постоянно усложняющейся деятельности» [16, с. 7].

Деятельностный подход строится на организации самостоятельной учебной и познавательной деятельности обучаемых, когда сами ученики выстраивают свою деятельность и активно участвуют в процессе познания.

Когнитивный подход к обучению программированию. Основной целью данного подхода является развитие когнитивных способностей учеников, а именно таких, как быстрота мышления, готовность к решению нестандартных задач и умение воспринимать и оценивать правильно противоречивую информацию.

Применительно к программированию данный подход базируется на овладении обучающимися элементарных операций языков программирования и постепенно овладении более сложными.

Проблемный подход к обучению программированию. Основой проблемного подхода является постановка некой проблемы или проблемной ситуации, требующей решения. Такая проблема должна быть интересна учащимся, каким-то образом быть связана с их интересами или будущей деятельностью [20, с. 46-48].

Семиотический подход к обучению программированию. Суть данного подхода заключается в развитии некоторых знаково-символических действий у обучающихся. Конечно же, какие-то начальные знания символов и знаков у учеников имеются и именно используя эти знания и опыт работы с такими знаками ученики смогут освоить языки программирования, если будут рассматривать их как знаковые системы [20, с. 46-48].

Основная идея ментально-телесного подхода строится на особенностях восприятия и усвоения информации. Если взять обучение как информационный процесс, то можно сделать вывод, что он заключается в получении и обработки новой информации, пригодной для жизни и развития личности в социальной среде. Причем, новая информация должна быть понята и включена в систему имеющихся знаний человека. [5, с. 3]

Ментальный подход

Ментальный подход к обучению основывается на смещении целеполагания учебного процесса в сторону развития когнитивных способностей обучаемых. Основной особенностью ментального подхода

является направленность на развитие алгоритмического стиля мышления. Поэтому в процессе обучения логическому программированию используется методика ментальных карт, как средств развития алгоритмического мышления [40, с. 45].

Ментальный подход предлагает рассматривать обучение как процесс формирования и развития ментальных схем человека при его информационном воздействии с окружающей средой [18, с. 31].

Ментальной схемой называют набор некоторых понятий и зависимостей между ними, которые выполняют люди для решения проблемы интерпретации явлений, без какой-либо ссылки на научные знания и дисциплинарные парадигмы.

Согласно исследованиям И.В. Баженовой, Н. Бабича и Н.И. Пака ментальная схема есть модель отражения реального мира в памяти в виде пространственно-временной динамической диаграммы [3, 18 с. 32].

Использование различных подходов к программированию существенно влияет на эффективность процесса создания компьютерных программ. Так, например, процесс разработки высокоинтеллектуальной экспертной программной системы существенно упрощается при использовании логической парадигмы. В связи с этим выделяется логический стиль мышления. В основе логических языков программирования лежит формализованная логика в отличие от императивных языков программирования, ориентированных на компьютер, т. е. основной принцип состоит в том, что нужно подробно, на логически точном языке описать условие задачи. Решение ее получается в результате определенного процесса, который выполняет компьютером. В этом заключается разница между логическими языками программирования и традиционными, требующими описания процедуры решения задачи. Изучение логической парадигмы обработки информации дает новое понимание при изучении программирования, способствуя тем самым развитию у студентов иного стиля мышления, предполагающего отказ от императивных стереотипов.

Основной теорией ментального подхода считается предположение психолога Найссера, в котором говорится о том, что ментальные схемы являются главной формой мышления человека [19, с. 81-84]. Именно исследуя человеческие способности ориентации во времени и пространстве, и конечно же, способность к деятельности, психологи пришли к выводу о том, что в нашей голове создаются и хранятся разнообразные, отвечающие за все ментальные схемы. В попытках доказать это и объяснить появилось понятие ментальной карты как формы ментальной схемы.

Ментальные карты считаются одним из эффективных средств организации знаний и понятий, что также в своих работах по психологии познания упоминал Дэвид Аусубел. Также свои работы теории ментальных карт посвящал и профессор Джозеф Новак [9, эектронный ресурс]. Именно он придумал правила создания ментальных карт, как инструмента визуализации и создания новых идей или теорий. Концепция ментальных карт основана на представлениях о принципах работы человеческого мозга, а именно, таких как ассоциативное (нелинейное) мышление, визуализация мысленных образов, целостное восприятие (гештальт). Дальнейшее развитие теория ментальных карт получила в работах психолога Тони Бьюзена. Именно Тони Бьюзен превратил метод ментальных карт, о котором писал, в коммерческий продукт.

Не менее успешно, чем в коммерции, ментальные карты применяются в педагогике [8, с.3; 30, с.102]. При традиционном изучении учебного материала, как правило, активизируется лишь незначительная часть огромных возможностей мозга. Традиционно тексты учебников, содержание лекций состоит из фраз, списков и цифр. При его восприятии используются принципы запоминания, рассчитанные на функции левого полушария головного мозга, отвечающего за язык, логику, составление списков и операции с числами, и совсем не учитываются такие аспекты работы мозга, как воображение, ассоциативность, цвет, ритм и ощущения. А применение методики ментальных карт дает такие преимущества как, использование для

решения проблемы способностей обоих полушарий, и логического и интуитивного [3, с.2].

Пример простой ментальной карты можно увидеть на рисунке 1.

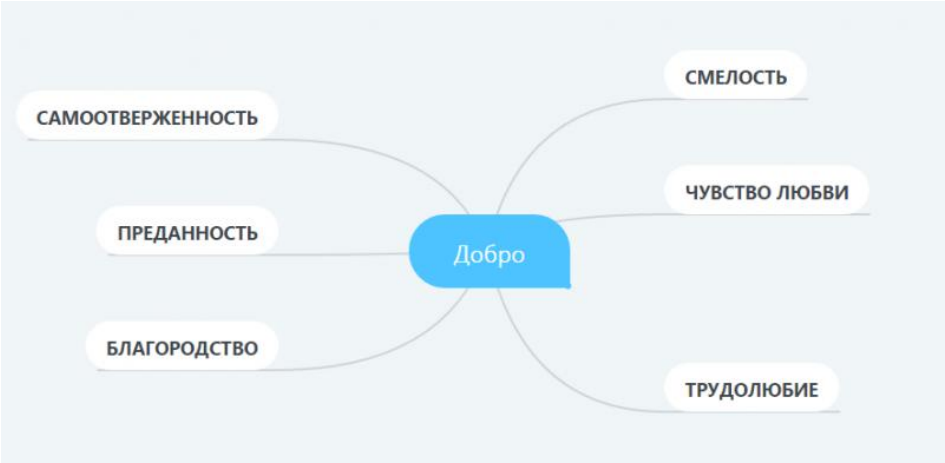


Рисунок 1. Ментальная карта

Основа метода ментальных карт в том, что изучаемая информация ставится в основе какого-то ключевого образа, являющегося главной темой материала. От этого центрального образа отходят соединительные линии, над которыми написаны или нарисованы ключевые понятия для составления образа. Эти линии в свою очередь соединены с другими, на которых расположены ключевые слова, описывающие образы или сами ключевые образы. Именно так выстраивается многомерная, ассоциативная и образная «карта памяти» (Mind Map) всего материала. Для большей наглядности и лучшего понимания можно изменять размер и стиль используемого шрифта, использовать цвет и рисунки.

Данную методику ментальных карт можно использовать в обучении на таких этапах как: объяснение новой темы, закрепление имеющихся знаний и контроль знаний и умений для выявления пробелов в понимании.

Методику ментальных карт эффективно применять и для самообучения. Ведь если все основные понятия и закономерности выстраивать в форме ментальной схемы, то материал будет усваиваться легче и быстрее. Даже при прочтении книги можно изложить основные моменты и связи этих моментов в ментальную схему и осмысление пройдет быстрее, и к

тому же через время по этой ментальной схеме вы сможете быстро вспомнить, о чем шла речь в данной книге.

Если рассматривать использование методики ментальных карт конкретно в обучении программированию, то можно сделать вывод, что здесь нас будут интересовать деятельностные ментальные схемы, поскольку ментальной картой именно таких схем является алгоритм.

В подавляющем большинстве работ, посвященных использованию ментальных карт в учебном процессе, ментальные карты представляются в виде простой семантической сети или даже семантического графа, поскольку в них только обозначены понятия как узлы семантической сети и установлены связи между понятиями (нарисованы стрелочки между узлами). Для того чтобы семантический граф превратился в семантическую сеть, надо указать еще и тип связи (чтобы эти стрелочки были подписаны). А чтобы ментальная карта, представляющая собой модель ментальной схемы, не сводилось к семантической сети, еще необходимо, чтобы в ней был отражены не только понятия и связи между ними, но и некоторые знаки, символы, активизирующие чувственные образы. Тогда она будет работать не только с модельной, понятийной и абстрактной зонами памяти, но и с чувственной зоной памяти [42, с. 95]. Только в этом случае, ментальная карта будет действительно ментальной, будет представлять собой модель ментальной схемы и будет обеспечивать более эффективное восприятие информации и более эффективное протекание процесса мышления [8, с.182].

Телесный подход.

Суть телесного подхода заключается в том, чтобы воздействовать преимущественно на кинестетические каналы восприятия, моторную область памяти, при том, что все остальные подходы к обучению воздействуют на дигитальные, аудиальные и/или визуальные каналы восприятия. Между тем, согласно телесному подходу, наше мышление «отелеснено», неразрывно связано с телесными ощущениями, и, следовательно, этот подход является единственно приемлемым для обучающихся с ведущим кинестетическим

каналом восприятия и существенно более эффективным для всех остальных, поскольку задействует все каналы восприятия.

Телесный подход начал интенсивно развиваться в западной когнитивной науке примерно с начала 1990-х годов. Английское словосочетание «embodied cognition approach» точнее было бы переводить как подход с точки зрения «отелесненности» процесса познания, телесной облеченности всякого познающего существа [2, с. 89]. Такое уточнение имеется в виду, когда говорят неуклюже, но кратко: «телесный подход». Среди создателей нового подхода такие ученые, как Р. Бир, Р. Брукс, Т. Ванн Гелдер, Э. Кларк, Ж. Лакофф, П. Маес, Э. Прем, Э. Телен, Ф. Варела и ряд других [3, с. 106]. Другими словами, телесный подход фокусирует свое внимание на телесной организации познающего существа, так как то, что познается и как познается, зависит от телесности живого существа и его конкретных функциональных особенностей.

Телесный подход говорит о единстве реальности и принадлежности к ней со свойственными ему когнитивными процессами, так и внешней ему среды. Эта единая реальность имеет физический характер, а процессы в ней являются динамическими процессами самоорганизации.

Понятие тела рассматриваются с двух противоположных точек зрения в телесном подходе. Согласно первой точке зрения познание формируется только через тело и его движение. Вторая же точка зрения, говорит о том, что тело привносит искажающие влияния, от которых следовало бы избавиться. Идея чистого интеллекта присутствует и как предмет критики, с точки зрения ее нежизненности и научной несостоятельности, и как идеал. Сам по себе недостижимый, именно он задает направленность вектора объективации. Такое противоречие можно снять в уме с помощью гегелевской диалектической триады: до помещения в тело чистый интеллект есть выхолощенное представление, но после помещения в тело и с учетом такой помещенности – на новом витке спирали, он становится важным притягивающим аттрактором и задающим критерием.

Если исходить из физического существования тела человека, то в телесном подходе имеет место быть развитие темы категоризации человеком мира. Совокупность этих категорий как бы накладывается на мир, и только согласно ее ячейкам мир и воспринимается. Категории эти не случайны и не произвольны, как полагалось ранее. Они происходят из телесного опыта, а он имеет повторяющийся и устойчивый характер [45, с. 75].

Телесный подход открывает новую перспективу на культурный мир и человека в нём, по мнению М.А. Степанова, где главное отличие проходит через смену отношения к месту и времени: истолкование традиционно происходит после события, телесное познание требует готовности к будущему и решительности в спонтанном акте его реализации. Или, иначе говоря, компетентности и способности к действию [39, с. 8].

Согласно основным положениям теории телесного подхода необходимость разработки средств обучения, которые были бы взаимосвязаны с телесными, а именно кинестетическими ощущениями и процессом развития мышления, получает теоретическое обоснование.

Особенно актуально применение подобных средств обучения в обучении техническим дисциплинам, требующим высокого уровня развития логического, алгоритмического и абстрактного мышления. Сложность аппарата технических дисциплин приводит к понижению фундаментальной базы знаний: освоение практическими навыками, как правило, проходит без глубокой осознанности сути и механизмов предмета изучения, являясь причиной выполнения ряда заданий на механическом уровне – без полного и глубокого понимания.

