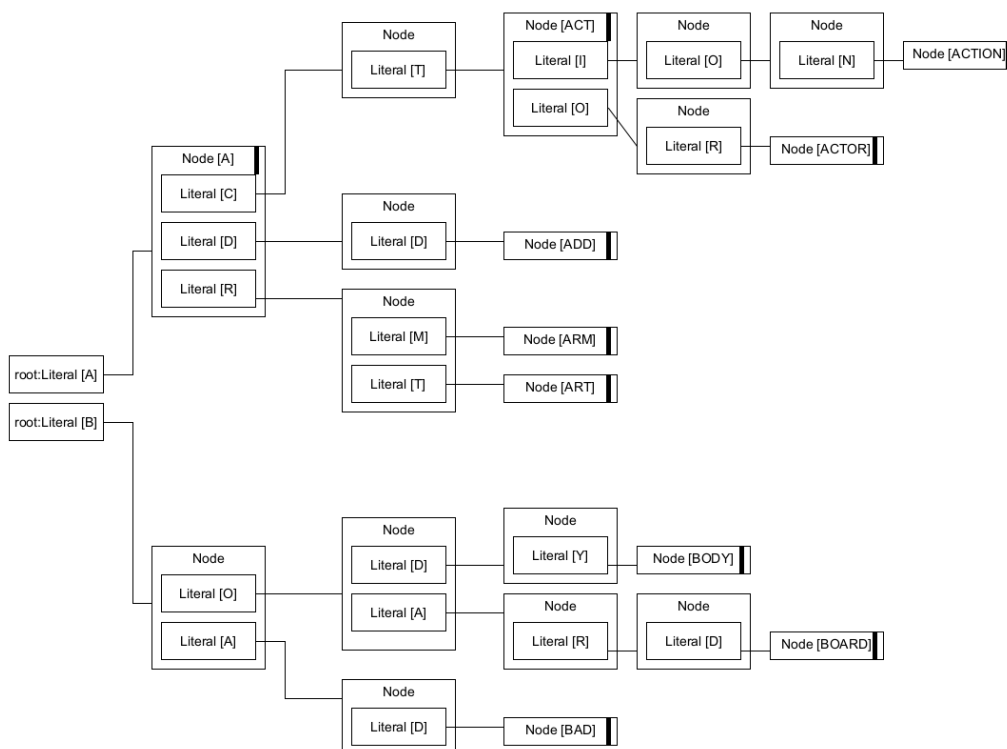


DDC23 Техническое задание

Контейнер данных для использования в разработке приложений,
требующих литеральные идентификаторы, типа словарей



Проектная группа	--		
Заказчик	Самостоятельный проект		
Проект	Контейнер данных для использования в разработке приложений, требующих литеральные идентификаторы, типа словарей		
Руководитель проекта	Артур Мангус		
Участники проекта	Самостоятельный проект		
Документ	1. Техническое задание	Количество страниц	9
Автор документа	Артур Мангус		
Создан	14.01.2023		
Последнее изменение	16.01.2023		
Статус обработки	X	В обработке Представлен разработчикам и заказчику Одобен Закрит	

История документа

Версия	Дата	Автор изменения	Описание / замечание
0.1	14.01.2023	Артур Мангус	Старт проекта
0.2	15.01.2023	Артур Мангус	Редактирование требований надежности
0.3	16.01.2023	Артур Мангус	Редактирование назначений
0.4	17.01.2023	Артур Мангус	Редактирование назначений

