

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

Институт компьютерных наук и кибербезопасности

Высшая школа технологий искусственного интеллекта

Направление: 02.03.01 «Математика и компьютерные науки»

Технологии разработки ПО

*Построение функциональной модели для процесса
разработки приложения «Словарь-КЧД» с помощью
методологии IDEF0*

Студент,

группы 5130201/20102

_____ Тищенко А. А.

Преподаватель,

к.т.н., доц.

_____ Курочкин М. А.

«_____» _____ 2024г.

Санкт-Петербург, 2024

Содержание

Введение	3
1 Постановка задачи	4
2 Концепция IDEF0	5
2.1 Модель	5
2.2 Блочное моделирование и его графическое представление	5
2.3 Лаконичность и точность	5
2.4 Передача информации	6
2.5 Строгость и формализм	6
2.6 Итеративное моделирование	6
2.7 Отделение «организации» от «функций»	6
3 Синтаксис графического языка IDEF0	8
3.1 Блок	8
3.2 Стрелка	8
3.3 Синтаксические правила	9
4 Семантика языка IDEF0	10
4.1 Семантика блоков и стрелок	10
4.2 Семантические правила блоков и стрелок	11
4.3 Родительская диаграмма	11
4.4 Контекстная диаграмма верхнего уровня	11
4.5 Декомпозиция блоков	12
5 Модель IDEF0 для «Словарь-КЧД»	14
5.1 Контекстная диаграмма верхнего уровня A-0	14
5.2 Дочерняя диаграмма A0	16
5.3 Диаграмма A1	18
5.4 Диаграмма A2	20
5.5 Диаграмма A3	22
5.6 Диаграмма A4	24
5.7 Диаграмма A5	26
Заключение	28

Введение

IDEF (I-CAM DEFinition или Integrated DEFinition) — методологии семейства ICAM (Integrated Computer-Aided Manufacturing) для решения задач моделирования сложных систем, позволяют отображать и анализировать модели деятельности широкого спектра сложных систем в различных разрезах. При этом ширина и глубина обследования процессов в системе определяется самим разработчиком, что позволяет не перегружать создаваемую модель излишними данными.

Общая методология IDEF состоит из трех частных методологий моделирования, основанных на графическом представлении систем

- **IDEF0** используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции.
- **IDEF1** применяется для построения информационной модели, отображающей структуру и содержание информационных потоков, необходимых для поддержки функций системы;
- **IDEF2** позволяет построить динамическую модель меняющихся во времени поведения функций, информации и ресурсов системы.

Методология IDEF0 основана на подходе, разработанном Дугласом Т. Россом в начале 70-ых годов и получившем название SADT (Structured Analysis and Design Technique - метод структурного анализа и проектирования). Основу подхода и, как следствие, методологии IDEF0, составляет графический язык описания (моделирования) систем.

Графический язык - полное и выразительное средство, способное наглядно представлять широкий спектр деловых, производственных и других процессов и операций предприятия на любом уровне детализации. Он обеспечивает точное и лаконичное описание моделируемых объектов, удобство использования и интерпретации этого описания. Язык облегчает взаимодействие и взаимопонимание системных аналитиков, разработчиков и персонала изучаемого объекта (фирмы, предприятия), т.е. служит средством «информационного общения» большого числа специалистов и рабочих групп, занятых в одном проекте, в процессе обсуждения, рецензирования, критики и утверждения результатов.

Для выполнения работы была использована методология IDEF0. Объектом моделирования является процесс разработки приложения «Словарь-КЧД»

1 Постановка задачи

- Изучить методологию IDEF0, опираясь на руководящий документ «Методология функционального моделирования IDEF0», принятый и введенный в действие Постановлением Госстандарта России от 2000 г.
- Построить функциональную модель для процесса разработки приложения «Словарь-КЧД» с помощью методологии IDEF0 с точки зрения программиста-разработчика

2 Концепция IDEF0

Функциональная модель IDEF0 – это набор блоков, каждый из которых представляет собой «черный ящик» со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня.

Разработка модели в IDEF0 представляет собой пошаговую, итеративную процедуру. Наиболее важная функция расположена в верхнем левом углу. А соединяются функции между собой при помощи стрелок и описаний функциональных блоков. При этом каждый вид стрелки или активности имеет собственное значение. Данная модель позволяет описать все основные виды процессов, как административные, так и организационные

Методология IDEF0 основана на следующих концептуальных положениях.

2.1 Модель

Модель – искусственный объект, представляющий собой отображение (образ) системы и ее компонентов. Модель разрабатывают для понимания, анализа и принятия решений о реконструкции (реинжиниринге) или замене существующей, либо проектировании новой системы. Система представляет собой совокупность взаимосвязанных и взаимодействующих частей, выполняющих некоторую полезную работу.

2.2 Блочное моделирование и его графическое представление

Основной концептуальный принцип методологии IDEF – представление любой изучаемой системы в виде набора взаимодействующих и взаимосвязанных блоков, отображающих процессы, операции, действия (определения – см. ниже), происходящие в изучаемой системе. В IDEF0 все, что происходит в системе и ее элементах, принято называть функциями. Каждой функции ставится в соответствие блок. На IDEF0 – диаграмме, основном документе при анализе и проектировании систем, блок представляет собой прямоугольник. Интерфейсы, посредством которых блок взаимодействует с другими блоками или с внешней по отношению к моделируемой системе средой, представляются стрелками), входящими в блок или выходящими из него. Входящие стрелки показывают, какие условия должны быть одновременно выполнены, чтобы функция, описываемая блоком, осуществилась.