Обозначенная проблема актуализирует необходимость создания таких средств обучения, которые задействовали бы чувственную (сенсорную) систему человека. В условиях современных разработок здесь важен комплексный подход, в котором учитываются психолого-педагогические, дидактические, методические составляющие предметной подготовки, обеспечивающие синтез объяснительно-иллюстративных и наглядно-

образных методов обучения с позиции метального и телесного подходов. Такими средствами могут стать кинестетические тренажеры, которые позволят раскрыть изучаемые понятия или методы на тактильном уровне, погрузить обучающегося в процесс выполнения алгоритма или рассмотреть изучаемый объект с разных позиций.

Мы предполагаем, что если строить систему обучения, опираясь на ментальный и телесный подходы, использовать средства обучения, нацеленные на активизацию не только визуальной, аудиальной, абстрактной, но еще и моторной зоны памяти, процесс обучения программированию будет проходить более эффективно.

1.3.Логическое мышление как необходимое условие успешного изучения языка Prolog

Ментально-телесный подход к обучению предполагает смещение целеполагания учебного процесса в сторону развития когнитивных способностей обучаемых, в частности, если рассматривать обучение логическому программированию – то основной целью будет являться развитие логического стиля мышления.

Следовательно, для реализации процесса обучения программированию на основе ментального подхода необходимо уточнить понятие логического мышления применительно к обучению программированию на языке Prolog, поскольку ментальный подход предполагает предварительное исследование особенностей когнитивных процессов.

Мышление человека — это высшая форма познавательной деятельности человека, психический процесс обработки информации и установления связей между предметами и их свойствами. Мышление позволяет человеку находить связи между феноменами окружающей среды, но чтобы найденные связи, действительно, отражали истинное положение дел, мышление должно быть логичным, подчиненным законам логики [29, с. 63].

В соответствии с целью нашего исследования нас будет интересовать определение логики как нормативной науки о методах, формах и законах мыслительной деятельности человека. Логика изучает способы достижения истины в процессе познания опосредованным путём, из знаний полученных ранее [42, с. 94].

Логическое мышление – это мыслительный процесс, при котором человек использует логические понятия и конструкции, которому свойственна доказательность, рассудительность, и целью которого является получение обоснованного вывода из имеющихся предпосылок [43, с.16]. Логическое мышление объединяет в себе образно-логическое, абстрактно-логическое и словесно-логическое мышление (рис. 2).

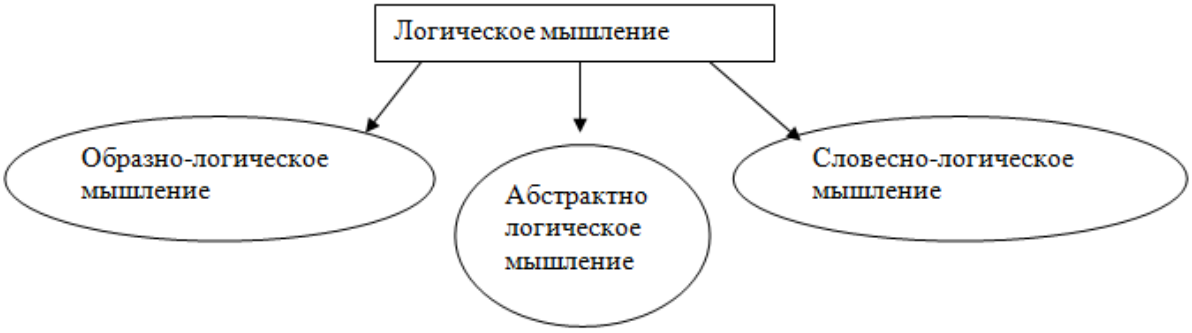


Рисунок 2. Структурная модель логического мышления

Образно-логическое мышление (наглядно-образное мышление) – различные мыслительные процессы так называемого «образного» решения задач, которое предполагает визуальное представление ситуации и оперирование образами составляющих её предметов. Наглядно-образное мышление, по сути, является синонимом слова «воображение», которое позволяет нам наиболее ярко и четко воссоздавать все многообразие различных фактических характеристик предмета или явления. Данный вид мыслительной деятельности человека формируется в детском возрасте, начиная, примерно, с 1,5 лет.

Абстрактно-логическое мышление – это совершение мыслительного процесса при помощи категорий, которых нет в природе (абстракций). Абстрактное мышление помогает человеку моделировать отношения не только между реальными объектами, но также и между абстрактными и образными представлениями, которые создало само мышление. Абстрактно-логическое мышление имеет несколько форм: понятие, суждение и умозаключение.

Словесно-логическое мышление (вербально-логическое мышление)— один из видов логического мышления, характеризующийся использованием языковых средств и речевых конструкций. Данный вид мышления предполагает не только умелое использование мыслительных процессов, но и грамотное владение своей речью. Словесно-логическое мышление необходимо нам для публичных выступлений, написания текстов, ведения

споров и в других ситуациях, где нам приходится излагать свои мысли при помощи языка [43, с. 21].

Логическое мышление - это очень широкое понятие. Цель нашего исследования - выделить в нем тот аспект, который необходим для успешного обучения логическому программированию, программированию на языке Prolog.

Особенности логического программирования заключаются в том, что если в императивном программировании программа описывается как последовательность действий, то в логическом – это набор фактов (аксиом), правил (механизмов вывода нового знания), целей и подцелей (теорем, требующих доказательства или опровержения).

Программа на языке Prolog состоит из описания типов используемых в ней предикатов, задания фактов и правил, а также вопроса о том, что нас интересует и является решением логической задачи. Вопрос - это последовательность, состоящая из одной цели или нескольких подцелей. Чтобы ответить на вопрос, Prolog пытается достичь цели. Другими словами, достичь цели - значит показать, что она логически следует из фактов и правил программы.

Особенности языка Prolog — что это язык, основанный на логике предикатов, структура и особенности программы на Prolog, из чего она состоит, и отсюда выделяются этапы решения задачи на Prolog и мыслительные операции, соответствующие каждому этапу — это и будет схема логического мышления, уточненного применительно к процессу обучения логическому программированию (рис.3).

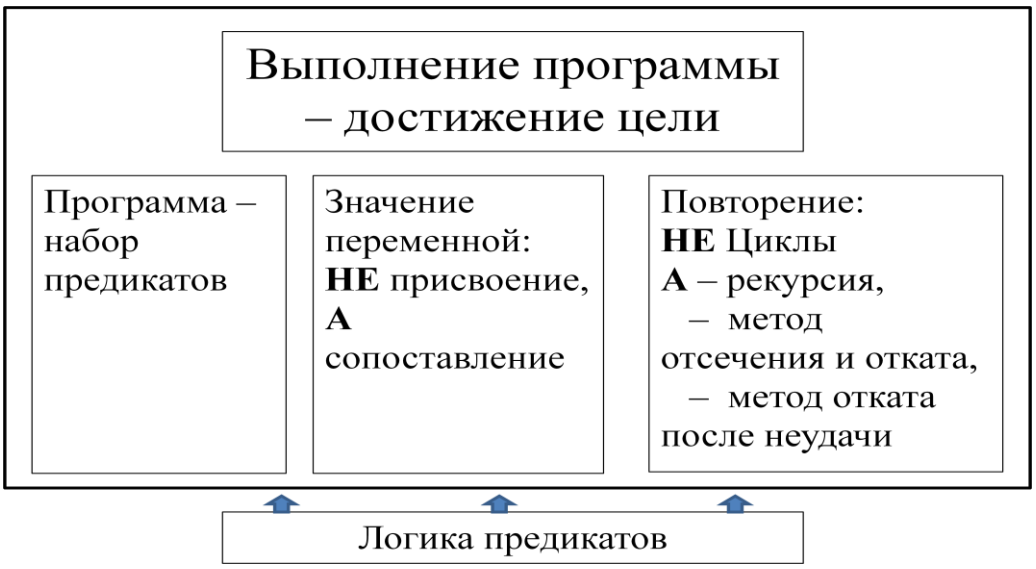


Рисунок 3. Структура логического аспекта программистского мышления

Логический аспект программистского мышления, формирующийся в процессе изучения логического программирования, предполагает способность увидеть желаемый результат выполнения программы как некоторую цель, которую необходимо достигнуть, и осознать, что сама программа – это набор предикатов [29, с. 64].

Программируя в логике, следует понимать, что в Prolog нет:

– оператора присваивания значения переменной, которое сохранялось бы в ячейке с ее именем до присваивания ей нового значения, как это имеет место в алгоритмических языках программирования;

– операторов цикла, но широко используется повторение. Организация повторов базируется на использовании метода отката после неудачи, методе отсечения и отката, определяемом пользователем, и обобщенном правиле рекурсии;

– что переменная получает свое значение в результате сопоставления, сохраняется только до тех пор, пока переменная остается связанной, т.е. участвует в цепочке операций сопоставления при поиске решения. После этого она становится свободной и не имеющей значения.

Исходя из того, что логическое программирование основано на математической логике, на логике предикатов, для начала формирования логического аспекта программистского мышления необходимы

определенный уровень сформированности математического мышления, а именно - владение основами математической логики, логикой высказываний и логикой предикатов, понимание сути операций над высказываниями, умение составлять сложные логические высказывания из простых при помощи логических связок, способность свободно оперировать логическими формулами.

Уточненное применительно к процессу обучения логическому программированию понятие логического стиля мышления позволит построить систему диагностики уровня его сформированности, с помощью которых будет выявлен необходимый для успешного усвоения учебного материала уровень его сформированности, а также позволит разработать эффективные средства его развития.

Выводы по первой главе

В первой главе нами рассмотрена и описана сущность ментально-телесного подхода к обучению логическому программированию и к теории мышления. Выявлено, что основной особенностью ментального подхода является направленность на развитие алгоритмического стиля мышления. Поэтому в процессе обучения логическому программированию должна использоваться методика ментальных карт, как средств развития алгоритмического мышления. Учитывая положения телесного подхода о влиянии на обучение кинестетических каналов восприятия информации, в процессе обучения также необходимо задействовать кинестетические тренажеры.

Проведен краткий обзор по парадигмам программирования, описаны особенности обучения логическому программированию и выделены основные сложности, возникающие при обучении логическим языкам программирования у студентов. Представлены и описаны основные подходы к обучению программированию.

Уточнено понятие логического стиля мышления применительно к процессу обучения логическому программированию и построена его структурная модель, что позволит разработать систему диагностики уровня сформированности логического стиля мышления.

Глава 2. Условия реализации ментально-телесного подхода к обучению логическому программированию

2.1.Диагностика уровня сформированности логического мышления

Как мы выяснили ранее, для успешного обучения логическому программированию необходимым условием является определенный уровень сформированности логического мышления. Следовательно, для разработки эффективных методик обучения логическому программированию необходимо составить и провести соответствующие диагностики для выявления уровня его сформированности.

Проанализировав работы посвященные уточнению понятий логического мышления в программировании, таких авторов как Газейкина А.П., Нигматулина Э.А. и Степановой Т.А. можно говорить о том, что изучение логической парадигмы дает новое понимание при изучении программирования, способствуя развитию у студентов иного стиля мышления, предполагающего отказ от императивных стереотипов [7, 23, 27, 35, 42].

Зная тот факт, что в основе логических языков программирования лежит формализованная логика, что отличает их от императивных языков, ориентированных на компьютер, можем предположить, что строить диагностики выявления уровня логического мышления необходимо на основе математической логики. Но и от вопросов на житейскую логику отказываться не стоит.

Для определения уровня сформированности логического мышления измеряют способность к выполнению следующих логических операций:

– Сравнение. Относительно простая логическая операция, она заключается в установлении сходства или различия предметов по признакам.

– Анализ. Логический прием, заключающийся в разделении предмета на отдельные части; анализ проводится для выделения признаков, характеризующих данный предмет или группу предметов.

– Синтез. Можно охарактеризовать как мысленное соединение частей предмета в единое целое с учетом их правильного расположения в предмете.

– Упорядоченность действий. Логический прием, формирующий навыки последовательных действий.

– Классификация. Распределение предметов по группам на основании общих признаков. Данная классификация включает два логических действия: выделение общего признака и деление на классы [42, с. 36].

Нами предлагаются следующие уровни сформированности логического мышления:

– Начальный уровень. Подразумевает способность к сравнению и классификации;

– Средний уровень. Включает в себя способность к упорядочиванию действий;

– Высокий уровень. Подразумевает способность ко всем логическим операциям, включая анализ и синтез.

Исходя из всего вышесказанного, мы подобрали и разработали систему для диагностики уровня сформированности логического мышления. Данная система состоит из трех различных тестов, имеющих по два варианта. Тесты разнообразны и затрагивают как вопросы о простой житейской логике, так и вопросы на математическую логику, и то и то способствует формированию логического мышления человека, а также интересующего нас, логического стиля мышления применительно к программированию.

Первый тест состоит из логических задач. Содержит два варианта по 9 вопросов.

Каждая задача состоит из двух логических посылок. Буквы в них находятся в каких-то численных взаимоотношениях между собой. Опираясь на эти логические предпосылки, надо решить, в каком соотношении находятся между собой буквы, стоящие под чертой при помощи математических знаков «>» или «<». Оценка производится по количеству правильных ответов. Задания приведены на рисунках 4 и 5.

