

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:

Зав. кафедрой ПОАС

_____ Ю.А. Орлова

«__» _____ 2023 г.

Логическая модель составления выражений доступа к элементам данных

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

ВКРБ–09.03.04–10.19–07–23–91

Листов 24

Руководитель работы

Сычев О.А. _____

«__» _____ 2023 г.

Нормоконтролер

_____ Кузнецова А.С.

«__» _____ 2023 г.

Исполнитель

студент группы ПрИн-467

«__» _____ 2023 г.

Волгоград, 2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

УТВЕРЖДАЮ:

Зав. кафедрой ПОАС

_____ Ю.А. Орлова

«__» _____ 2023 г.

Логическая модель составления выражений доступа к элементам данных

ЛИСТ УТВЕРЖДЕНИЯ

А.В.00001-01 91 01-1-ЛУ

Листов 1

Руководитель работы

Сычев О.А. _____

«__» _____ 2023 г.

Нормоконтролер

_____ Кузнецова А.С.

«__» _____ 2023 г.

Исполнитель

студент группы ПрИн-467

«__» _____ 2023 г.

Волгоград, 2023 г.

Аннотация

Документ представляет собой техническое задание к выпускной работе бакалавра на тему «Логическая модель составления выражений доступа к элементам данных». В документе изложены основания и назначения разработки программы, требования к разрабатываемой программе, технико-экономические показатели. Документ включает в себя страниц – 23, приложений – 6, рисунков – 6.

Ключевые слова: доступ к элементам данных, деревья решений, тренажер.

Содержание

Введение	6
1.1 Наименование программы	6
1.2 Область применения	6
2 Основание для разработки	6
3 Назначение разработки	6
4 Требования к программе	7
4.1 Требования к функциональным характеристикам	7
4.2 Требования к надежности	8
4.2.1 Требования к обеспечению надежного функционирования программы	8
4.2.2 Время восстановления после отказа	9
4.2.3 Отказы из-за некорректных действий пользователя	9
4.3 Требования к условиям эксплуатации	10
4.3.1 Климатические условия эксплуатации	10
4.3.2 Требования к квалификации и численности персонала	10
4.4 Требования к составу и параметрам технических средств	10
4.5 Требования к информационной и программной совместимости	11
4.5.1 Требования к информационным структурам и методам решения	11
4.5.2 Требования к исходным кодам и языкам программирования	11
4.5.3 Требования к программным средствам, используемым программой	11
4.5.4 Требования к защите информации	12
4.6 Требования к маркировке и упаковке	12
4.7 Требования к транспортированию и хранению	12
4.8 Специальные требования	12
5 Требования к программной документации	12
6 Технико-экономические показатели	13
6.1 Экономические преимущества разработки	13
7 Стадии и этапы разработки	13

7.1 Стадии разработки	13
7.2 Содержание работ по этапам	13
8 Порядок контроля и приемки	14
8.1 Виды испытаний	14
Приложение А.1 – Описание процесса составления Выражений Доступа к Необходимому Элементу Данных	16
Приложение А.2 – Виды обнаруживаемых ошибок	18
Приложение Б.1 – Диаграммы вариантов использования	22
Приложение Б.2 – Сценарии вариантов использования	23
Приложение Б.3 – Макеты экранных форм	24
Приложение Б.4 – Структура и формат данных	26

Введение

1.1 Наименование программы

Element Access Expressions Trainer.

1.2 Область применения

Приложение, далее Тренажер, рассчитано на обучение студентов начальных курсов по программированию умению составлять выражения доступа к элементам данных.

2 Основание для разработки

Тренажер разрабатывается на основании приказа №1235-ст от 6 сентября 2022 года «Об утверждении тем и руководителей выпускных работ бакалавров», утвержденного кафедрой ПОАС по теме «Логическая модель составления выражений доступа к элементам данных».

3 Назначение разработки

Усовершенствование технологий интеллектуального обучения в сфере программирования в разделе развития умения составления выражений доступа к

элементам данных. Данная программа предназначена для эксплуатации студентами начальных курсов по программированию.