2.3 Лаконичность и точность

Документация, описывающая систему, должна быть точной и лаконичной. Многословные характеристики, изложенные в форме традиционных текстов, неудовлетворительны. Графический язык позволяет лаконично, однозначно и точно показать все элементы (блоки) системы и все отношения и связи между ними, выявить ошибочные, лишние или дублирующие связи и т.д.

2.4 Передача информации

Средства IDEF0 облегчают передачу информации от одного участника разработки модели (отдельного разработчика или рабочей группы) к другому. К числу таких средств относятся:

- диаграммы, основанные на простой графике блоков и стрелок, легко читаемые и понимаемые;
- метки на естественном языке для описания блоков и стрелок, а также глоссарий и сопроводительный текст для уточнения смысла элементов диаграммы;
- последовательная декомпозиция диаграмм, строящаяся по иерархическому принципу, при котором на верхнем уровне отображаются основные функции, а затем происходит их детализация и уточнение;
- древовидные схемы иерархии диаграмм и блоков, обеспечивающие обзорность модели в целом и входящих в нее деталей

2.5 Строгость и формализм

Разработка моделей IDEF0 требует соблюдения ряда строгих формальных правил, обеспечивающих преимущества методологии в отношении однозначности, точности и целостности сложных многоуровневых моделей. Эти правила описываются ниже. Здесь отмечается только основное из них: все стадии и этапы разработки и корректировки модели должны строго, формально документироваться с тем, чтобы при ее эксплуатации не возникало вопросов, связанных с неполнотой или некорректностью документации

2.6 Итеративное моделирование

Разработка модели в IDEF0 представляет собой пошаговую, итеративную процедуру. На каждом шаге итерации разработчик предлагает вариант модели, который подвергают обсуждению, рецензированию и последующему редактированию, после чего цикл повторяется. Такая организация работы способствует оптимальному использованию знаний системного аналитика, владеющего методологией и техникой IDEF0, и знаний специалистов – экспертов в предметной области, к которой относится объект моделирования

2.7 Отделение «организации» от «функций»

При разработке моделей следует избегать изначальной «привязки» функций исследуемой системы к существующей организационной структуре моделируемого объекта (предприятия, фирмы). Это помогает избежать субъективной точки зрения, навязанной организацией и ее руководством. Организационная структура должна явиться результатом использования (применения) модели. Сравнение результата с

существующей структурой позволяет, во-первых, оценить адекватность модели, а во-вторых – предложить решения, направленные на совершенствование этой структуры.

3 Синтаксис графического языка IDEF0

Набор структурных компонентов языка, их характеристики и правила, определяющие связи между компонентами, представляют собой синтаксис языка. Компоненты синтаксиса IDEF0 – блоки, стрелки, диаграммы и правила.

3.1 Блок

Блок описывает функцию. Типичный блок показан на рис. 1. Внутри каждого блока помещается его имя и номер. Имя должно быть активным глаголом или глагольным оборотом, описывающим функцию. Номер блока размещается в правом нижнем углу. Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.

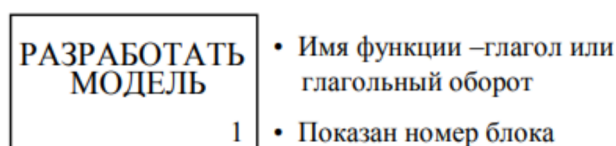


Рис. 1. Устройство блока

3.2 Стрелка

Стрелка формируется из одного или более отрезков прямых и наконечника на одном конце. Как показано на рис. 2, сегменты стрелок могут быть прямыми или ломаными; в последнем случае горизонтальные и вертикальные отрезки стрелки сопрягаются дугами, имеющими угол 90 градусов.

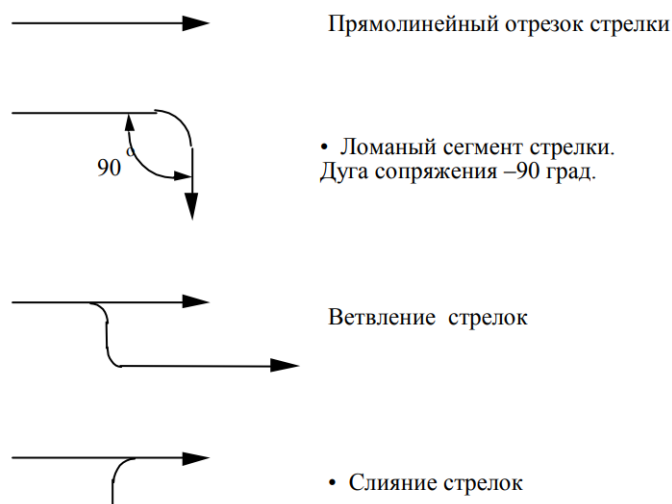


Рис. 2. Виды стрелок

Стрелки не представляют поток или последовательность событий, как в традиционных блок-схемах потоков или процессов. Они лишь показывают, какие данные

или материальные объекты должны поступить на вход функции для того, чтобы эта функция могла выполняться.

3.3 Синтаксические правила

Блоки должны удовлетворять следующим правилам:

1. Размеры блоков должны быть достаточными для того, чтобы включить имя блока.
2. Блоки должны быть прямоугольными, с прямыми углами.
3. Блоки должны быть нарисованы сплошными линиями.