1 вариант

1	А больше Б в 9 раз Б меньше В в 4 раза В А	2	А меньше Б в 10 раз Б больше В в 6 раз А В
3	А больше Б в 3 раза Б меньше В в 6 раз В А	4	А больше Б в 4 раз Б меньше В в 3 раза В А
5	А меньше Б в 3 раза Б больше В в 7 раз А В	6	А больше Б в 9 раз Б меньше В в 12 раз В А
7	А больше Б в 6 раз Б больше В в 7 раз А В	8	А меньше Б в 3 раза Б больше В в 5 раз В А
9	А меньше Б в 10 раз Б больше В в 3 раза В А		

Рисунок 4. Тест на определение уровня логического мышления, 1 вариант

2 вариант

1	А меньше Б в 3 раза Б больше В в 4 раза В А	2	А больше Б в 2 раза Б меньше В в 5 раз А В
3	А меньше Б в 5 раз Б больше В в 6 раз В А	4	А меньше Б в 5 раз Б меньше В в 2 раза А В
5	А больше Б в 4 раза Б меньше В в 3 раза В А	6	А меньше Б в 5 раз Б больше В в 3 раза А В
7	А больше Б в 4 раза Б меньше В в 7 раз В А	8	А больше Б в 3 раза Б меньше В в 5 раз А В
9	А меньше Б в 2 раза Б больше В в 8 раз А В		

Рисунок 5. Тест на определение уровня логического мышления, 2 вариант

Ответы на тест представлены на рисунке 6. Каждый правильный ответ оценивается в один балл.

1 вариант	2 вариант
1. B < A	1. B < A
2. A < B	2. A < B
3. B > A	3. B < A
4. B < A	4. A < B
5. A > B	5. B < A
6. B > A	6. A < B
7. A > B	7. B > A
8. B < A	8. A < B
9. B > A	9. A > B

Рисунок 6. Ключ к тесту

Второй тест основан на оценке текстовой информации. Содержит два варианта по 6 вопросов. Необходимо просмотреть утверждения и выбрать логически верный ответ из предложенных (рис. 7 и 8).

Данный тест направлен именно на определение способности к логическому мышлению, а не на знание математической логики. В нем необходимо определить формальную правильность того или иного логического умозаключения на основе определенного утверждения (или ряда утверждений), причем реальная действительность не играет при этом никакой роли. Это немного усложняет тест, поскольку содержание утверждений абсурдно, но зато логически безупречно. Абсурдность утверждений побуждает пользоваться не житейской, а формальной логикой, что более действенно выявляет способность к логическому мышлению.

1 вариант

1. Некоторые улитки являются горами. Все горы любят кошек. Значит, все улитки любят кошек.

а) правильно

б) неправильно

2. Все крокодилы умеют летать. Все великаны являются крокодилами. Значит, все великаны могут летать.

а) правильно

б) неправильно

3. Некоторые головки капусты - паровозы. Некоторые паровозы играют на рояле. Значит, некоторые головки капусты играют на рояле.

а) правильно

б) неправильно

4. Две поляны никогда не похожи одна на другую. Сосны и ели выглядят совершенно одинаково. Значит, сосны и ели не являются двумя полянами.

а) правильно

б) неправильно

5. Каждый квадрат круглый. Все квадраты красные.

а) бывают квадраты с красными углами

б) бывают квадраты с круглыми углами

в) бывают круглые красные углы

г) углы и квадраты - круглые и красные

д) ни одно из вышеперечисленных

6. Хорошие начальники падают с неба. Плохие начальники могут петь.

а) Плохие начальники летят с неба вниз.

б) Хорошие начальники, которые умеют летать - могут петь.

в) некоторые плохие начальники не могут петь.

г) некоторые хорошие начальники -плохие, так как они умеют петь.

д) ни одно из вышеперечисленных

Рисунок 7. Вариант 1 теста на логические суждения

2 вариант

1. Никто из людей не может стать президентом, если у него красный нос. У всех людей нос красный. Значит, никто из людей не может стать президентом.

а) правильно

б) неправильно

2. Все вороны собирают картины. Некоторые собиратели картин сидят в птичьей клетке. Значит, некоторые вороны сидят в птичьей клетке.

а) правильно

б) неправильно

3. Только плохие люди обманывают или крадут. Вера - хорошая.

а) Вера не крадет

б) Вера обманывает и крадет

в) ни одно из вышеперечисленных

4. Все воробьи не умеют летать. У всех воробьев есть ноги.

а) без ног воробьи не могут летать

б) все воробьи, у которых есть ноги, не могут летать

в) воробьи не могут летать, потому что у них есть ноги

г) ни одно из вышеперечисленных

5. Некоторые люди - европейцы. Европейцы имеют три ноги.

а) люди с двумя ногами не являются европейцами

б) европейцы с двумя ногами иногда являются людьми

в) Люди имеют три ноги потому что они европейцы

г) ни одно из вышеперечисленных

6. Цветы - это зеленые звери. Цветы пьют кофе.

а) все зеленые звери пьют кофе

б) все зеленые звери являются цветами

в) некоторые зеленые звери пьют кофе

г) Зеленые звери не пьют кофе

д) зеленые звери не являются цветами

е) ни одно из вышеперечисленных

Рисунок 8. Вариант 2 теста на логические суждения

Ответы к данному тесту представлены на рисунке 9. Каждый правильный ответ оценивается в 1 балл.

1 вариант	2 вариант
1. б	1. а
2. а	2. б
3. б	3. а
4. а	4. б
5. д	5. а
6. д	6. в

Рисунок 9. Ответы к тесту

Третий тест направлен на выявление знаний по математической логике. Так как студенты уже прошли курс математической логики на первых курсах, то мы смело можем давать им этот тест и надеяться хотя бы на средний результат. Ведь многое при работе с языком программирования Prolog зависит от знания и понимания основ математической логики. Тест состоит из 2 вариантов по 5 вопросов. Каждый правильный ответ оценивается в 2 балла (рис. 10,11).

1 вариант

1) $A \vee B = B \vee A$
   а) Да
   б) Нет

2) Выбрать равенство двойственное данному: A(AUB) = A
   а) $A(\overline{A}\,UB) = AB$
   б) AUAB = A
   в) A(AUB) = A
   г) $ABUA\overline{B} = A$

3) Выбрать операцию алгебры логики, задаваемую таблицей истинности:

а	в	с
1	1	1
1	0	0
0	1	0
0	0	0

$а) c = a \vee в \quad б) c = a \Leftrightarrow в \quad в) c = a \wedge в \quad г) c = a \Rightarrow в$

4) Найти: $|AUB|$ *если* $|A| = 16 \quad |B| = 8 \quad |AB| = 5$
   а) 14
   б) 22
   в) 19
   г) 18

5) Обозначим через a высказывание «Летом я поеду в деревню», а через b- «Летом я поеду в туристическую поездку».. Тогда высказывание c - ««Летом я поеду в деревню или в туристическую поездку» запишем так
   $а) c = a \vee в \quad б) c = a \Leftrightarrow в \quad в) c = a \wedge в \quad г) c = a \Rightarrow в$

Рисунок 10. Тест на математическую логику, 1 вариант

2 вариант

1) $A \wedge B = B \wedge A$
   а) Да
   б) Нет

2) Выбрать равенство двойственное данному равенству: AUAB = A
   а) (UB) = AB
   б) AUAB = A
   в) A(AUB) = A
   г) ABUA = A

3) Какая формула тождественна $x \leftrightarrow y$
   а) $\bar{x} \wedge \bar{y}$
   б) $\bar{x} \vee \bar{y}$
   в) $\bar{x} \vee y$;
   г) $(x \to y) \wedge (y \to x)$

4) Выбрать операцию алгебры логики, задаваемую таблицей истинности:

а	в	с
1	1	1
1	0	0
0	1	1
0	0	1

а) $c = a \vee в$ *б)* $c = a \Leftrightarrow в$ *в)* $c = a \wedge в$ *г)* $c = a \Rightarrow в$

5) Обозначим через a высказывание «пришла весна»; а через b- «грачи прилетели». Тогда высказывание c - «пришла весна, и грачи прилетели» запишем так

а) $c = a \vee в$ *б)* $c = a \Leftrightarrow в$ *в)* $c = a \wedge в$ *г)* $c = a \Rightarrow в$

Рисунок 11. Тест на математическую логику, 2 вариант

Ответы на данный тест представлены на рисунке 12. Каждый правильный ответ оценивается в 2 балла.

1 вариант	2 вариант
1. а	1. а
2. в	2. б
3. в	3. г
4. б	4. б
5. в	5. а

Рисунок 12. Ответы к тесту на математическую логику

По результатам всех трех тестов мы можем сделать вывод об уровне сформированности логического мышления. Для этого каждому уровню мы соотнесли количество баллов, которое необходимо набрать за каждый тест (табл. 1). Для более точного определения общего уровня сформированности логического мышления мы предлагаем высчитать процентное соотношение баллов для каждого студента.

Таблица 1. Критерии оценки уровня логического мышления

Уровни сформированности логического мышления	Тест 1 (Кол-во баллов)	Тест 2 (Кол-во баллов)	Тест 3 (Кол-во баллов)	Общий % баллов за все три теста
Начальный	1-3 балла	1-2 балла	1-3 балла	До 30
Средний	4-6 баллов	3-4 балла	4-6 баллов	От 30 до 60
Высокий	7-9 баллов	5-6 баллов	7-10 баллов	От 60 до 100

Можем предположить, что предварительно проведеннная диагностика уровня сформированности логического мышления позволит определить, какие средства обучения необходимы каждому учащемуся, чтобы сделать процесс его обучения наиболее эффективным. Для студентов с низким уровнем логического стиля мышления мы дополним процесс обучения кинестетическим тренажером и ментальными картами. Студентам со средним уровнем, как мы предполагаем, будет достаточно использовать в ходе объяснения учебного материала ментальные карты. А студентам с высоким уровнем логического стиля мышления дополнительные средства обучения не требуются.

Итоговая же диагностика позволит ответить на вопрос, насколько повысился уровень сформированности логического стиля мышления после обучения логическому программированию с использованием наших средств ментально-телесного подхода, и тем самым подтвердить нашу гипотезу.

2.2.Разработка ментальных карт и кинестетических тренажеров по теме «Списки» на языке программирования Prolog курса «Языки и методы программирования»

Как мы уже говорили ранее, изучение логического программирования вызывает у студентов сложности в связи перехода на изучение абсолютно другой парадигмы программирования. И для того чтобы решить данную проблему, мы предложили использовать ментально-телесный подход. Для реализации данного подхода, нам необходимо разработать ментальные карты и кинестетические тренажеры.

В качестве примера, возьмем тему «Списки» языка логического программирования Prolog и составим ментальную карту для поиска элемента в списке.

Для составления ментальной карты будем использовать on-line сервис Bubbl.us. Данный сервис является бесплатным, также позволяет работать над картой нескольким пользователям, готовую интеллект-карту можно сохранить в формате jpg ,png, html.

В языке программирования Prolog поиск элемента в списке представляет собой просмотр списка на предмет выявления соответствия между элементом данных (объектом поиска) и элементом просматриваемого списка. Если такое соответствие найдено, то поиск заканчивается успехом. В противном случае, поиск заканчивается неудачей. Просмотр списка осуществляется следующим образом. Единственная возможность работы со списком, предусмотренная в Prolog – это деление его на «голову» и «хвост». От исходного списка отделяется голова, которая и сравнивается с искомым элементом, затем это же действие повторяется с хвостом списка. Для организации повторения используется рекурсия.

Алгоритм решения данной задачи изображен на ментальной карте на рисунке 13.