4 Требования к программе

4.1 Требования к функциональным характеристикам

Тренажер должен:

- обеспечивать пошаговое составление Выражения Доступа к Необходимому Элементу Данных (ВДНЭД) студентом. Этапы составления ВДНЭД перечислены в приложении А.1 – Описание процесса составления Выражений Доступа к Необходимому Элементу Данных;
- обеспечивать проверку выбираемого ответа студента на каждом этапе составления ВДНЭД. Этапы составления ВДНЭД перечислены в приложении А.1 – Описание процесса составления Выражений Доступа к Необходимому Элементу Данных. Виды обнаруживаемых ошибок представлен в Приложении А.2 – Виды обнаруживаемых ошибок;
- выводить осмысленный текст сообщений ошибок неверного выбора студентом элемента начала ВДНЭД. Виды обнаруживаемых ошибок представлен в Приложении А.2 – Виды обнаруживаемых ошибок;
- выводить осмысленный текст сообщений ошибок неверного выбора студентом элемента продолжения ВДНЭД. Виды обнаруживаемых ошибок представлен в Приложении А.2 – Виды обнаруживаемых ошибок;
- выводить осмысленный текст сообщений ошибок неверного окончания ВДНЭД. Виды обнаруживаемых ошибок представлен в Приложении А.2 – Виды обнаруживаемых ошибок;
- выводить подсказки по начальному элементу ВДНЭД при запросе студента;

- выводить советы по следующему элементу ВДНЭД при запросе студента;
- обеспечивать проверку ответов студента согласно моделям, разработанным в виде деревьев решений в виде логики первого порядка, описываемым в Приложении А.1 – Спецификации разрабатываемых деревьев решений;
- вычислять следующий ожидаемый ответ студента согласно моделям, разработанным в виде деревьев решений в виде логики первого порядка, описываемым в Приложении А.1 – Спецификации разрабатываемых деревьев решений;
- обеспечивать возможность интеграции с кодом генерации наводящих вопросов и их показа.

4.2 Требования к надежности

4.2.1 Требования к обеспечению надежного функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением пользователем совокупности организационно-технических мероприятий, перечень которых приведен ниже:

- организацией бесперебойного питания технических средств;
- использованием лицензионного программного обеспечения;
- регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;

– регулярным выполнением требований ГОСТ 51188-98 «Защита информации. Испытания программных средств на наличие компьютерных вирусов».

4.2.2 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать времени, необходимого на перезагрузку операционной системы и запуск программы, при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

4.2.3 Отказы из-за некорректных действий пользователя

Отказы программы возможны вследствие некорректных действий оператора (пользователя) при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу конечного пользователя без предоставления ему административных привилегий.

4.3 Требования к условиям эксплуатации

4.3.1 Климатические условия эксплуатации

Климатические условия эксплуатации, при которых программа должна функционировать, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

4.3.2 Требования к квалификации и численности персонала

Для использования программного средства требуется один человек, являющийся пользователем ПК.

4.4 Требования к составу и параметрам технических средств

В состав технических средств должен входить персональный компьютер, включающий в себя:

- процессор Intel Core i5 4th-gen и лучше;
- оперативную память объемом 4 ГБ и более;
- операционную систему Windows 8.1 и выше.

В состав периферийных устройств должны входить:

- монитор;
- клавиатура;
- мышь.

4.5 Требования к информационной и программной совместимости

4.5.1 Требования к информационным структурам и методам решения

Требования к информационным структурам и методам решения не предъявляются.

4.5.2 Требования к исходным кодам и языкам программирования

Требования к исходным кодам и языкам программирования не предъявляются.

4.5.3 Требования к программным средствам, используемым программой

Программные средства, используемые программой, должны быть представлены:

- лицензионной локализованной версией операционной системы Windows 8.1 и выше;
- виртуальной машиной JAVA с JDK 14 и выше;
- фреймворком Apache Jena версии 4.6 и выше.

4.5.4 Требования к защите информации

Требования к защите информации не предъявляются.

4.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

4.7 Требования к транспортированию и хранению

Требования к транспортированию и хранению не предъявляются.

4.8 Специальные требования

Специальные требования не предъявляются.