Стрелки должны удовлетворять следующим правилам:

1. Ломаные стрелки изменяют направление только под углом 90 град.
2. Стрелки должны быть нарисованы сплошными линиями различной толщины.
3. Стрелки могут состоять только из вертикальных или горизонтальных отрезков; отрезки, направленные по диагонали, не допускаются.
4. Концы стрелок должны касаться внешней границы функционального блока, но не должны пересекать ее.
5. Стрелки должны присоединяться к блоку на его сторонах. Присоединение в углах не допускается.

4 Семантика языка IDEF0

Семантика определяет содержание (значение) синтаксических компонентов языка и способствует правильности их интерпретации. Интерпретация устанавливает соответствие между блоками и стрелками с одной стороны и функциями и их интерфейсами – с другой.

4.1 Семантика блоков и стрелок

Поскольку IDEF0 есть методология функционального моделирования, имя блока, описывающее функцию, должно быть глаголом или глагольным оборотом; например, имя блока "Выполнить проверку означает, что блок с таким именем превращает непроверенные детали в проверенные. После присваивания блоку имени, к соответствующим его сторонам присоединяются входные, выходные и управляющие стрелки, а также стрелки механизма, что и определяет наглядность и выразительность изображения блока IDEF0.

Чтобы гарантировать точность модели, следует использовать стандартную терминологию. Блоки именуются глаголами или глагольными оборотами и эти имена сохраняются при декомпозиции. Стрелки и их сегменты, как отдельные, так и связанные в «пучок», помечаются существительными или оборотами существительного. Метки сегментов позволяют конкретизировать данные или материальные объекты, передаваемые этими сегментами, с соблюдением синтаксиса ветвлений и слияний.

Каждая сторона функционального блока имеет стандартное значение с точки зрения связи блок/стрелки. В свою очередь, сторона блока, к которой присоединена стрелка, однозначно определяет ее роль. Стрелки, входящие в левую сторону блока - входы.

Входы преобразуются или расходуются функцией, чтобы создать то, что появиться на ее выходе. Стрелки, входящие в блок сверху - управления.

Управления определяют условия, необходимые функции, чтобы произвести правильный выход. Стрелки, покидающие блок справа – выходы, т.е. данные или материальные объекты, произведенные функцией.

Стандартное расположение стрелок показано на рис.3.

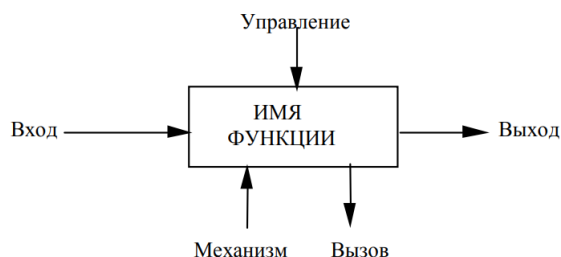


Рис. 3. Семантика блоков и стрелок

Стрелки, подключенные к нижней стороне блока, представляют механизмы. Стрелки, направленные вверх, идентифицируют средства, поддерживающие вы-

полнение функции. Другие средства могут наследоваться из родительского блока. Стрелки механизма, направленные вниз, являются стрелками вызова. Стрелки вызова обозначают обращение из данной модели или из данной части модели к блоку, входящему в состав другой модели или другой части модели, обеспечивая их связь, т.е. разные модели или разные части одной и той же модели могут совместно использовать один и тот же элемент (блок).

4.2 Семантические правила блоков и стрелок

1. Имя блока должно быть активным глаголом или глагольным оборотом.
2. Каждая сторона функционального блока должна иметь стандартное отношение блок/стрелки:
 - а) входные стрелки должны связываться с левой стороной блока;
 - б) управляющие стрелки должны связываться с верхней стороной блока;
 - в) выходные стрелки должны связываться с правой стороной блока;
 - г) стрелки механизма (кроме стрелок вызова) должны указывать вверх и подключаться к нижней стороне блока.
 - д) стрелки вызова механизма должны указывать вниз, подключаться к нижней стороне блока, и помечаться ссылкой на вызываемый блок.
3. Сегменты стрелок, за исключением стрелок вызова, должны помечаться существительным или оборотом существительного, если только единственная метка стрелки несомненно не относится к стрелке в целом.
4. Чтобы связать стрелку с меткой, следует использовать "тильду".
5. В метках стрелок не должны использоваться следующие термины: функция, вход, управление, выход, механизм, вызов

4.3 Родительская диаграмма

Родительская диаграмма – та, которая содержит один или более родительских блоков. Каждая обычная (не-контекстная) диаграмма является также дочерней диаграммой, поскольку, по определению, она подробно описывает некоторый родительский блок. Таким образом, любая диаграмма может быть как родительской диаграммой (содержать родительские блоки), так и дочерней (подробно описывать собственный родительский блок). Аналогично, блок может быть как родительским (подробно описываться дочерней диаграммой) так и дочерним (появляющимся на дочерней диаграмме).

4.4 Контекстная диаграмма верхнего уровня

Каждая модель должна иметь контекстную диаграмму верхнего уровня, на которой объект моделирования представлен единственным блоком с граничными стрелками. Эта диаграмма называется А-0 (А минус нуль). Стрелки на этой диаграмме

отображают связи объекта моделирования с окружающей средой. Поскольку единственный блок представляет весь объект, его имя – общее для всего проекта. Это же справедливо и для всех стрелок диаграммы, поскольку они представляют полный комплект внешних интерфейсов объекта. Диаграмма A-0 устанавливает область моделирования и ее границу.