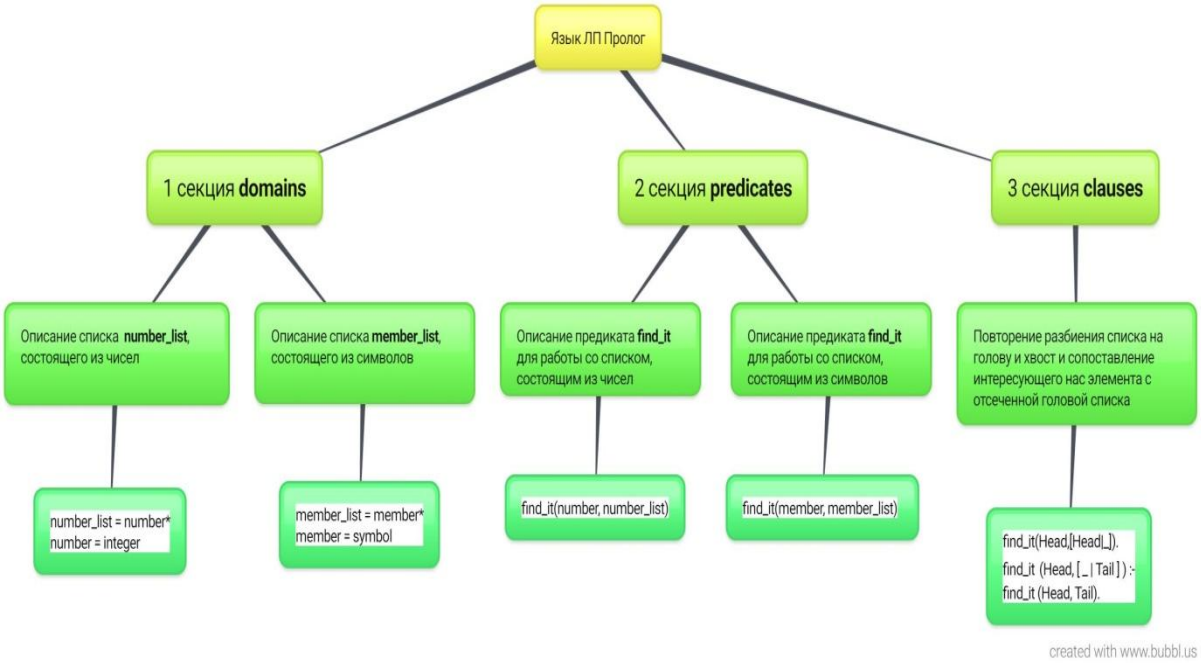


Рисунок 13. Ментальная карта «Поиск элемента в списке»

На первом уровне нашей ментальной карты представлены основные секции программы языка логического программирования Prolog и пронумерованы в соответствии со своим появлением в программе.

Вторым уровнем мы поместили словесное описание действий, которые необходимо выполнить в данной секции для решения нашей задачи.

На третьем уровне записаны фрагменты программного кода каждой секции, необходимые для решения задачи.

Записав фрагменты кода в порядке, указанном нумерацией на ментальной карте, получаем программу "Поиск элемента в списке" (рис.14), которая реализует операцию поиска элемента в списке.

```
domains
number_list = number*
number = integer
member_list = member*
member = symbol
predicates
find_it(number, number_list)
find_it(member, member_list)
clauses
find_it(Head,[Head|_]).
find_it(Head, [_ | Tail ] ) :- find_it(Head, Tail).
```

Рисунок 14. Программа, реализующая операцию поиска элемента в списке на языке программирования Prolog

Аналогично мы разработали и другие ментальные карты, которые представлены в приложениях.

Представленная ментальная карта наглядно активизирует чувственную зону памяти, иллюстрирует мыслительные процессы. Создавая такие карты, мы отражаем процесс нашего мышления. Кроме того, при самостоятельном решении задач на лабораторной работе, полезно предложить студентам предварительно составить ее ментальную алгоритмическую карту. Наглядно изображая процесс мышления, протекающий при решении задачи, можно увидеть пробелы, недостающую для решения задачи информацию и т.п.

Согласно телесному подходу для повышения эффективности обучения логическому программированию стоит задействовать кинестетические каналы восприятия с помощью кинестетических (натурных) тренажеров.

Кинестетический тренажер представляет собой некое приспособление, изготовленное преподавателем или самими студентами, позволяющее реализовать алгоритм решения задачи.

Такие средства обучения позволяют раскрыть изучаемые понятия или алгоритмы на тактильном уровне, погрузить учащегося в процесс

прохождения алгоритма или рассмотреть изучаемый объект с разных позиций. Иначе говоря, используя данные средства обучения, учащийся может сначала «вручную» разобрать смысл изучаемого процесса или понятия, а затем на основе выявленных действий построить алгоритм, создать информационную модель или выполнить другое абстрактное действие.

Мы попытались спроектировать такие тренажеры для изучения темы «Списки» языка логического программирования Prolog.

При объяснении тем связанных с работой со списками студентам необходимо напоминать о том, что списки всегда разбиваются на голову и хвост, где голова это всегда, как максимум, один элемент, а хвост это оставшиеся элементы, которые становятся новым списком. Каждая голова списка присваивается переменной, над которой производится операция, заложенная в данной программе.

Рассмотрим программу деления исходного списка на два новых. Допустим, мы имеем список L [30,50,20,25,65,95] и компаратор (критерий деления списка) Middle со значением 40. Данный список необходимо разделить на два новых – L1 и L2 [49, стр. 169].

Опишем разработанный нами тренажер. Исходный список представляет собой кубики разного размера в соответствии с присвоенными им значениями, соединенными друг с другом последовательно. За два новых будущих списка отвечают два пустых ящика с именами L1 и L2. Компаратором Middle является коробочка размером соответствующим присвоенному ей значению (рис 15).

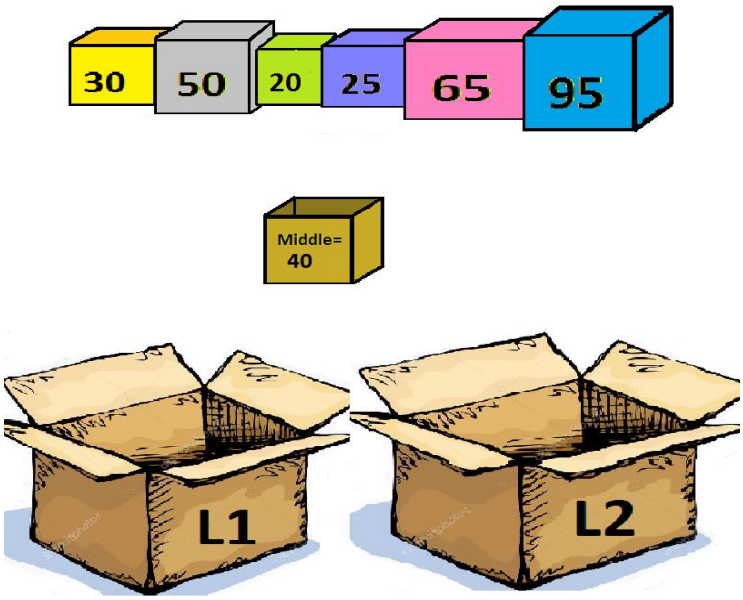


Рисунок 15. Проект разработанного кинестетического тренажера

Для того чтобы разделить список на два новых, для начала, нам необходимо отделить первый элемент исходного списка (первый кубик), который теперь становится головой, а оставшийся список (соединенные кубики) – хвостом (рис 16). Этот кубик мы пытаемся поместить в коробочку (компаратор), если кубик в нее вошел, то мы кладем кубик в первый ящик, это означает что данный элемент списка, за который отвечает этот кубик, меньше либо равен значению компаратора и теперь по условию программы является первым элементом нового списка L1. Если же кубик не помещается в компаратор, это означает, что данный элемент списка больше значения компаратора и, следовательно, становится элементом второго списка L2 и помещается во второй ящик.

Далее каждый кубик исходного списка поочередно сравнивается с компаратором и помещается в подходящий ему ящик следом за уже имеющимися там элементами (кубиками), пока не закончатся все кубики в исходном списке. Тогда программа считается завершенной. В результате в двух ящиках L1 и L2 окажутся по три соединенных кубика, то есть два новых списка - L1 [30,20,25] и L2 [50,65,95].

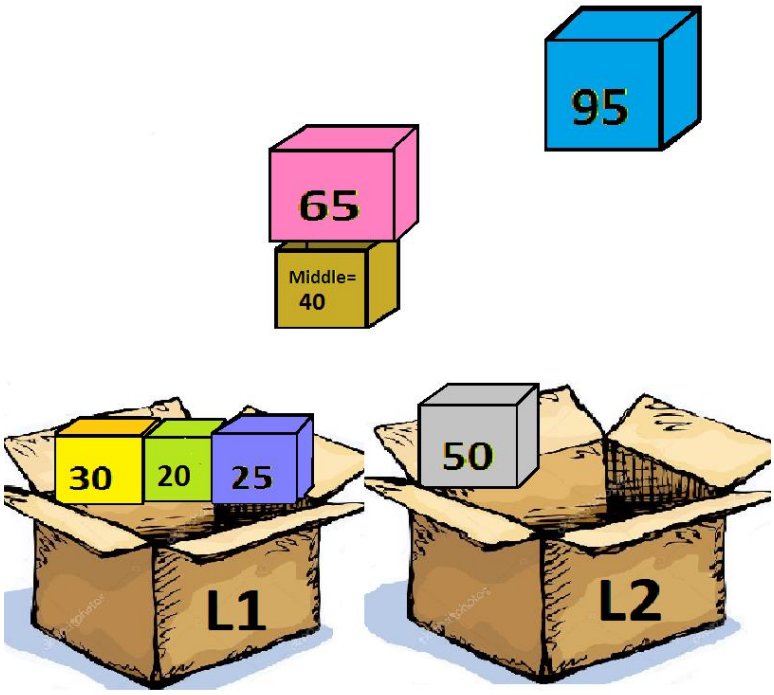


Рисунок 16. Деление списка с помощью разработанного кинестетического тренажера

Подобный тренажер можно использовать для наглядной иллюстрации и других задач языка Prolog. Например, для поиска конкретного элемента списка, трассировки списка и др.

Использование такого рода кинестетических тренажеров и ментальных карт, по нашему мнению, позволит повысить уровень понимания и качество усвоения материала дисциплины «Языки и методы программирования» не только в сфере логического программирования, а также применительно и к другим видам программирования.

2.3.Экспериментальная проверка эффективности реализации ментально-телесного подхода

Для проверки эффективности ментально-телесного подхода в обучении будущих учителей информатики и диагностики уровня сформированности логического мышления, на базе отделения информатики Института Математики, Физики и Информатики Красноярского государственного педагогического университета им. В.П. Астафьева в 2018 году среди студентов 3 курса в рамках профильного исследования была проведена частичная апробация.

Целью педагогического эксперимента являлась проверка достоверности выдвинутой нами гипотезы, а также проверки эффективности действия ментально-телесного подхода.

В ходе апробации нами были использованы такие методы исследования как анализ, обобщение, наблюдение и анкетирование.

Перед началом изучения тем языка программирования Prolog с помощью ментально-телесного подхода было проведено тестирование, определяющее уровень сформированности логического стиля мышления. Данное тестирование на определение уровня сформированности логического мышления позволит определить, какие средства обучения необходимы каждому учащемуся, чтобы сделать процесс его обучения наиболее эффективным. Для студентов с низким уровнем логического стиля мышления мы дополним процесс обучения кинестетическим тренажером и ментальными картами. Студентам со средним уровнем, как мы предполагаем, будет достаточно использовать в ходе объяснения учебного материала ментальные карты. А студентам с высоким уровнем логического стиля мышления дополнительные средства обучения не требуются.

Входное тестирование включало в себя систему диагностики представленную ранее, состоящую из трех различных тестов.

Общее время, необходимое на выполнение трех тестов, включая инструктаж перед началом тестирования, составляет 60 минут. Студентам

выдаются бланки тестов, состоящие из двух вариантов, в которых необходимо выбрать правильные ответы.

Перед проведением входной диагностики студентам объясняется цель ее проведения и проводится инструктаж по выполнению тестов, после ответа на вопросы студентов, если они возникли, начинается тестирование.

После выполнения тестов, для того чтобы проверить начальные знания по языку программирования Prolog и уровень логического мышления студентам предлагается составить ментальную карту по простейшей задаче для вывода на экран списка. В данной ментальной карте необходимо отразить ход решения данной задачи, который видит студент. На данное задание отводится 30 минут.

Правильность ментальной карты мы определяем путем экспертной оценки. Для этого мы разработали следующие критерии:

1. Соответствие структуре программы. Отражена ли в ментальной карте структура программы языка Prolog, и насколько она правильная;
2. Установление верных взаимоотношений. Прослеживается ли логика и последовательность связей в ментальной карте;
3. Уровень детализации. Оценить общее количество элементов на ментальной карте.

Соответствие одному критерию – есть один балл. Для выявления с уровня сформированности логического мышления по составленной ментальной карте мы предполагаем, что:

– Начальному уровню соответствует один набранный балл;
– Среднему уровню соответствует два набранных балла;
– Высокому уровню соответствуют три набранных балла, то есть соответствие всем трем критериям.

По окончании наших занятий на основе ментально-телесного подхода, с использованием разработанных нами средств обучения, проводится контроль знаний. Для этого студентам предлагается решить другой вариант

тестов, используемых при входном контроле. Также предлагается составить ментальную карту уже на решение более сложной задачи по языку Prolog. На выполнение итоговой диагностики отводится чуть меньше времени, чем требовалось на выполнение входной.

С помощью этого контроля выявляется уровень усвоения знаний, и насколько изменился уровень сформированности логического мышления. Начальные результаты диагностики уровня сформированности логического стиля мышления и уровня усвоения предметных знаний сопоставляются с результатами диагностики, проводимой по окончании наших занятий.

Итоги обработки результатов тестовых диагностик представлены на рисунке 17.