5 Требования к программной документации

Состав разрабатываемой программной документации должен включать в себя:

– техническое задание, оформленное в соответствии с ГОСТ 19;

- пояснительную записку, оформленную в соответствии с СТП 24;
- руководство системного программиста, оформленное в соответствии с ГОСТ 19.

6 Технико-экономические показатели

6.1 Экономические преимущества разработки

Ориентировочная экономическая эффективность не рассчитывается.

7 Стадии и этапы разработки

7.1 Стадии разработки

Разработка приложения должна быть проведена в 5 стадий:

- анализ и сбор информации по предметной области;
- проектирование приложения;
- разработка приложения;
- тестирование приложения;
- документирование приложения.

7.2 Содержание работ по этапам

Анализ и сбор информации по предметной области:

- поиск и чтение статей по предметной области;
- поиск конкурентов по предметной области.

Проектирование приложения:

- разработка диаграммы и сценариев использования;
- проектирование макета интерфейса.

Разработка приложения:

- разработка модели решения:
 1. разработка логических моделей в виде деревьев решений.
 2. разработка модели данных.
- разработка клиента приложения;
- разработка сервера приложения.

Тестирование приложения:

- тестирование приложения на заранее подготовленных входных данных;
- тестирование приложения с участием студентов ВолГТУ, обучающихся на направлении «Программная инженерия».

Документирование приложения:

- разработка ПЗ приложения.

8 Порядок контроля и приемки

8.1 Виды испытаний

Программа сдаётся на проверку заказчику 05.06.2023. При обнаружении в программе ошибок или недостатков исполнитель обязуется устранить их в недельный срок и предоставить программу на повторную проверку.

Программа сдаётся на проверку независимому тестировщику не позднее

09.06.2023. Результаты тестирования предоставляются на защите дипломного проекта членам ГАК.

Описание процесса составления Выражений Доступа к Необходимому Элементу Данных

Выражение доступа к необходимому элементу данных- IValue выражение, служащее для получения доступа к определенному элементу данных, находящемуся в памяти выполняемой программы.

Данный тренажер разрабатывается для тренировки навыков на языке C++ и учитывает следующие типы объектов языка:

- классы;
- объекты классов;
- указатели;
- массивы;
- динамические массивы;
- указатели на члены классов (pointer-to-member);
- контейнеры;
- ссылки;
- STL итераторы;
- функции;
- пространства имен.

Тренажер рассчитывается на отработку навыков со следующими операторами:

- получение поля или метода объекта по значению (.);
- получение поля или метода объекта по указателю (->);
- получение статического члена класса члена пространства имен (::);
- получение элемента данных по указателю (*);
- взятие элемента контейнера или массива по ключу или индексу ([]);

- взятие поля или метода объекта посредством указателя на член класса по значению (.*);
- взятие поля или метода объекта посредством указателя на член класса по указателю (->*).

В Тренажере приняты следующие упрощения:

- все функции и методы, используемые в задаче, имеют фиксированные наборы значений параметров, перечисленных либо в условии, либо в предлагаемых вариантах ответа;
- индексы или ключи массивов или контейнеров не вычисляются Тренажером. Они задаются жесткой формулой, упомянутой либо в условии задачи, либо в предлагаемых вариантах ответа.

Тренажер разбивает решение задачи на следующие этапы:

- выбор начального элемента выражения;
- последовательный выбор промежуточных элементов выражения;
- завершение ввода выражения.

Второго этапа может не быть если задача предполагает выражение из одного элемента. Второй этап повторяется до тех пор, пока целевой элемент не будет достигнут.

Третий этап инициируется нажатием обучающегося по кнопке «Завершить ввод выражения».

Макет приложения и соответствующие кнопки изображены в Приложении Б.3 – Макеты экранных форм.

Если в процессе ввода выражения обучаемый совершает ошибку, тренажер сообщает о ней в текстовой форме под областью ввода ответа.