Каждая модель IDEF0 должна также иметь по крайней мере одну дочернюю диаграмму, содержащую декомпозицию блока A0 на 3 ... 6 дочерних блоков. Этим блокам присваиваются уникальные узловые номера A1, A2, A3, ... A6. Таким образом, последовательность [A0, A1,..., A2,..., A3,...] начинает нумерацию узлов для любой модели

4.5 Декомпозиция блоков

Одной из наиболее важных особенностей методологии IDEF0 является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель. Таким образом обеспечивается представление информации, и читатель располагает хорошо очерченным предметом изучения с приемлемым объемом новой информации на каждой следующей диаграмме.

На рисунке ниже представлена структура IDEF0 модели

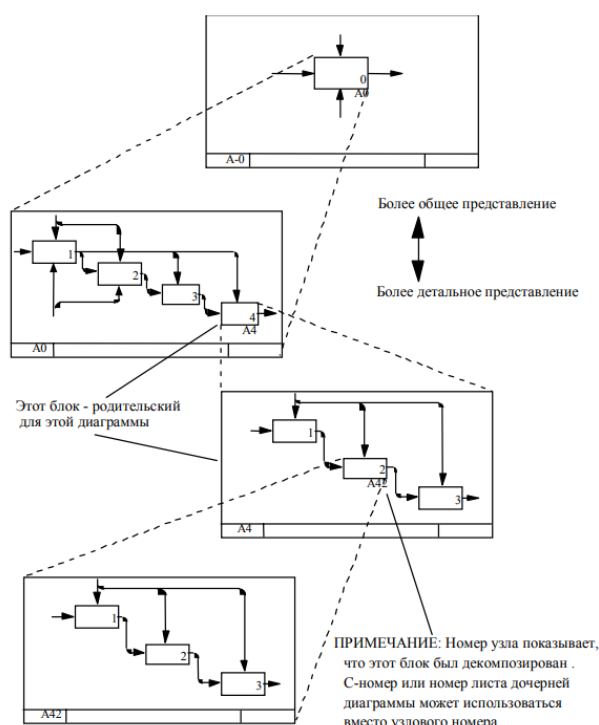


Рис. 4. Иерархия диаграмм

Построение IDEF0-модели начинается с представления всей системы в виде простейшей компоненты - одного блока и дуг, изображающих интерфейсы с функциями вне системы.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки представляют основные подфункции (подмодули) единого исходного модуля.

Данная декомпозиция выявляет полный набор подмодулей, каждый из которых представлен как блок, границы которого определены интерфейсными дугами. Каждый из этих подмодулей может быть декомпозирован подобным же образом для более детального представления.

5 Модель IDEF0 для «Словарь-КЧД»

5.1 Контекстная диаграмма верхнего уровня А-0

На следующей странице представлена диаграмма А-0 для разработки приложения «Словарь-КЧД».

Точка зрения: программист-разработчик.

Входные данные: техническое задание.

Выходные данные: разработанное приложение «Словарь-КЧД».

Управляющие данные (механизмы управления): стандарт программирования C++14.

Механизмы, необходимые для выполнения работы: программист, язык программирования C++, математическое описание КЧД.

Используется в: Курс "Технология разработки программного обеспечения"	Автор: Тищенко Артём	Дата: 01.12.2024	✓	Рабочая версия	Читатель	Дата	Контекст: Верхняя диаграмма
	Проект: Словарь-КЧД	Время: 22:00		Проект			
	Замечания: 1 2 3 4 5 6 7 8 9 10	Версия: 1		Рекомендовано			
				Публикация			

Стандарт разработки C++ 14

↓

Техническое задание

→

Разработать приложение "Словарь-КЧД"

→

Разработанное приложение "Словарь-КЧД"

↑

Программист

↑

Язык программирования C++

↑

Математическое описание КЧД

А₀

ЦЕЛЬ: сокращение времени управления данными в красно-чёрном дереве.

ТОЧКА ЗРЕНИЯ: программист-разработчик

Узел: А-0	Заголовок: Словарь-КЧД	Номер: 1
-----------	------------------------	----------