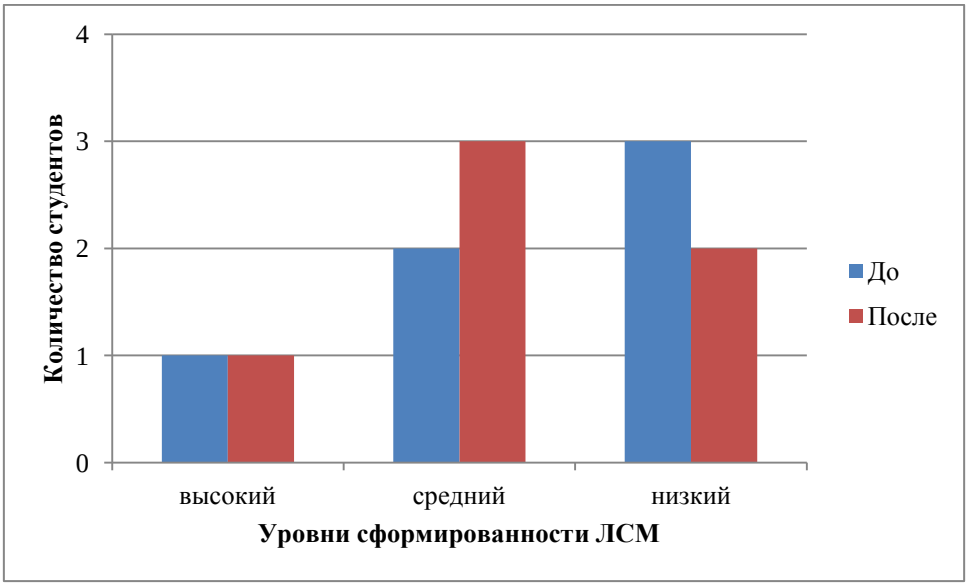


Рисунок 17. Уровень сформированности ЛСМ измеренный с помощью тестовых заданий до и после внедрения подхода

Итоги обработки результатов по выполнению ментальных карт представлены на рисунке 18.

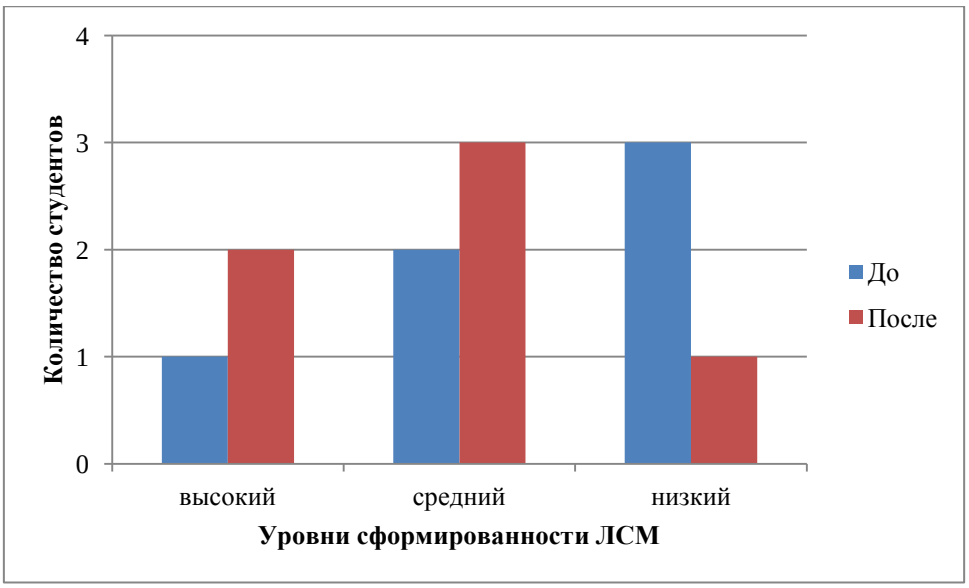


Рисунок 18. Уровень сформированности ЛСМ измеренный с помощью ментальных карт до и после внедрения подхода

Полученные результаты позволяют утверждать, что применение разработанных нами средств обучения на основе ментально-телесного подхода способствует повышению уровня сформированности логического стиля мышления и эффективно влияет на уровень понимания логического программирования на языке Prolog, что говорит о справедливости выдвинутой нами гипотезы.

Выводы по второй главе

В данной главе описана разработанная нами система диагностики, состоящая из тестовых заданий и использования ментальных карт для определения уровня сформированности логического стиля мышления. Результаты апробации позволяют утверждать, что с помощью данной диагностики можно адекватно оценить уровень развития логического мышления студентов и подобрать для каждого необходимые средства обучения.

Также в данной главе приводится описание разработанной ментальной карты и кинестетического тренажера по теме: «Списки» языка Prolog, которые использованы в учебном процессе по курсу «Профильное исследование».

Для доказательства гипотезы исследования проведено сравнение уровня сформированности логического стиля мышления студентов обучающихся языку Prolog до использования разработанных нами средств на основе ментально-телесного подхода и после обучения языку логического программирования с применением наших средств.

Заключение

В результате проведенного исследования достигнута цель и выполнены все поставленные задачи, а именно:

1. Рассмотрена и описана сущность ментально-телесного подхода к обучению логическому программированию и к теории мышления. Выявлено, что основной особенностью ментального подхода является направленность на развитие алгоритмического стиля мышления. Поэтому в процессе обучения логическому программированию используется методика ментальных карт, как средств развития алгоритмического мышления. Учитывая положения телесного подхода о влиянии на обучение кинестетических каналов восприятия информации, в процессе обучения также задействованы кинестетические тренажеры;
2. Проведен краткий обзор по парадигмам программирования, описаны особенности обучения логическому программированию и выделены основные сложности, возникающие при обучении логическим языкам программирования у студентов. Представлены и описаны основные подходы к обучению программированию;
3. Уточнено понятие логического стиля мышления, также построена его структурная модель, определено, что уровень сформированности логического стиля мышления влияет на успешность обучения логическому программированию. Для диагностики уровня сформированности логического стиля мышления разработаны тестовые задания и предложено использование методики построения ментальных карт. Проведена дифференциация студентов на основе проведенных диагностик по уровню сформированности их логического мышления и для каждого уровня предложены средства обучения, способствующие развитию логического стиля мышления и пониманию логического программирования в целом;
4. Разработаны ментальные карты и кинестетический тренажер по теме «Списки» языка логического программирования Prolog и использованы

нами в учебном процессе. Данные разработки являются неотъемлемой частью ментально-телесного подхода, поскольку воздействуют на кинестетические каналы восприятия, тем самым повышая уровень понимания материала, и наглядно отражают процесс нашего мышления, позволяя увидеть пробелы в понимании материала.

Перспективы данной работы мы видим в следующем:

– совершенствование системы диагностики уровня сформированности логического стиля мышления;

– доработка средств развития логического стиля мышления: метальных карт и кинестетических тренажеров;

– использование технологий 3D-печати для создания кинестетических тренажеров;

– Возможность применения наших разработок к другим языкам программирования.

Библиография

1. Алексеева Д.В., Степанова Т.А., Хайбрахманова А.Т. Кинестетические тренажеры как средства развития алгоритмического мышления // Материалы XIV Всероссийской (с международным участием) научно-практической конференции студентов, аспирантов и молодых ученых «Молодежь и наука» – Красноярск, 2014
2. Алюшин А. Л., Князева Е. Н. Телесный подход в когнитивной науке // Философские науки. 2009. № 2. С. 106-125.
3. Баженова И.В., Степанова Т. А. Использование методики ментальных карт при обучении программированию в высшей школе // Материалы II Всероссийской научно-практической конференции с международным участием «Перспективы и вызовы информационного общества», 2013.
4. Бархатова Д.А., Нигматулина Э.А., Степанова Т.А. Натурные средства обучения информатике в условиях реализации телесно-ментального подхода // Открытое образование. 2017. Т. 21. № 4. С. 73-83
5. Бархатова Д.А., Степанова Т.А., Нигматулина Э.А. Концепция телесно-ментального подхода в обучении // 2018
6. Бешенков С.А., Акимова И.В. «задачный» подход при обучении программированию // Международный журнал экспериментального образования. 2016. №9-2 С.178-180.
7. Газейкина А.И. Обучение программированию будущего учителя информатики // Педагогическое образование в России. 2012. №5. URL: https://cyberleninka.ru/article/n/obuchenie-programmirovaniyu-buduschego-uchitelya-informatiki (дата обращения: 30.05.2018).
8. Голубцова А.В.,Грук Е.Д., Степанова Т.А. Разработка ментальных алгоритмических карт по теме «Основные алгоритмические структуры» // Материалы II Всероссийской научно-практической конференции с международным участием «Перспективы и вызовы информационного

общества». – Красноярск: Красноярский государственный педагогический университет им. В.П. Астафьева, 2013. – с.180-183

9. Городняя Л.В. Парадигмы программирования в профессиональной подготовке информатиков // Проблемы специализированного образования. - Новосибирск, 1998.- с. 115-124 .

10. Жемчужников Д. Г. Разработка динамических игр как средства обучения программированию // Вестник Московского городского педагогического университета. Серия: Информатика и информатизация образования. 2010. № 2. С. 49–51.

11. Жужжалов В.Е. Основы интеграции парадигм программирования в курсе информатики // Российская академия образования. Институт содержания и методов обучения.- М.: Образование и информатика, 2004. – 127 с.

12. Зак А.З. Развитие и диагностика мышления подростков и старшеклассников / А.З. Зак. – М.; Обнинск: ИГ – СОЦИН, 2010. – 350 с.

13. Зеленин В.М., Пак Н.И., Якушевич В.И. Система искусственного интеллекта и язык Пролог в обучении: Учебное пособие и автоматизированный учебный курс для студентов педвузов, учителей и учащихся старших классов. Красноярск: РИО КГПУ, 1999., стр.45.

14. Калитина В.В., Пушкарева Т.П., Степанова Т. А. Алгоритмические ментальные карты как эффективное средство обучения программированию // Сборник статей Международной научно-практической конференции «Фундаментальные и прикладные научные исследования», Москва, 2015. С. 179.

15. Калитина В.В., Пушкарева Т.П., Степанова Т. А. Развитие алгоритмического стиля мышления при обучении программированию в вузе // «Теоретические и практические аспекты психологии и педагогики». Аэтерна, Уфа, 2015. С. 101.

16. Киргизова Е.В, Нигматулина Э.А. «Специфика обучения программированию будущих учителей информатики» // Научно-технический

журнал «Образовательные технологии», г.Воронеж, ВГПУ, №1, 2008. [с. 7-9](УДК 681.142.1.01).

17. Кухтина Е.С., Сорокин А.В., Бархатова Д.А. Развитие мышления учащихся с использованием натурных дидактических средств на уроках информатики // Актуальные проблемы авиации и космонавтики. 2015. №11 С.957-959.

18. Лебедева Т.П., Пак Н.И., Рукосуева Н.В. О сущности обучения с позиции информационно-ментального подхода. // Информация и образование: границы коммуникации. 2015. № 7(15). С. 31-33

19. Найсер У. Познание и реальность. М.: Прогресс, 1981.

20. Нигматулина Э.А. Методика телесно-полиязыкового обучения будущих учителей информатики программированию: магистерская диссертация , 2017

21. Нигматулина Э.А. Использование практико-ориентированных «живых» задач при обучении программированию/ Э. А. Нигматулина // Студенческая наука XXI: материалы XIIМеждунар. студенч. науч.-практ. конф. (Чебоксары, 25 янв. 2017 г.) / редкол.: О. Н. Широков [и др.]. – Чебоксары: ЦНС «Интерактив плюс», 2017. – № 1 (12). – ISSN 2413-3825.

22. Нигматулина Э.А. Интеграция парадигм программирования в вузовском курсе информатики / Открытое образование: опыт.проблемы, перспективы: материалы V Всероссийской научно-практической конференции с международным участием. Ерасноярск, 21-22 мая 2009 года / Красноярский государственный педагогический университет им. В.П., Астафьева. – Красноярск, 2009. – 332 с. [с. 43-46]

23. Нигматулина Э.А. Особенности формирования программистского стиля мышления будущих учителей информатики / Информатизация образования и методика электронного обучения: материалы I Международной научной конференции в рамках IV Международного научно-образовательного форума «Человек, семья и общество: история и

перспективы развития» (Красноярск, 27-30 сентября 2016 г.); под общ.ред. М.В. Носова. – Красноярск: Сиб. федер. ун-т, 2016. – 468 с. С.209-213.

24. Нигматулина Э.А., Сокольская М.А., Степанова Т.А. Расширение понятия алгоритмического мышления при изучении современных технологий программирования в педагогическом вузе (статья) // Материалы VIII Международной научно-практической конференции «Педагогический профессионализм в образовании». – Новосибирск, 2012. – с.152-158.

25. Нигматулина Э.А., Степанова Т.А Реализация параллельного изучения в педвузе языков программирования различных парадигм. //«Информатика и образование», 2014, № 4.