Виды обнаруживаемых ошибок

Данный тренажер рассчитан на обнаружение и объяснение следующих видов ошибок:

- на этапе выбора стартового элемента:
 - 1) выбранного элемента не существует в памяти;
 - 2) выбранный элемент существует, может быть стартовым элементом, но не может привести к необходимому элементу данных;
 - 3) выбранный элемент не может быть стартовым т.к. есть только поля с именем выбранного элемента;
 - 4) выбранный элемент не может быть стартовым т.к. он содержится во вложенном пространстве имен или является вложенным классом;
 - 5) выбранный элемент не может быть стартовым т.к. он содержится внутри контейнера или массива.
- на этапе выбора промежуточного элемента выражения:
 - 1) выбранный оператор невозможно использовать с элементом данных, к которому ведет текущее выражение, введенное обучаемым;
 - 2) выбранный тип операнда несовместим с выбранным оператором;
 - 3) выбранный элемент данных не может привести к необходимому элементу данных;
 - 4) элемента данных с выбранным именем не существует в памяти программы;
 - 5) пропущен промежуточный элемент выражения;
 - 6) попытка продолжить выражение после достижения

необходимого элемента данных;

- 7) выбранного элемента не существует в пространстве имен, к которому ведет текущее выражение, введенное обучаемым, элемент с выбранным именем, способный привести к необходимому элементу данных, существует в другом пространстве имен;
- 8) выбранного элемента не существует в пространстве имен, к которому ведет текущее выражение, введенное обучаемым, существует статическое поле класса с выбранным именем, способное привести к необходимому элементу данных;
- 9) выбранного элемента не существует в пространстве имен, к которому ведет текущее выражение, введенное обучаемым, существует поле класса с выбранным именем, способное привести к необходимому элементу данных;
- 10) выбранного статического поля не существует в классе, к которому ведет текущее выражение, введенное обучаемым, есть поле класса с выбранным именем;
- 11) выбранного статического поля класса не существует в классе, к которому ведет текущее выражение, введенное обучаемым, существует статическое поле класса с выбранным именем, способное привести к необходимому элементу данных, в другом классе;
- 12) выбранного статического поля класса не существует в классе, к которому ведет текущее выражение, введенное обучаемым, существует член пространства имен с выбранным именем, способный привести к необходимому элементу данных;
- 13) использование указателя на член класса с классом, а не с объектом класса;
- 14) выбранного поля объекта класса, к которому ведет текущее выражение, введенное обучаемым, не существует в классе, которому принадлежит объект, существует поле объекта с выбранным именем,

способное привести к необходимому элементу данных, в объекте другого класса;

15) выбранного поля объекта класса, к которому ведет текущее выражение, введенное обучаемым, не существует в классе, которому принадлежит объект, существует статическое поле класса, которому принадлежит объект, с выбранным именем;

16) выбранного поля объекта класса, к которому ведет текущее выражение, введенное обучаемым, не существует в классе, которому принадлежит объект, существует член пространства имен с выбранным именем, способный привести к необходимому элементу данных;

17) выбранный указатель на член класса указывает на поле, не способное привести к необходимому элементу данных;

18) выбранный указатель на член класса указывает на поле класса, отличного от того, к которому принадлежит объект класса, к которому ведет текущее выражение, введенное обучаемым;

19) элемента данных с выбранным ключом или индексом не существует в массиве или контейнере, к которому ведет текущее выражение, введенное обучаемым;

20) элемента данных с выбранным ключом или индексом не существует в массиве или контейнере, к которому ведет текущее выражение, введенное обучаемым, элемент данных с выбранным ключом или индексом содержится во вложенном контейнере или массиве;

21) элемента данных с выбранным ключом или индексом не существует в массиве или контейнере, к которому ведет текущее выражение, введенное обучаемым, элемент данных с выбранным ключом или индексом, способный привести к необходимому элементу данных, содержится в другом контейнере или массиве;

22) получение запоследнего или предпервого элемента

контейнера посредством итератора.

– на завершения ввода выражения:

- 1) не взято поле объекта;
- 2) на взят элемент массива;
- 3) не взято статическое поле объекта;
- 4) не взят член пространства имен;
- 5) получен указатель на необходимый элемент данных.

Диаграммы вариантов использования

На рисунке Б.1.1 представлена диаграмма вариантов использования.