5.2 Дочерняя диаграмма A0

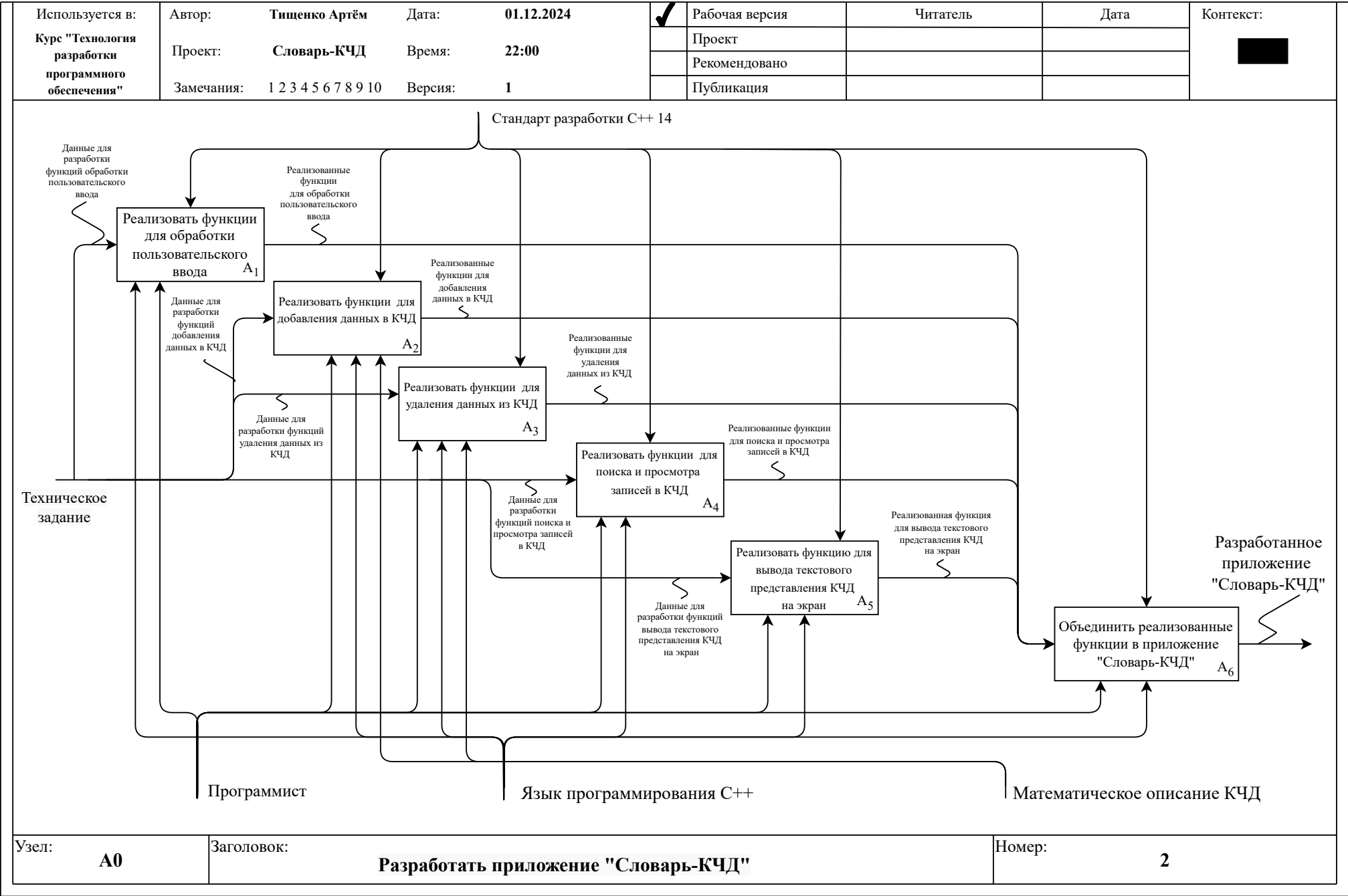
На следующей странице представлена диаграмма декомпозиции первого уровня A0. Данная диаграмма является дочерней диаграммой для A-0, поскольку она детально описывает родительский блок, изображенный на диаграмме A-0, и разбивает функцию «Разработать приложение "Словарь-КЧД"» на 6 подфункций:

- A1. Реализовать функции для обработки пользовательского ввода;
- A2. Реализовать функции для добавления данных в КЧД;
- A3. Реализовать функции для удаления данных из КЧД;
- A4. Реализовать функции для поиска и просмотра записей в КЧД;
- A5. Реализовать функцию для вывода текстового представления КЧД на экран;
- A6. Объединить реализованные функции в приложение "Словарь-КЧД".

Управляющим воздействием всех блоков является стандарт программирования C++14, механизмами являются язык программирования C++, программист и математическое описание КЧД.

На вход блокам подается техническое задание на приложение «Словарь-КЧД». На каждый блок поступают конкретные требования из технического задания, необходимые для разработки конкретных функций системы.

На выходе блоков получают разработанные функции для обработки пользовательского ввода, добавления, удаления и поиска данных в КЧД, а также функции для вывода КЧД на экран.