26. Нигматулина Э.А., Степанова Т.А. Телесный подход к обучению программированию // Перспективы и вызовы информационного общества: материалы IV Всероссийской научно-практической конференции с международным участием. Красноярск, 12 ноября 2015 г. [Электронное ресурс] / КГПУ им. В.П. Астафьева. – Красноярск, 2015. С. 131-136.

27. Орел Е.А. Диагностика особенностей мыслительной деятельности специалистов в области информационных технологий (программистов): Автореферетдис…псих.нак. – Москва – 2007. – 23 с.

28. Пак Н.И., Зеленин, Якушевич В.И. Программирование на Прологе

29. Петькина Е.Д., Марченко Л.С. Уточнение понятия «Логический стиль мышления» как необходимое условие успешного обучения языку Пролог будущих учителей информатики // Актуальные проблемы информатики и информационных технологий в образовании: материалы Всероссийской конференции с международным участием. Красноярск, 23 мая 2017 г. / отв. ред. П.С. Ломаско; ред. кол.; Электрон. дан. Краснояр. гос. пед.ун-т им. В.П. Астафьева. – Красноярск. - 2017. С.63-66

30. Петькина Е.Д., Мамаева А.А., Баран М.И. Ментальное обучение информатике / Материалы международной научно - практической конференции «Внедрение результатов инновационных разработок: проблемы и перспективы», Стерлитамак, 8 января 2018

31. Петькина Е.Д., Мамаева А.А., Марченко Л.С. Разработка кинестетических тренажеров по теме «классы» в с++, / Материалы международной научно - практической конференции «Роль и место информационных технологий в современной науке», Магнитогорск, 28 декабря 2017

32. Петькина Е.Д., Мамаева А.А., Марченко Л.С. Использование методики ментальных карт в учебном процессе / Материалы международной научно - практической конференции «Вопросы образования и науки», Тамбов, 30 ноября 2017

33. Петькина Е.Д., Марченко Л.С., Баран М.И. Ментальные карты как средство формирования информационной компетентности учащихся / Материалы международной научно - практической конференции «Проблемы и приоритеты развития науки в XXI веке», Смоленск, 30 декабря 2017

34. Программирование: В 2т. Т1: учебник для студ.учреждений высш.проф.образование / Э.А. Нигматулина, Н.И.Пак, М.А.Сокольская, Т.А.Степанова:под ред.Н.И.Пак-М.:издательский центр «Академия» 2013г

35. Пушкарева Т.П., Степанова Т.А., Калитина В.В., Нигматулина Э.А. Развитие алгоритмического мышления в вузе как основа успешности в профессиональной деятельности / Materials of the XII International scientific and practical conference, «Modern scientific potential – 2016». Volume 9. Pedagogical sciences. Sheffield. Science and education LTD – 88 стр. С. 9-12.

36. Пушкарева Т.П., Степанова Т.А., Калитина В.В., Нигматулина Э.А. Кинестетические тренажеры как эффективное средство развития алгоритмического стиля мышления / Материали за 12-а международнанаучна практична конференция, «Бъдещитеизследвания», - 2016. Том 5. Педагогически науки. София. «Бял ГРАЛ-БГ» ООД -112 стр. С.69-71

37. Пушкарева Т.П., Степанова Т.А., Калитина В.В. Дидактические средства развития алгоритмического стиля мышления студентов // Образование и наука. 2017. №9. URL:

https://cyberleninka.ru/article/n/didakticheskie-sredstva-razvitiya-algoritmicheskogo-stilya-myshleniya-studentov (дата обращения: 15.05.2018)

38. Равнова К.А. Диагностика уровня сформированности логического стиля мышления // Материалы IV Международного Научно-образовательного форума. «Человек, семья и общество: история и перспективы развития» //. Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2015

39. Степанов М.А. Опыт мышления тела: автореф. дис. … канд. философ.наук. СПб, 2011. с.6

40. Степанова Т.А. Теория алгоритмического мышления. Красноярск: РИО КГПУ, 2014

41. Степанова Т.А. Методическая система обучения курсу «Численные методы» в условиях информационно-коммуникационной предметной среды: Дис. ... канд. пед. наук. - Красноярск, 2003.

42. Степанова Т. А. Сущность алгоритмического мышления с позиций информационного подхода // Инновации в непрерывном образовании, 2012, №3. С. 95.

43. Степанова Т.А. Теория алгоритмического мышления: учебное пособие для магистрантов, учителей общеобразовательных учреждений, преподавателей вузов; Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2014. – 72 с.

44. Стоякова К.Л. Использование логической парадигмы программирования для обучения информатике студентов в инженерных вузах: Дис. ... канд. пед. наук. -. Москва, 2008. – 159 с.

45. Тхостов А.Ш. Психология телесности. — М.: Смысл, 2002. – 287 с.

46. Фёдоров Е.Н. Использование ментальных схем в обучении студентов основам искусственного интеллекта // Решетневские чтения. 2014. №18. URL: https://cyberleninka.ru/article/n/ispolzovanie-mentalnyh-shem-v-obuchenii-studentov-osnovam-iskusstvennogo-intellekta (дата обращения: 30.05.2018).

47. Халитова Зульфия Равильевна Развитие абстрактно-логического мышления будущих учителей информатики при обучении программированию на основе интеграции различных парадигм // Вестник ТГГПУ. 2012. №1 С.273-277.

48. Чепикова Е.Ю. Степанова Т.А. Разработка кинестетического тренажера по теме «динамические структуры данных» курса «Языки и методы программирования» // Актуальные проблемы информатики и информационных технологий в образовании: материалы Всероссийской научно-практической конференции с международным участием в рамках XVIII международного научно-практического форума студентов, аспирантов и молодых ученых «Молодежь и наука XXI века». Красноярск, 23 мая 2017 г. [Электронный ресурс] / ред. кол.; отв. ред. П.С. Ломаско. – Электрон. дан. / Краснояр. гос. пед. ун-т им. В.П. Астафьева. – Красноярск, 2017

49. Ц.Ин, Д.Соломон. Использование Турбо-Пролога. – М., 1993

50. Шапиро Э., Стерлинг Л. «Искусство программирования на языке Пролог». М: Мир 1990г.

51. Joseph D. Novak, Cornell University. The Theory Underlying Concept Maps and How To Construct Them. – http://cmap.coginst.uwf.edu/info/index.html

Приложения

Приложение 1. Материалы для диагностики уровня сформированности логического стиля мышления

1 вариант

1	А больше Б в 9 раз Б меньше В в 4 раза В А	2	А меньше Б в 10 раз Б больше В в 6 раз А В
3	А больше Б в 3 раза Б меньше В в 6 раз В А	4	А больше Б в 4 раз Б меньше В в 3 раза В А
5	А меньше Б в 3 раза Б больше В в 7 раз А В	6	А больше Б в 9 раз Б меньше В в 12 раз В А
7	А больше Б в 6 раз Б больше В в 7 раз А В	8	А меньше Б в 3 раза Б больше В в 5 раз В А
9	А меньше Б в 10 раз Б больше В в 3 раза В А		

2 вариант

1 А меньше Б в 3 раза
Б больше В в 4 раза
В А

2 А больше Б в 2 раза
Б меньше В в 5 раз
А В

3 А меньше Б в 5 раз
Б больше В в 6 раз
В А

4 А меньше Б в 5 раз
Б меньше В в 2 раза
А В

5 А больше Б в 4 раза
Б меньше В в 3 раза
В А

6 А меньше Б в 5 раз
Б больше В в 3 раза
А В

7 А больше Б в 4 раза
Б меньше В в 7 раз
В А

8 А больше Б в 3 раза
Б меньше В в 5 раз
А В

9 А меньше Б в 2 раза
Б больше В в 8 раз
А В

Ответы к тесту 1

1 вариант	2 вариант
1. В < А	1. В < А
2. А < В	2. А < В
3. В > А	3. В < А
4. В < А	4. А < В
5. А > В	5. В < А
6. В > А	6. А < В
7. А > В	7. В > А
8. В < А	8. А < В
9. В > А	9. А > В

Тест 2 на логические суждения

1 вариант

1. Некоторые улитки являются горами. Все горы любят кошек. Значит, все улитки любят кошек.

а) правильно

б) неправильно

2. Все крокодилы умеют летать. Все великаны являются крокодилами. Значит, все великаны могут летать.

а) правильно

б) неправильно

3. Некоторые головки капусты - паровозы. Некоторые паровозы играют на рояле. Значит, некоторые головки капусты играют на рояле.

а) правильно

б) неправильно

4. Две поляны никогда не похожи одна на другую. Сосны и ели выглядят совершенно одинаково. Значит, сосны и ели не являются двумя полянами.

а) правильно

б) неправильно

5. Каждый квадрат круглый. Все квадраты красные.

а) бывают квадраты с красными углами

б) бывают квадраты с круглыми углами

в) бывают круглые красные углы

г) углы и квадраты - круглые и красные

д) ни одно из вышеперечисленных

6. Хорошие начальники падают с неба. Плохие начальники могут петь.

а) Плохие начальники летят с неба вниз.

б) Хорошие начальники, которые умеют летать - могут петь.

в) некоторые плохие начальники не могут петь.

г) некоторые хорошие начальники -плохие, так как они умеют петь.

д) ни одно из вышеперечисленных

2 вариант

1. Никто из людей не может стать президентом, если у него красный нос. У всех людей нос красный. Значит, никто из людей не может стать президентом.

а) правильно

б) неправильно

2. Все вороны собирают картины. Некоторые собиратели картин сидят в птичьей клетке. Значит, некоторые вороны сидят в птичьей клетке.

а) правильно

б) неправильно

3. Только плохие люди обманывают или крадут. Вера - хорошая.

а) Вера не крадет

б) Вера обманывает и крадет

в) ни одно из вышеперечисленных

4. Все воробьи не умеют летать. У всех воробьев есть ноги.

а) без ног воробьи не могут летать

б) все воробьи, у которых есть ноги, не могут летать

в) воробьи не могут летать, потому что у них есть ноги

г) ни одно из вышеперечисленных

5. Некоторые люди - европейцы. Европейцы имеют три ноги.

а) люди с двумя ногами не являются европейцами

б) европейцы с двумя ногами иногда являются людьми

в) Люди имеют три ноги потому что они европейцы

г) ни одно из вышеперечисленных

6. Цветы - это зеленые звери. Цветы пьют кофе.

а) все зеленые звери пьют кофе

б) все зеленые звери являются цветами

в) некоторые зеленые звери пьют кофе

г) Зеленые звери не пьют кофе

д) зеленые звери не являются цветами

е) ни одно из вышеперечисленных

Ответы к тесту 2

1 вариант	2 вариант
1. б	1. а
2. а	2. б
3. б	3. а
4. а	4. б
5. д	5. а
6. д	6. в

Тест 3 на математическую логику

1 вариант

1) $A \vee B = B \vee A$
   а) Да
   б) Нет

2) Выбрать равенство двойственное данному: A(AUB) = A
   а) $A(\overline{A}\,UB) = AB$
   б) AUAB = A
   в) A(AUB) = A
   г) $ABUA\overline{B} = A$

3) Выбрать операцию алгебры логики, задаваемую таблицей истинности:

а	в	с
1	1	1
1	0	0
0	1	0
0	0	0

$а) c = a \vee в \quad б) c = a \Leftrightarrow в \quad в) c = a \wedge в \quad г) c = a \Rightarrow в$

4) Найти: $|AUB|$ *если* $|A| = 16 \quad |B| = 8 \quad |AB| = 5$
   а) 14
   б) 22
   в) 19
   г) 18

5) Обозначим через a высказывание «Летом я поеду в деревню», а через b- «Летом я поеду в туристическую поездку».. Тогда высказывание c - ««Летом я поеду в деревню или в туристическую поездку» запишем так
   $а) c = a \vee в \quad б) c = a \Leftrightarrow в \quad в) c = a \wedge в \quad г) c = a \Rightarrow в$

2 вариант

1) $A \wedge B = B \wedge A$

   а) Да

   б) Нет

2) Выбрать равенство двойственное данному равенству: AUAB = A

   а) (UB) = AB

   б) AUAB = A

   в) A(AUB) = A

   г) ABUA = A

3) Какая формула тождественна $x \leftrightarrow y$

   а) $\overline{x} \wedge \overline{y}$

   б) $\overline{x} \vee \overline{y}$

   в) $\overline{x} \vee y$;

   г) $(x \to y) \wedge (y \to x)$

4) Выбрать операцию алгебры логики, задаваемую таблицей истинности:

а	в	с
1	1	1
1	0	0
0	1	1
0	0	1

$а) c = a \vee в$ $б) c = a \Leftrightarrow в$ $в) c = a \wedge в$ $г) c = a \Rightarrow в$

5) Обозначим через a высказывание «пришла весна»; а через b- «грачи прилетели». Тогда высказывание c - «пришла весна, и грачи прилетели» запишем так