Оглавление

1	Исходная ситуация.....	3
1.1	Общее описание.....	3
1.1.1	Полное наименование системы и ее условное обозначение	3
1.1.2	Шифр темы или шифр (номер) договора	3
1.1.3	Наименование предприятий разработчика и заказчика системы и их реквизиты Артур Мангус Самостоятельный проект	3
1.1.4	Сведения об источниках и порядке финансирования работ	3
1.1.5	Плановые сроки начала и окончания работы по созданию системы	3
1.1.6	Порядок оформления и предъявления заказчику результатов работ	3
2	Основания для разработки	3
	Перечень документов, на основании которых создается система	3
3	Назначение и цели создания (развития) системы.....	4
3.1	Назначение системы	4
3.1.1	Добавление литерального идентификатора (далее «ключа»)	4
3.1.2	Поиск ключа в контейнере.....	5
3.1.3	Вывод данных по заданному ключу	5
3.1.4	Удаление ключа и связанных с ним данных	5
3.1.5	Редактирование данных связанных с ключом	6
3.1.6	Вывод списка всех ключей.....	6
3.1.7	Редактирование ключа	7
3.2	Конструктор и деструктор.....	8
3.2.1	Деструктор.....	8
3.3	Цели создания системы.....	8
4	Характеристики объекта автоматизации	8
4.1	Краткие сведения об объекте автоматизации	8
4.2	Сведения об условиях эксплуатации	8
5	Требования к системе.....	8
5.1	Требования к системе в целом	8
5.1.1	Требования к структуре и функционированию системы	8
5.1.2	Требования к численности и квалификации персонала системы и режиму его работы	8
5.1.3	Требования к надежности	8
5.1.4	Требования к защите информации от несанкционированного доступа и безопасности	9
5.1.5	Требования по сохранности информации при авариях.....	9
5.1.6	Требования к патентной чистоте	9
5.1.7	Требования по стандартизации и унификации	9
5.2	Требования к выполняемым системой функциям	9
6	Состав и содержание работ по созданию (развитию) системы.....	9
7	Порядок контроля и приемки системы.....	9
8	Требования к документированию.....	9
8.1	Требования к тестовым случаям.....	9
8.2	Требования к эксплуатационной документации	10

1 Исходная ситуация

1.1 Общее описание

1.1.1 Полное наименование системы и ее условное обозначение

Полное наименование системы: соответствует наименованию документа.

Краткое наименование системы: Контейнер словарных данных

Условное обозначение: DDC (Dictionary Data Container)

1.1.2 Шифр темы или шифр (номер) договора

DDC23

1.1.3 Наименование предприятий разработчика и заказчика системы и их реквизиты

Артур Мангус

Самостоятельный проект

1.1.4 Сведения об источниках и порядке финансирования работ

Не требуется

1.1.5 Плановые сроки начала и окончания работы по созданию системы

Начало проекта: 14.01.2023

Окончание проекта: 14.03.2023

1.1.6 Порядок оформления и предъявления заказчику результатов работ

Не требуется

2 Основания для разработки

Многие программные продукты нуждаются в использовании связанных контейнеров данных, использующих литералы в качестве идентификаторов (ключей).

Перечень документов, на основании которых создается система

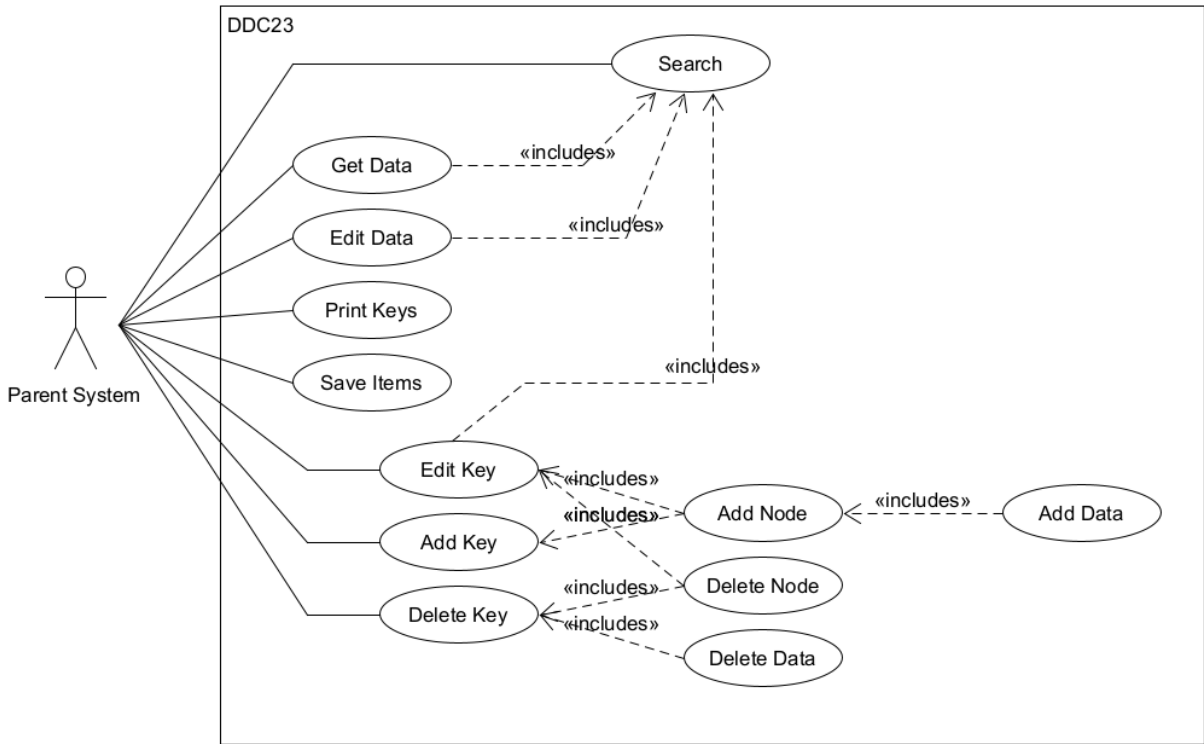
Для реализации требуются:

- «Спецификация DDC23»
- «Эксплуатационная документация DDC23»

Потребность в прочей документации отсутствует.

3 Назначение и цели создания (развития) системы

Раздел описывает функциональные требования системы, представленные в следующей диаграмме прецедентов (UseCase), где Actor Parent System является родительской программой использующей реализацию системы DDC23.



3.1 Назначение системы

В идеальном варианте система DDC23 должна хранить любое количество уникальных литеральных идентификаторов любой длины. Каждый литеральный идентификатор должен быть связан с целевым контейнером данных. Система должна предоставлять следующие функциональные возможности, описанные более подробно в спецификации этого проекта:

3.1.1 Добавление литерального идентификатора (далее «ключа»)

Функция: ADD KEY	ID: DDC23-01	Интерфейс:
Описание: <ul style="list-style-type: none">• Контейнер получает значение ключа в виде строковых данных.• Значение ключа разделяется на узлы, образующие запись ключа в модуле.• Узлы связываются между собой от корневого узла, образуя древовидную структуру.• Последний узел ключа получает ссылку на объект данных, связанных с добавленным ключом. Данные могут не содержать значений. Необходимые требования: <p>Ключ, передаваемый в модуль, может быть получен родительской программой любым путем, например, введен пользователем, получен в процессе сериализации, получен через парсер или по результатам действия других процессов.</p> Ожидаемое поведение: <ul style="list-style-type: none">• Контейнер получает введенную пару ключ/данные или введенный ключ• Добавляет ключ в дерево контейнера• Подтверждает добавление ключа и данных		

Ожидаемый результат:

Ключ и данные сохранены в контейнере

3.1.2 Поиск ключа в контейнере

Функция: FIND	ID: DDC23-02	Интерфейс:
<p>Описание:</p> <ul style="list-style-type: none">• Контейнер получает значение ключа в виде строковых данных.• Контейнер находит первый узел, соответствующий первому литералу ключа• Контейнер перебирает литералы ввода и находит соответствующие узлы ключа продвигаясь от корня дерева к вершине, пока не будет найден последний узел ключа.• Если последний узел ключа не достигнут, контейнер возвращает негативный результат• Если последний узел ключа достигнут, контейнер возвращает положительный результат. <p>Необходимые требования: Ключ, передаваемый в модуль для поиска, может быть получен родительской программой любым путем, например, введен пользователем, получен в процессе сериализации, получен через парсер или по результатам действия других процессов.</p> <p>Ожидаемое поведение:</p> <ul style="list-style-type: none">• Контейнер получает введенный ключ• Обходит дерево контейнера• Возвращает результат обхода <p>Ожидаемый результат: Контейнер возвращает результат поиска</p>		

3.1.3 Вывод данных по заданному ключу

Функция: GET DATA	ID: DDC23-04	Интерфейс:
<p>Описание:</p> <ul style="list-style-type: none">• Контейнер получает введенный ключ• Контейнер проводит проверку на существование редактируемого ключа (DCD23-02) Если ключ отсутствует контейнер прерывает выполнение• Контейнер находит последний узел ключа и возвращает извлеченные данные <p>Необходимые требования: В контейнере должен быть реализован поиск ключей.</p> <p>Ожидаемое поведение:</p> <ul style="list-style-type: none">• Контейнер получает введенный ключ• Обходит дерево контейнера• Возвращает результат данные <p>Ожидаемый результат: Контейнер возвращает данные связанные с ключом</p>		

3.1.4 Удаление ключа и связанных с ним данных

Функция: DELETE KEY	ID: DDC23-03	Интерфейс:
<p>Описание:</p> <ul style="list-style-type: none">• Контейнер получает значение ключа в виде строковых данных.• В контейнере осуществляется проверка на наличие ключа• Обрабатываются два случая:		

- После последнего узла ключа нет продолжения ветки дерева
- После последнего узла ключа есть продолжение ветки дерева
В первом случае удаляется только объект, содержащий данные. Во втором случае удаляются все узлы до следующего узла, содержащего данные.