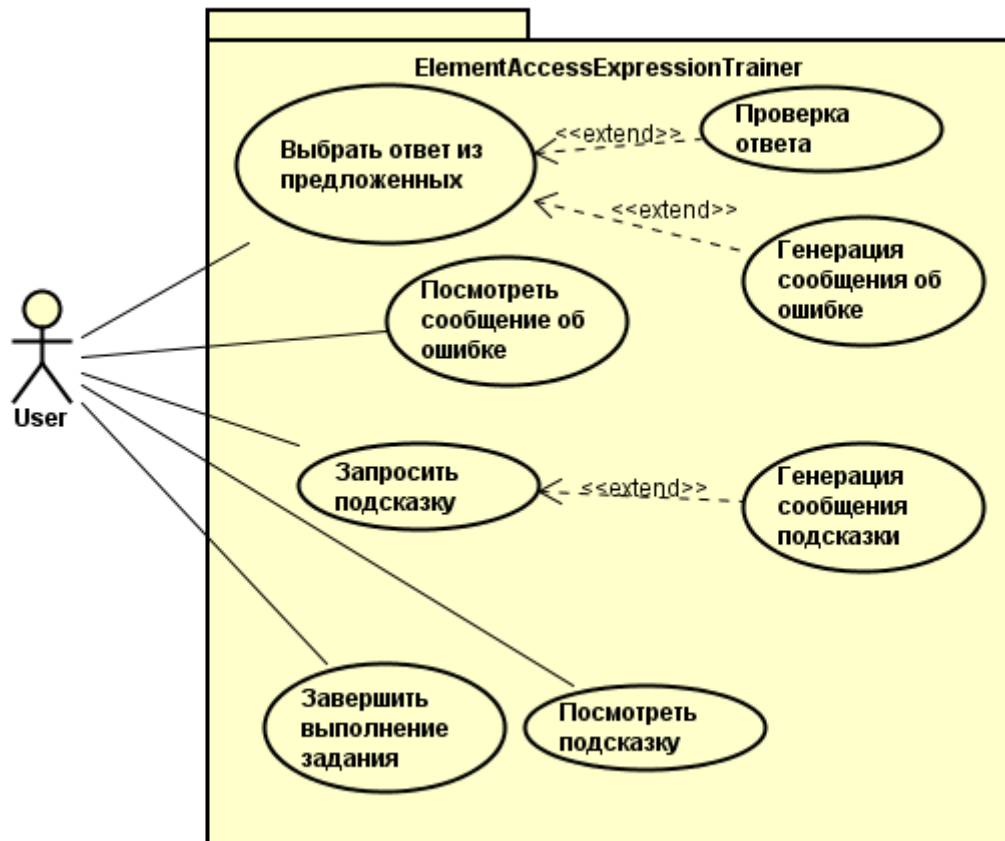


Рисунок Б.1.1 – Диаграмма вариантов использования программы

Сценарии вариантов использования

Пользователь выбирает вариант из предложенных:

- Пользователь нажимает выбранный вариант ответа из предложенных на зону ответа
- В текстовом поле под зоной ответа показывается результат (верно / ошибка)

Пользователь запрашивает подсказку:

- Во время выполнения задания пользователь нажимает ЛКМ по кнопке «Подсказка»
- В текстовом поле под зоной ответа показывается подсказка, наводящая на следующий элемент выражения

Пользователь завершает выполнение задания с ошибкой:

- Пользователь нажимает ЛКМ по кнопке «Завершить выполнение»
- Кнопка «Завершить выполнение» блокируется до следующего выбора ответа пользователя
- В текстовом поле под зоной ответа показывается сообщение об ошибке

Пользователь завершает выполнение задания с верным ответом:

- Пользователь нажимает ЛКМ по кнопке «Завершить выполнение»
- На экране всплывает сообщение о верном решении задачи