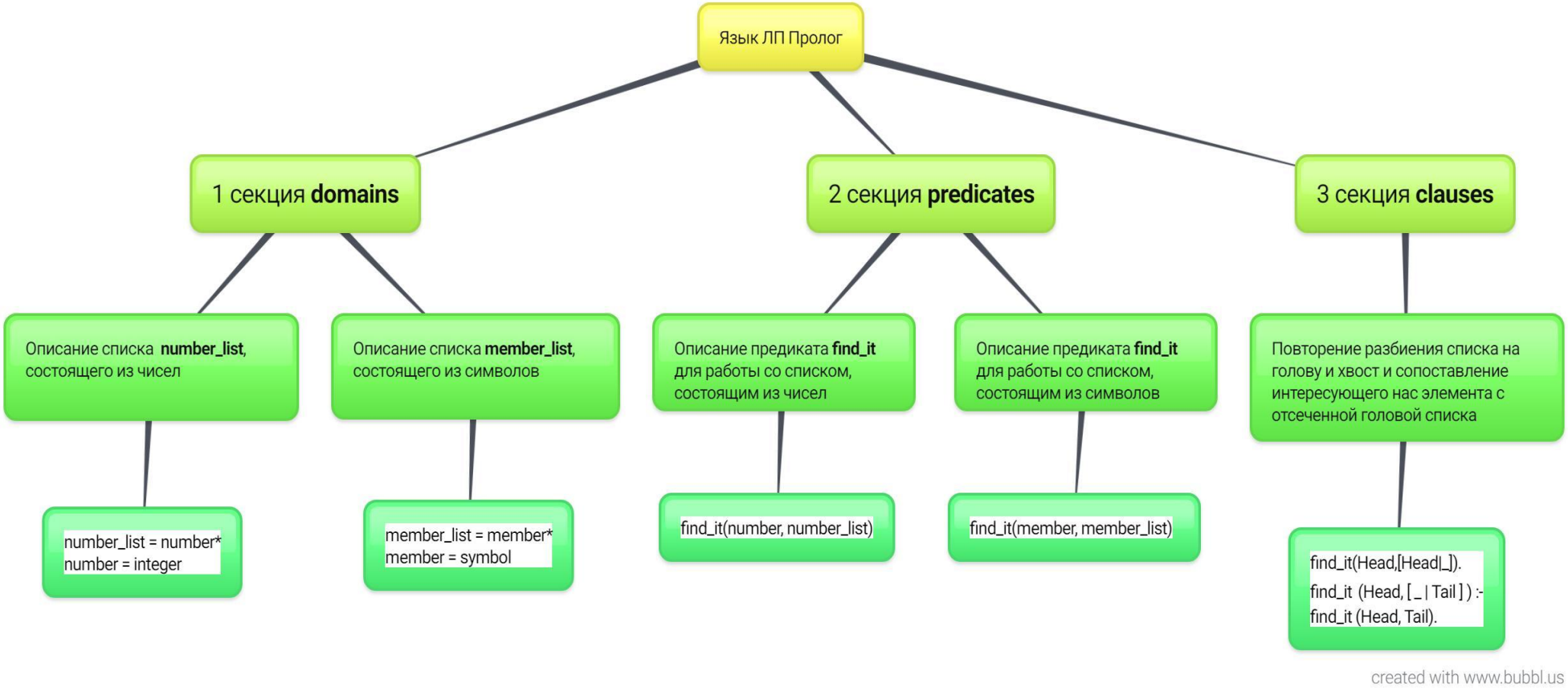
$а) c = a \vee в$ $б) c = a \Leftrightarrow в$ $в) c = a \wedge в$ $г) c = a \Rightarrow в$

Ответы к тесту 3

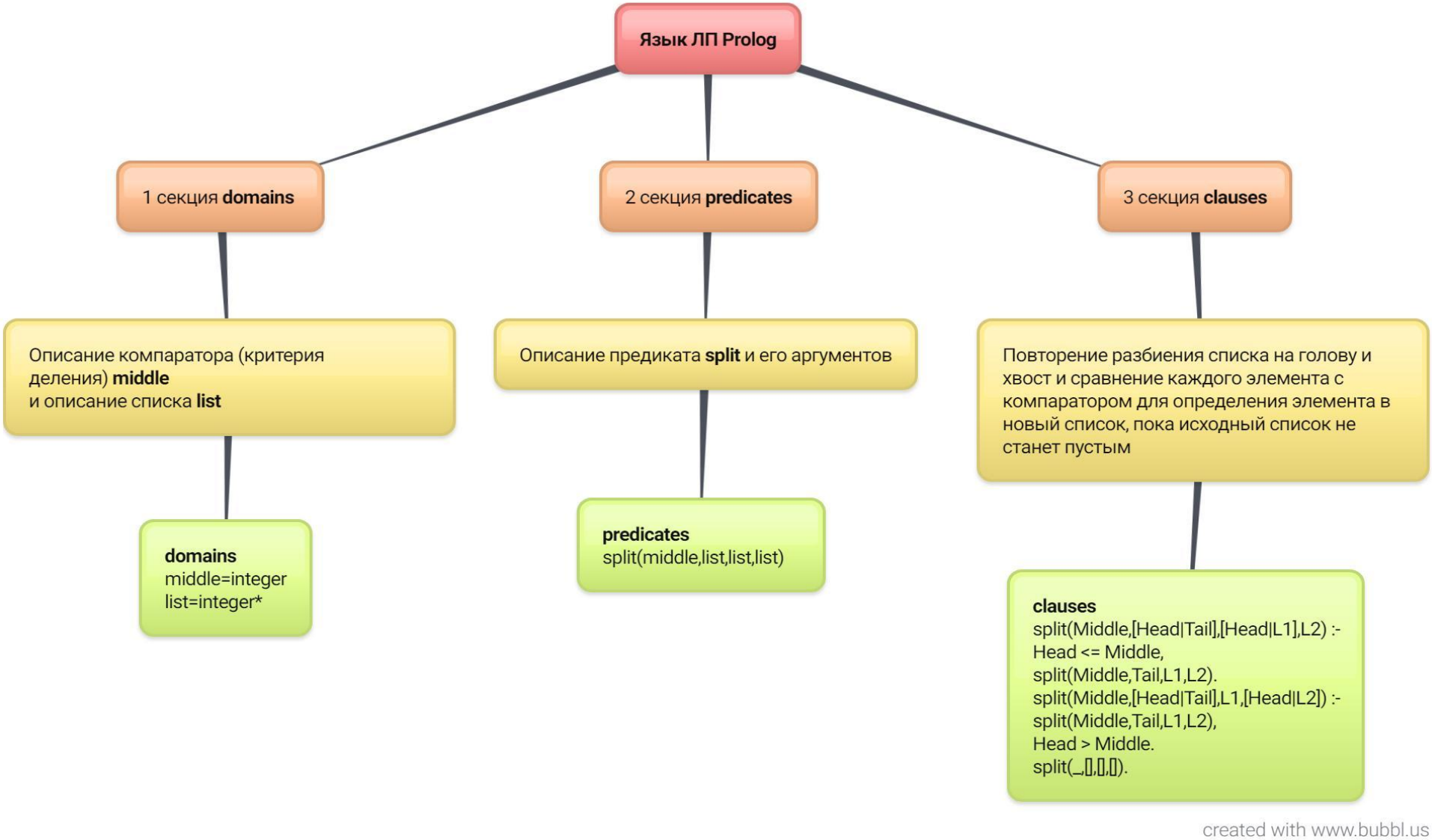
1 вариант	2 вариант
1. а	1. а
2. в	2. б
3. в	3. г
4. б	4. б
5. в	5. а

Приложение 2. Ментальные карты по теме "Язык логического программирования Prolog"

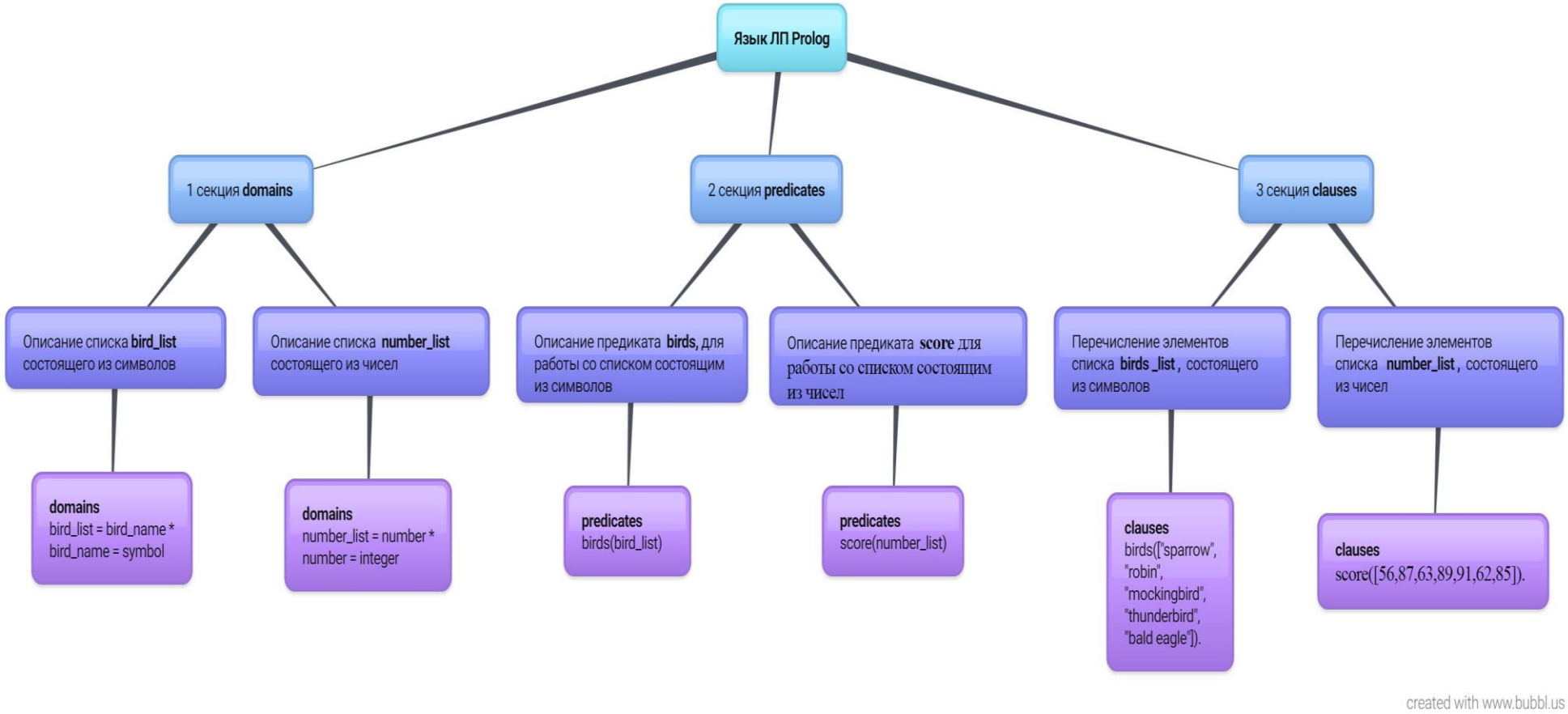
Ментальная карта по теме «Поиск элемента в списке»