Необходимые требования:

В контейнере должен быть реализован поиск ключей.

В контейнере должно быть реализовано удаление данных из узла ключа

В контейнере должно быть реализовано удаление отдельных узлов ключей от вершины к корню.

Ожидаемое поведение:

- Контейнер получает введенный ключ
- Контейнер проводит проверку на существование редактируемого ключа (DCD23-02)
Если ключ отсутствует контейнер прерывает выполнение
- Контейнер находит последний узел ключа
Если ключ не найден, контейнер возвращает флаг отрицательного результата
- Контейнер проверяет наличие дочерних узлов у последнего узла ключа
- Контейнер удаляет данные из последнего узла ключа
- Контейнер удаляет узлы ключа от вершины вниз к корню, при условии, что у последнего узла ключа нет дочерних элементов
- Подтверждает удаление ключа

Ожидаемый результат:

Ключ и данные удалены из контейнера

3.1.5 Редактирование данных связанных с ключом

Функция: EDIT DATA	ID: DDC23-05	Интерфейс:
Описание: <ul style="list-style-type: none">• Контейнер получает введенную пару ключ/данные или введенный ключ• Контейнер проводит проверку на существование редактируемого ключа (DDC23-02) Если ключ отсутствует контейнер прерывает выполнение• Если ключ найден, контейнер заменяет данные в последнем узле ключа и возвращает положительный результат Необходимые требования: <p>В контейнере должен быть реализован поиск ключей.</p> <p>В контейнере должно быть реализовано удаление данных из последнего узла ключа</p> <p>В контейнере должно быть реализовано добавление данных к последнему узлу ключа</p> Ожидаемое поведение: <ul style="list-style-type: none">• Контейнер получает введенную пару ключ/данные• Контейнер заменяет данные ключа на новые• Подтверждает добавление данных Ожидаемый результат: <p>Контейнер возвращает новые данные для заданного ключа</p>		

3.1.6 Вывод списка всех ключей

Функция: PRINT KEYS	ID: DDC23-06	Интерфейс:
Описание: <ul style="list-style-type: none">• Контейнер производит обход всего дерева от корня к вершинам и добавляет найденный ключ в несортированный список.• Полученный несортированный список выводится в консоль командой print(). Необходимые требования:		

В контейнере должен быть реализован обход дерева.
В контейнере должна быть реализована функция возврата значения ключа.
В контейнере должна быть реализована функция составления списка ключей.

Ожидаемое поведение:

- Родительская программа ставит запрос на распечатку ключей контейнера
- Контейнер распечатывает ключи

Ожидаемый результат:

В консоль выводится полный список ключей

3.1.7 Редактирование ключа

Функция: EDIT KEY	ID: DDC23-07	Интерфейс:
<p>Описание:</p> <ul style="list-style-type: none">• Контейнер получает значение редактируемого ключа и его нового значения в виде строковых данных.• Контейнер проводит проверку на существование редактируемого ключа• Контейнер проводит проверку на существование нового ключа• Контейнер проводит сравнение введенного существующего и нового ключа на идентичность. Если ключи идентичны, операция прерывается.• Если в контейнере отсутствует ключ с новым значением Контейнер переходит к последнему узлу ключа и переносит его данные в промежуточное хранилище. Далее производятся действия по удалению ключа описанные в DCD23-03 Контейнер создает новый ключ как описано в DCD23-01 и переносит в него данные из промежуточного хранилища. <p>Необходимые требования:</p> <p>В контейнере должен быть реализован поиск ключей. В контейнере должно быть реализовано добавление ключей. В контейнере должно быть реализовано редактирование данных для заданных ключей. В контейнере должно быть реализовано удаление данных из узла ключа В контейнере должно быть реализовано удаление отдельных узлов ключей от вершины к корню. В контейнере должно быть реализовано промежуточное хранение данных для операций редактирования.</p> <p>Ожидаемое поведение:</p> <ul style="list-style-type: none">• Контейнер получает введенную пару ключ/ключ• Контейнер проводит валидацию входных данных и состояния контейнера• Контейнер добавляет новый ключ• Контейнер удаляет старый ключ• Подтверждает редактирование <p>Ожидаемый результат:</p> <p>Контейнер заменит ключ. Поиск по старому ключу вернет негативный результат. Поиск по новому ключу вернет положительный результат. Данные, связанные с новым ключом будут теми же, которые содержал старый ключ.</p>		