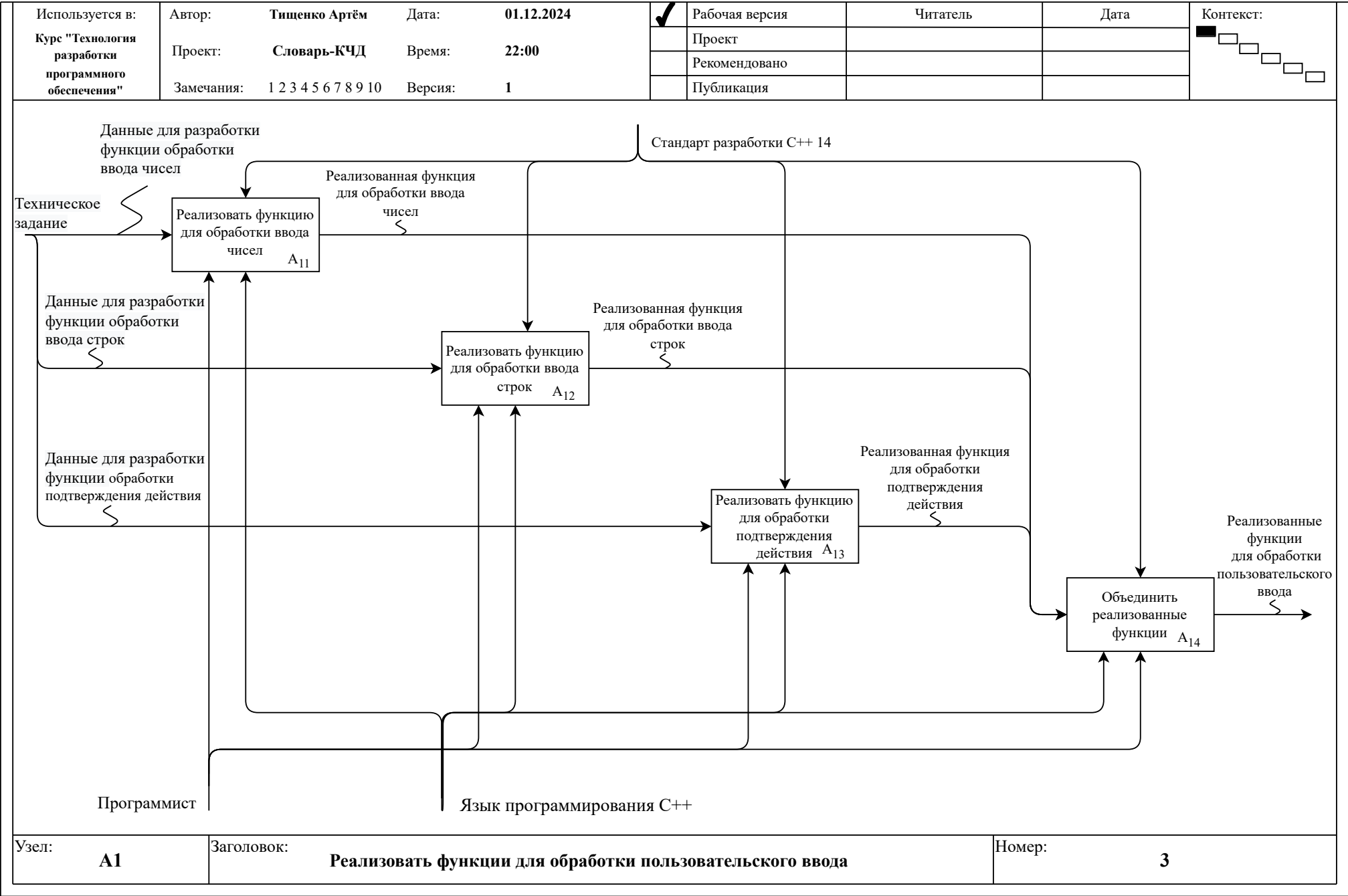
5.3 Диаграмма A1

На следующей странице представлено более подробное представление блока A1 «Реализовать функции для обработки пользовательского ввода». Диаграмма A1 является дочерней диаграммой для A0 и содержит в себе 4 подфункции:

- A11. Реализовать функцию для обработки ввода чисел;
- A12. Реализовать функцию для обработки ввода строк;
- A13. Реализовать функцию для обработки подтверждения действия;
- A14. Объединить реализованные функции.

На вход блокам поступают данные из технического задания приложения «Словарь-КЧД» для разработки каждой конкретной функции.

На выходе блоков получают разработанные функции для обработки ввода чисел, строк и для обработки подтверждения действия пользователем.



5.4 Диаграмма А2

На следующей странице представлено более подробное представление блока А2 «Реализовать функции для добавления данных в КЧД». Диаграмма А2 является дочерней диаграммой для А0 и содержит в себе 3 подфункции:

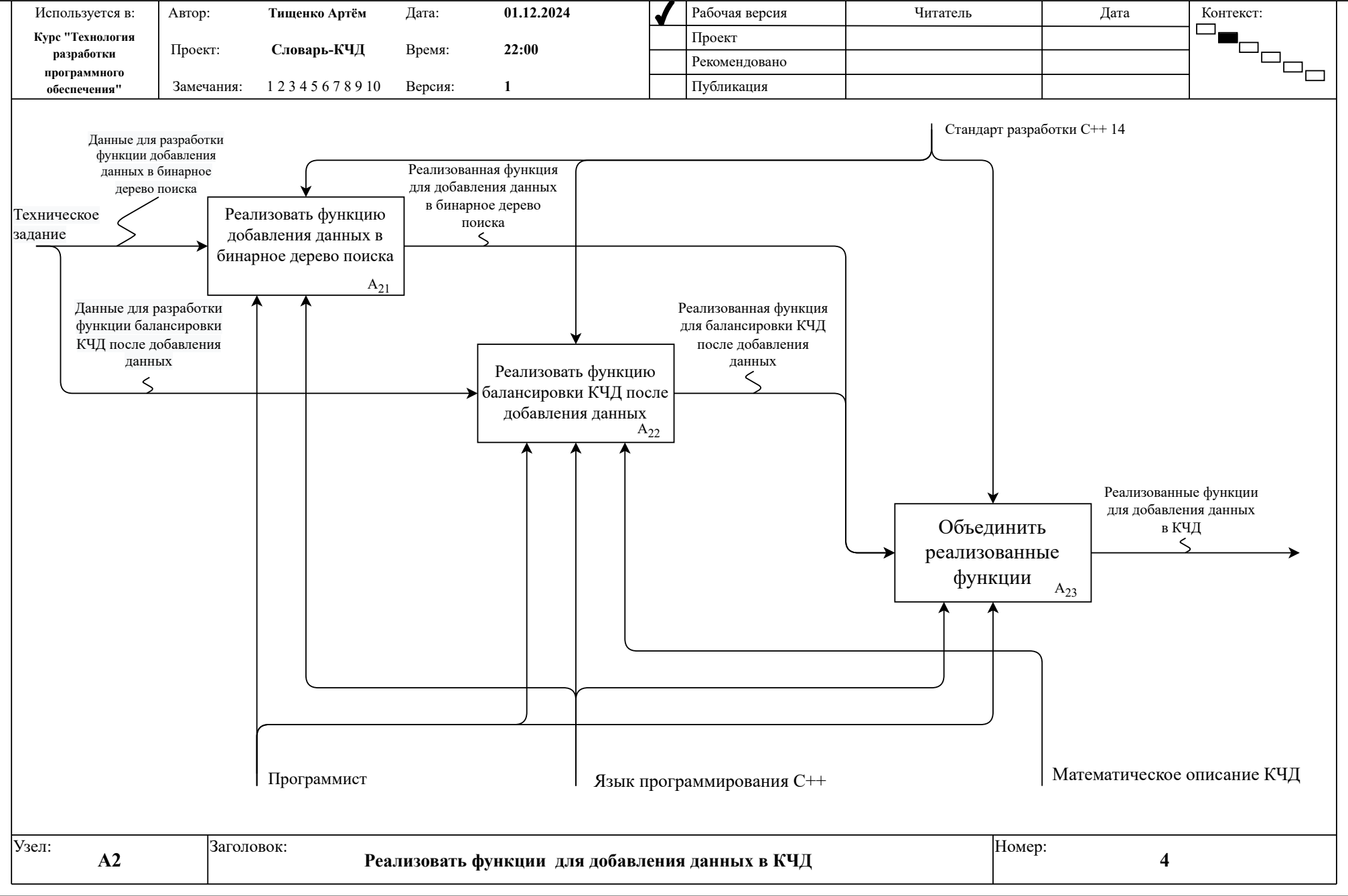
А21. Реализовать функцию добавления данных в бинарное дерево поиска;