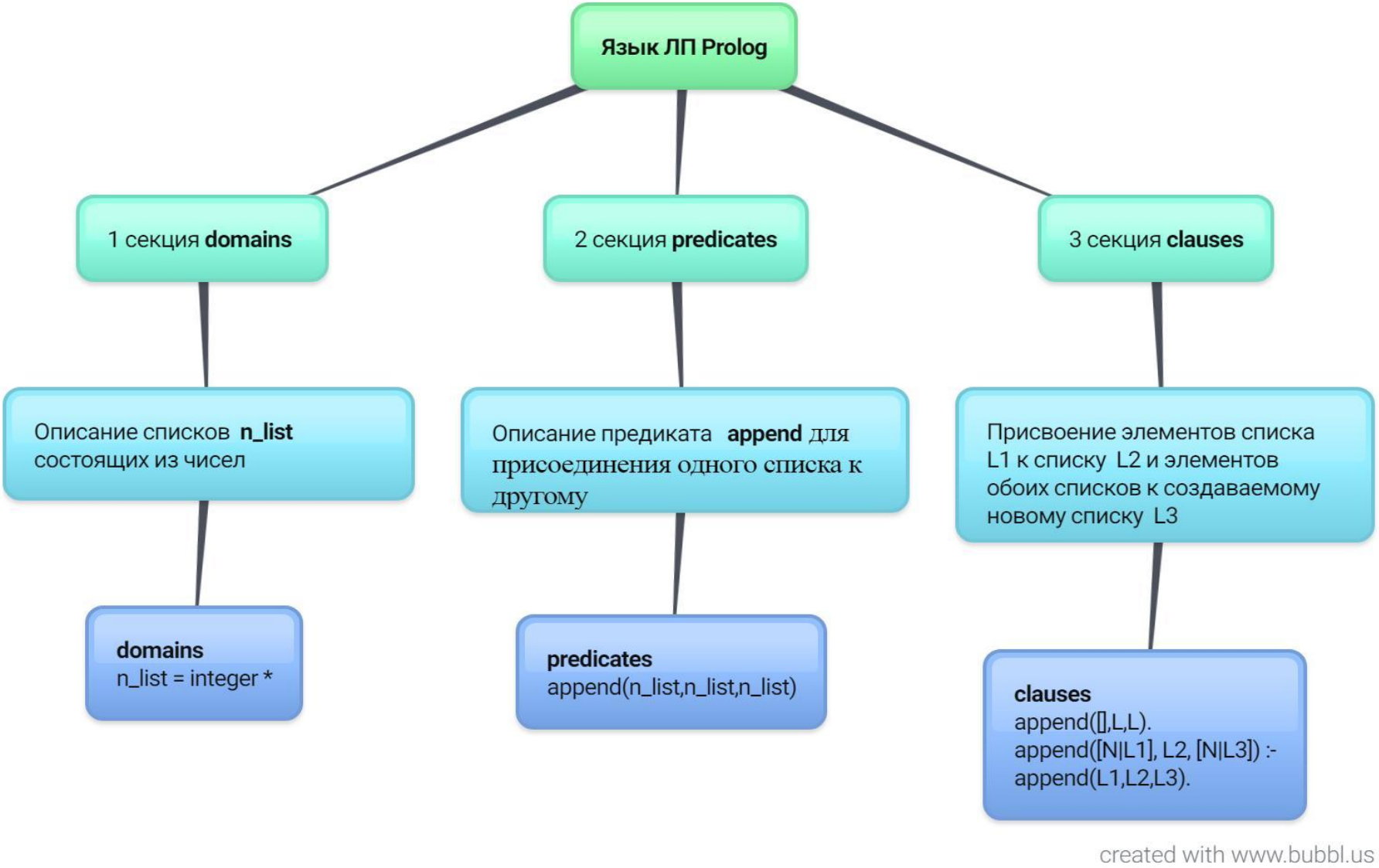
Ментальная карта по теме «Деление списка»



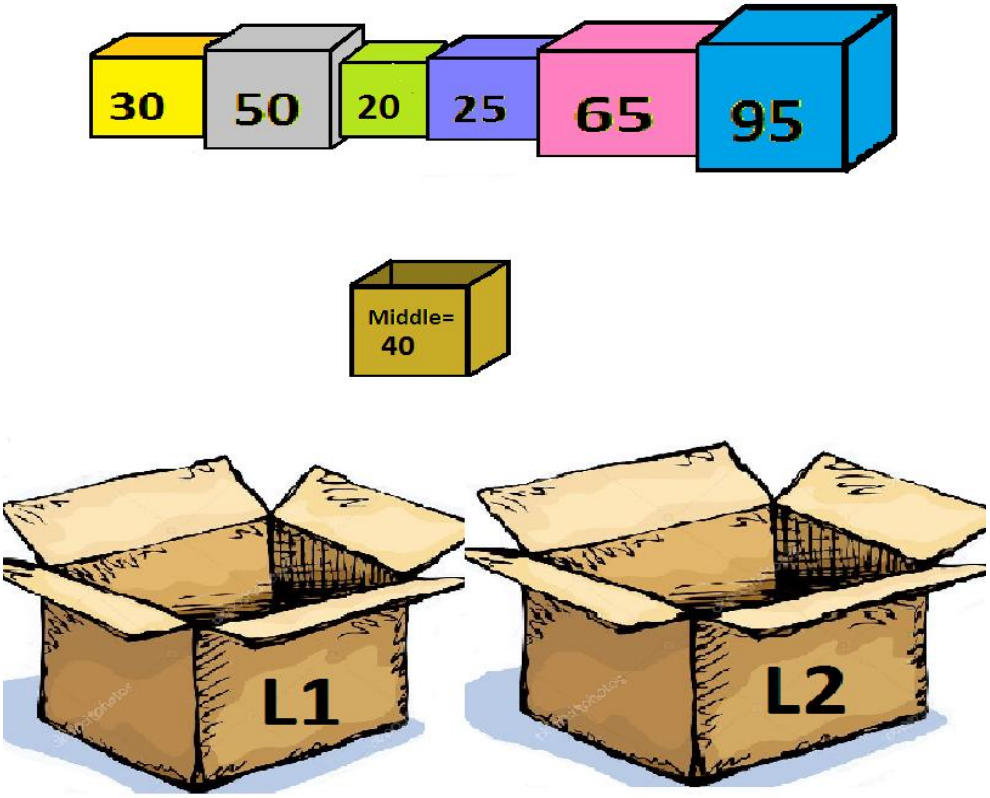
Ментальная карта по теме «Работа со списком



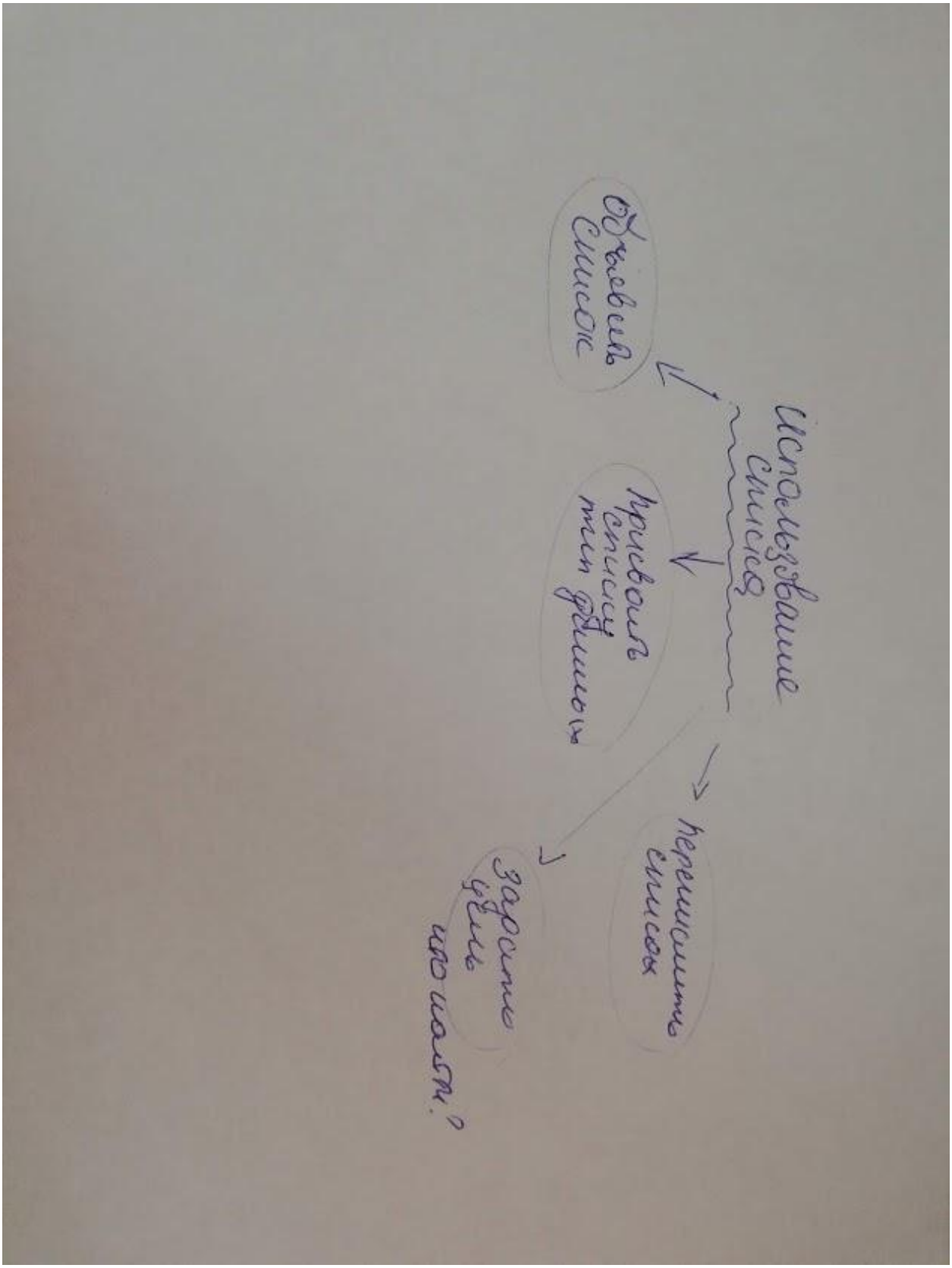
Ментальная карта по теме «Слияние списков»

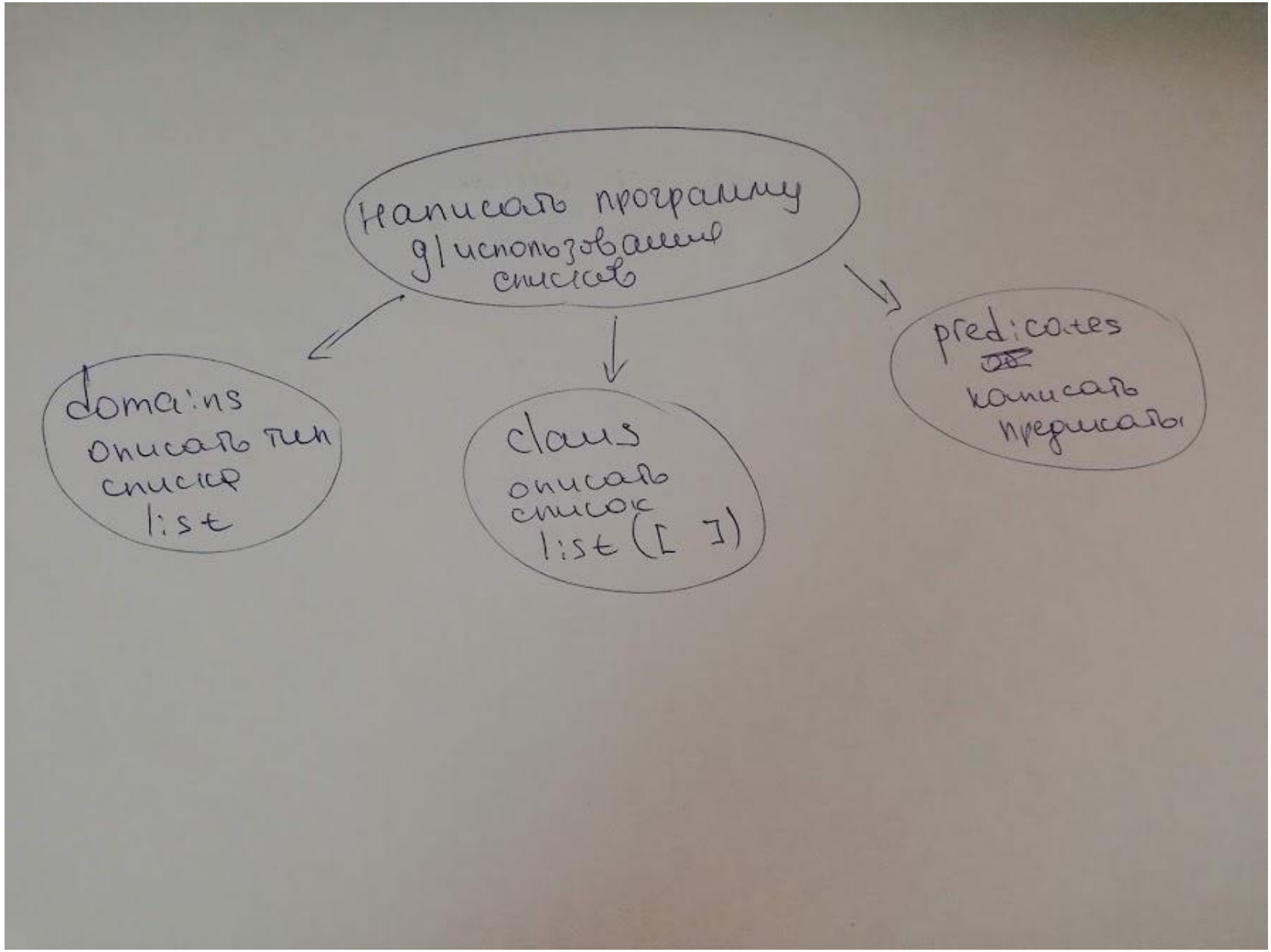


Приложение 3. Кинестетический тренажер по теме "Списки в Prolog"



Приложение 4. Ментальные карты студентов




Написать программу
с/использованием
списков
domains
описать тип
список
list
clauses
описать
список
list ([ ])
predicates
~~ОП~~
написать
предикаты