Макеты экранных форм

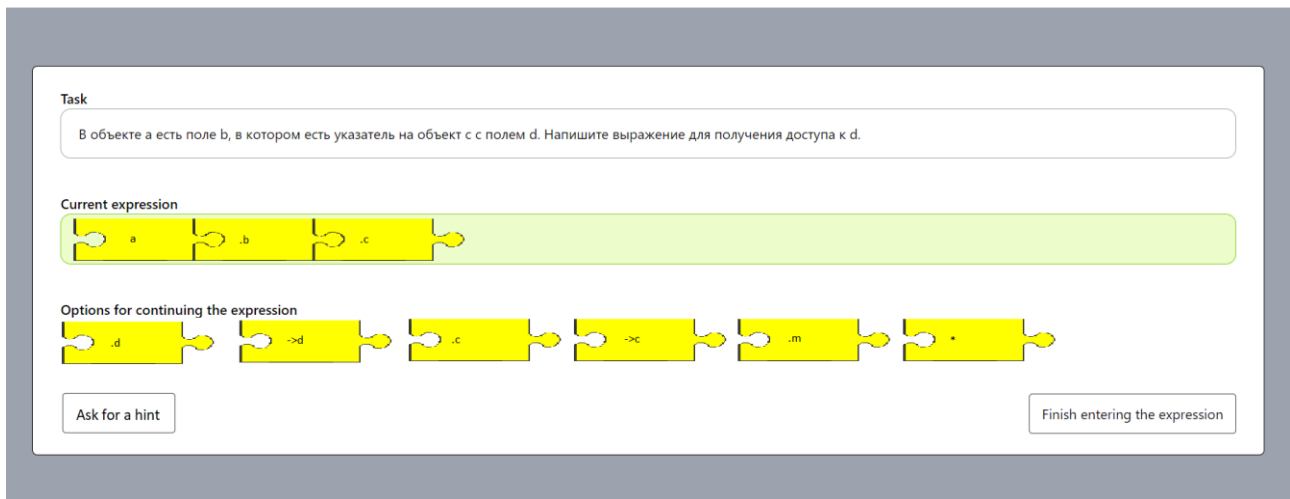


Рисунок Б.3.1 – Макет Тренажера при выборе верного варианта продолжения выражения



Рисунок Б.3.2 – Макет Тренажера при выборе неверного варианта продолжения выражения