А22. Реализовать функцию балансировки КЧД после добавления данных;

А23. Объединить реализованные функции.

На вход блокам поступают данные из технического задания приложения «Словарь-КЧД» для разработки каждой конкретной функции.

На выходе блоков получают разработанные функции для добавления данных в бинарное дерево поиска и балансировки КЧД после добавления данных.



5.5 Диаграмма А3

На следующей странице представлено более подробное представление блока А3 «Реализовать функции для удаления данных из КЧД». Диаграмма А3 является дочерней диаграммой для А0 и содержит в себе 3 подфункции:

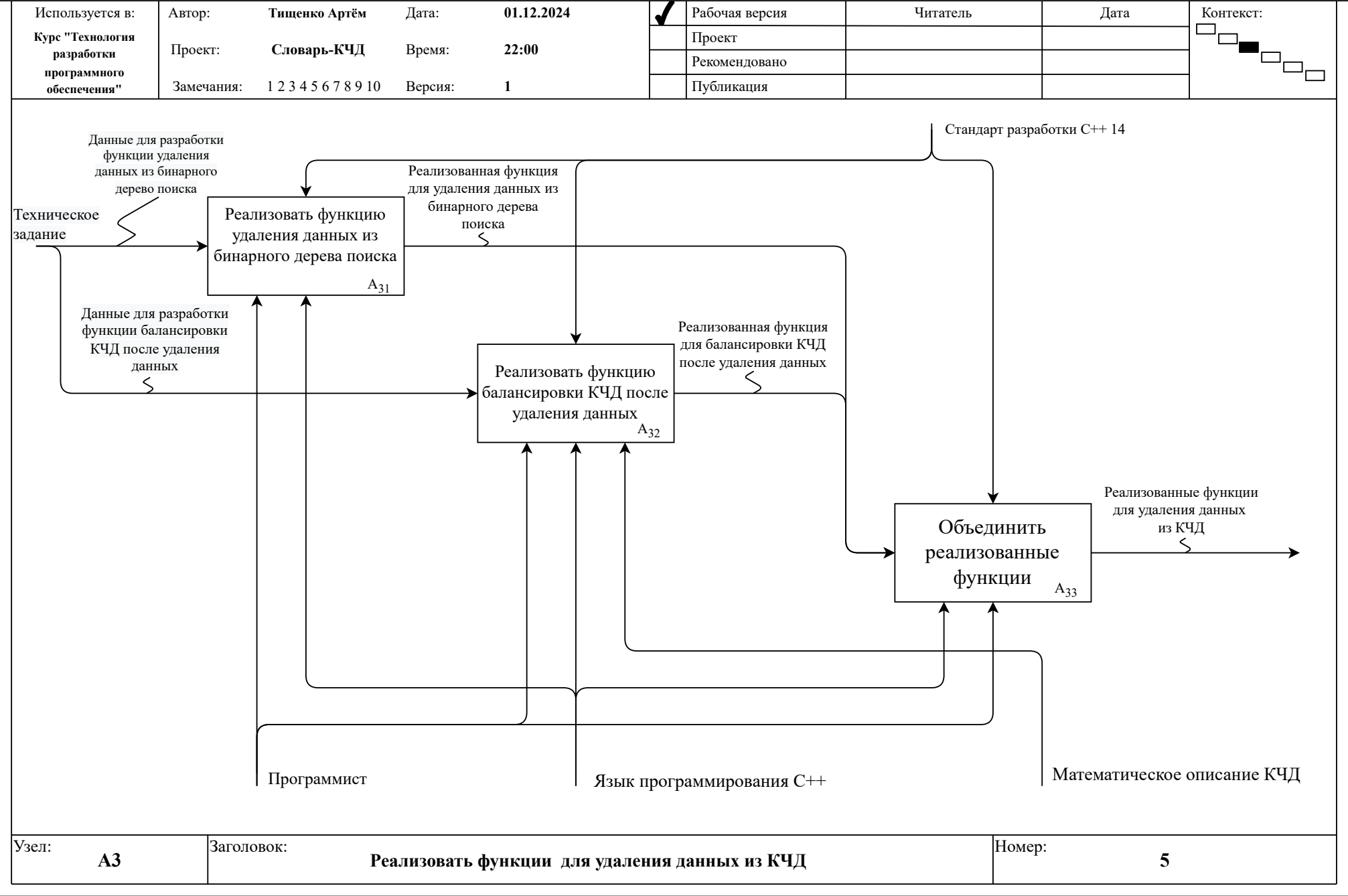
А31. Реализовать функцию удаления данных из бинарного дерева поиска;

А32. Реализовать функцию балансировки КЧД после удаления данных;

А33. Объединить реализованные функции.