3.2 Конструктор и деструктор

3.2.1 Деструктор

Функция: CLEAR	ID: DDC23-10	Интерфейс:
Описание: Контейнер обходит все ветви и удаляет каждую из ветвей от вершин к корню Необходимые требования: Все узлы должны быть связаны указателями Ожидаемое поведение: Родительская вершина удаляет дочерние Ожидаемый результат: Контейнер не содержит узлов и как следствие ключей и их данных		

3.3 Цели создания системы

Использование готового модуля DCD23 вносит дополнительную экономическую эффективность в проекты, осуществляя экономию времени и отсутствие необходимости подключения дополнительных ресурсов разработки.

4 Характеристики объекта автоматизации

4.1 Краткие сведения об объекте автоматизации

Контейнер должен представлять из себя модуль, загружаемый в RAM компьютера во время исполнения программы, и выполняющий функции, описанные в пункте 3.1.

4.2 Сведения об условиях эксплуатации

Система должна позволять использование всех функций в процессе эксплуатации на современных компьютерах всех известных типов.

Особенные требования к условиям эксплуатации отсутствуют.

5 Требования к системе

5.1 Требования к системе в целом

5.1.1 Требования к структуре и функционированию системы

Модуль системы DCD23 должен быть реализован на языке программирования C++.

5.1.2 Требования к численности и квалификации персонала системы и режиму его работы

Для реализации модуля в рамках проекта требуется как минимум один инженер-разработчик ПО. Разработка проекта имеет условные временные рамки, и может использовать время для реализации по усмотрению разработчика.

5.1.3 Требования к надежности

- Контейнер не должен допускать потерь литеральных идентификаторов (ключей) и связанных с ними данных.
- Контейнер не должен допускать произвольного или неконтролируемого изменения литеральных идентификаторов (ключей) и связанных с ними данных.
- Контейнер не должен допускать утечки памяти.
- Контейнер не должен допускать фрагментацию памяти.

5.1.4 Требования к защите информации от несанкционированного доступа и безопасности

Система DCD23 не имеет собственных требований безопасности.

Защиту данных (шифрование, авторизацию, аутентификацию, защищенное хранение и передача) контейнера DCD23 должна обеспечивать родительская система, в которую интегрируется DCD23.

5.1.5 Требования по сохранности информации при авариях

Система должна предоставлять возможность автоматизированного сохранения временных данных на жесткий носитель. В случае отказа системы, контейнер должен иметь возможность восстановить данные из файла временного хранения.

5.1.6 Требования к патентной чистоте

Модуль не должен использовать сторонние библиотеки за исключением стандартных библиотек C++ STD. Модуль не должен использовать код сторонних разработчиков.

5.1.7 Требования по стандартизации и унификации

Модуль должен соответствовать спецификации C++11.

5.2 Требования к выполняемым системой функциям

Все функции системы, работающие со связанными данными, должны предоставлять родительской программе возможность, связывать с ключом любой формат пользовательских данных.

6 Состав и содержание работ по созданию (развитию) системы

ID	Наименование работ	Длительность
SPEC	Разработка спецификации	2 д
TSTD	Разработка тестовых случаев	1 д
IMPL	Имплементация	4 д
TEST	Тестирование	1 д
DOC	Разработка эксплуатационной документации	2 д
Итого:		10 дней

7 Порядок контроля и приемки системы

Применительно к данному проекту не предусматривается порядок контроля и приемки системы.

По окончании проекта в папке проекта должны находиться:

- Этот документ
- Спецификация
- Тестовая документация
- Исходный код контейнера
- Эксплуатационная документация.

В случае отсутствия одного или нескольких компонентов из приведенного списка, проект не может считаться законченным.

8 Требования к документированию

8.1 Требования к тестовым случаям

Тестовые случаи

8.2 Требования к эксплуатационной документации

Эксплуатационная документация может быть, как сгенерированной из исходного кода с помощью макросов Doxygen, так и представлена в виде документа в формате *.pdf.