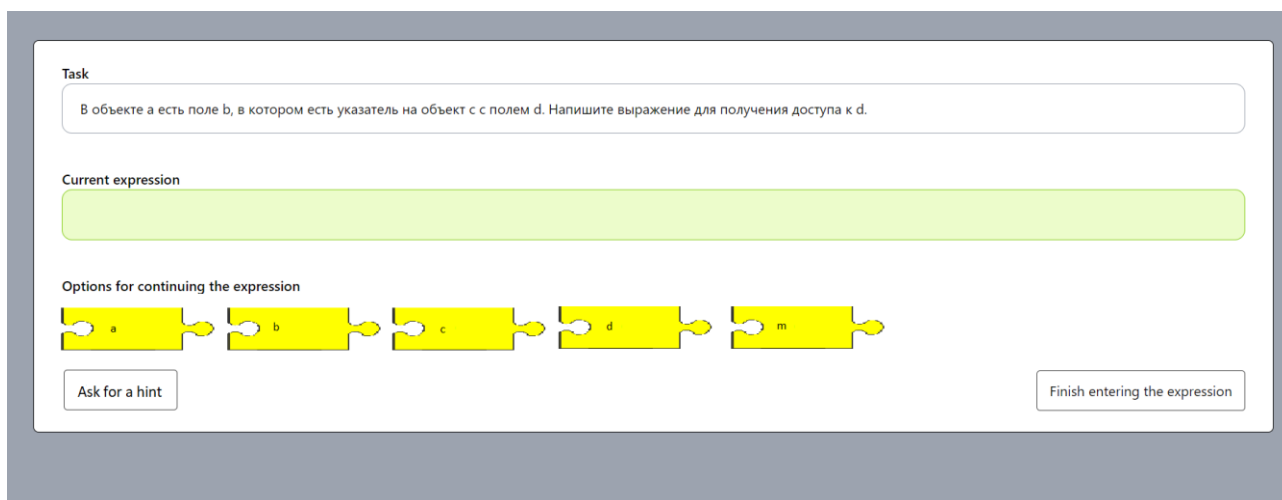


Рисунок Б.3.3 – Макет Тренажера при выборе начального элемента выражения

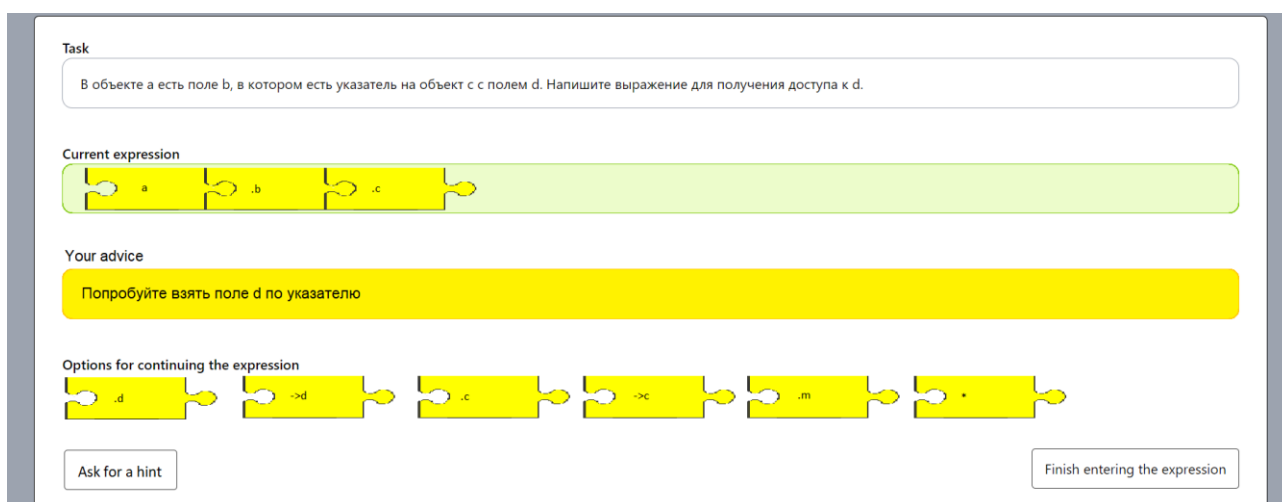


Рисунок Б.3.4 – Макет Тренажера при запросе подсказки

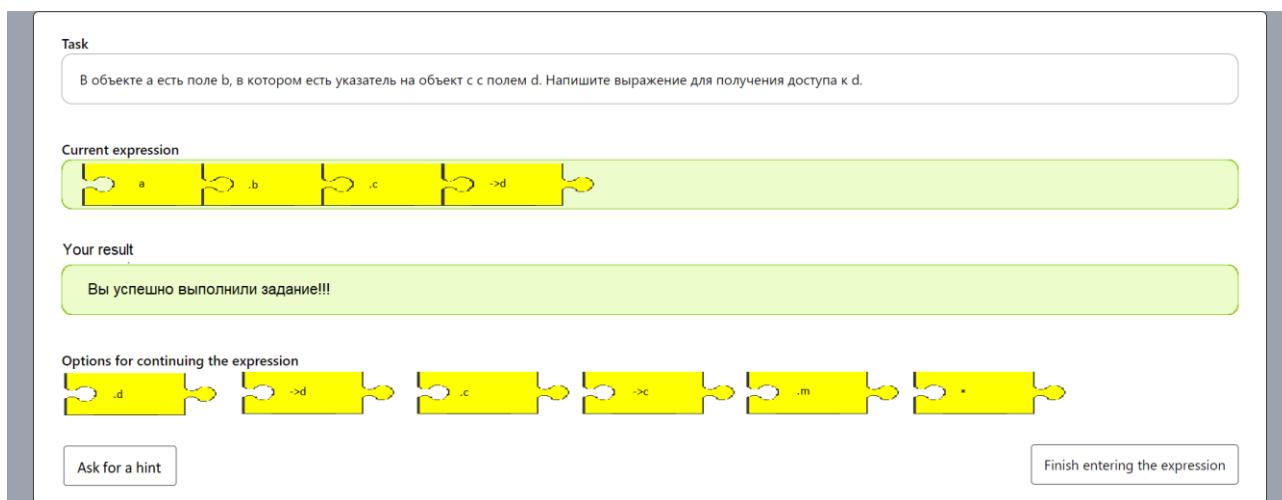


Рисунок Б.3.5 – Макет Тренажера при завершении задания

Структура и формат данных

Входные данные:

- структура объектов в памяти в виде Resource Description Framework (RDF) графа;
- текст задания.