На вход блокам поступают данные из технического задания приложения «Словарь-КЧД» для разработки каждой конкретной функции.

На выходе блоков получают разработанные функции для удаления данных из бинарного дерева поиска и балансировки КЧД после удаления данных.



5.6 Диаграмма А4

На следующей странице представлено более подробное представление блока А4 «Реализовать функции для поиска и просмотра записей в КЧД». Диаграмма А4 является дочерней диаграммой для А0 и содержит в себе 3 подфункции:

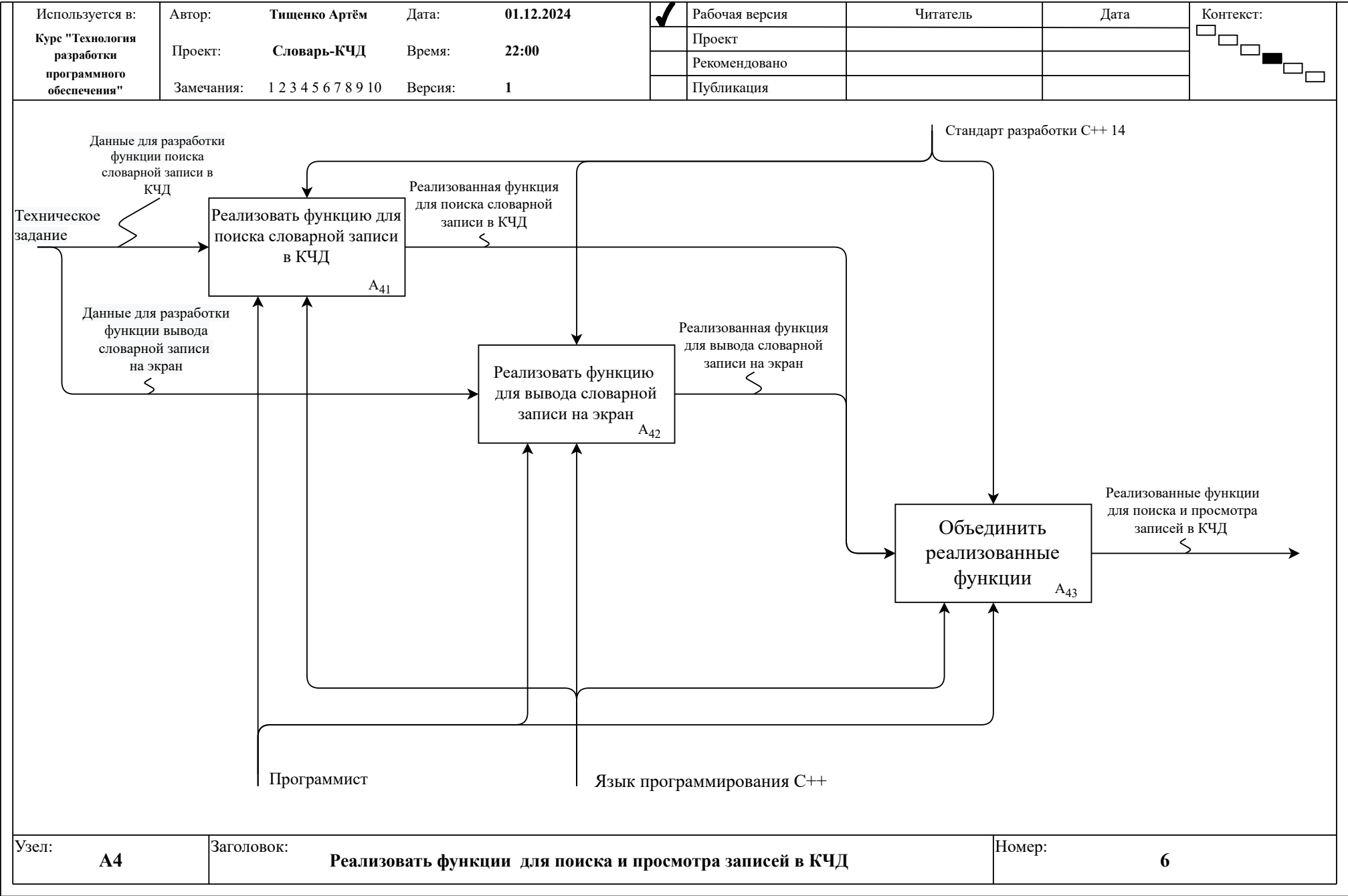
А41. Реализовать функцию для поиска словарной записи в КЧД;

А42. Реализовать функцию для вывода словарной записи на экран;

А43. Объединить реализованные функции.

На вход блокам поступают данные из технического задания приложения «Словарь-КЧД» для разработки каждой конкретной функции.

На выходе блоков получают разработанные функции для поиска словарной записи в КЧД и вывода словарной записи на экран.



5.7 Диаграмма А5

На следующей странице представлено более подробное представление блока А5 «Реализовать функции для поиска и просмотра записей в КЧД». Диаграмма А5 является дочерней диаграммой для А0 и содержит в себе 3 подфункции:

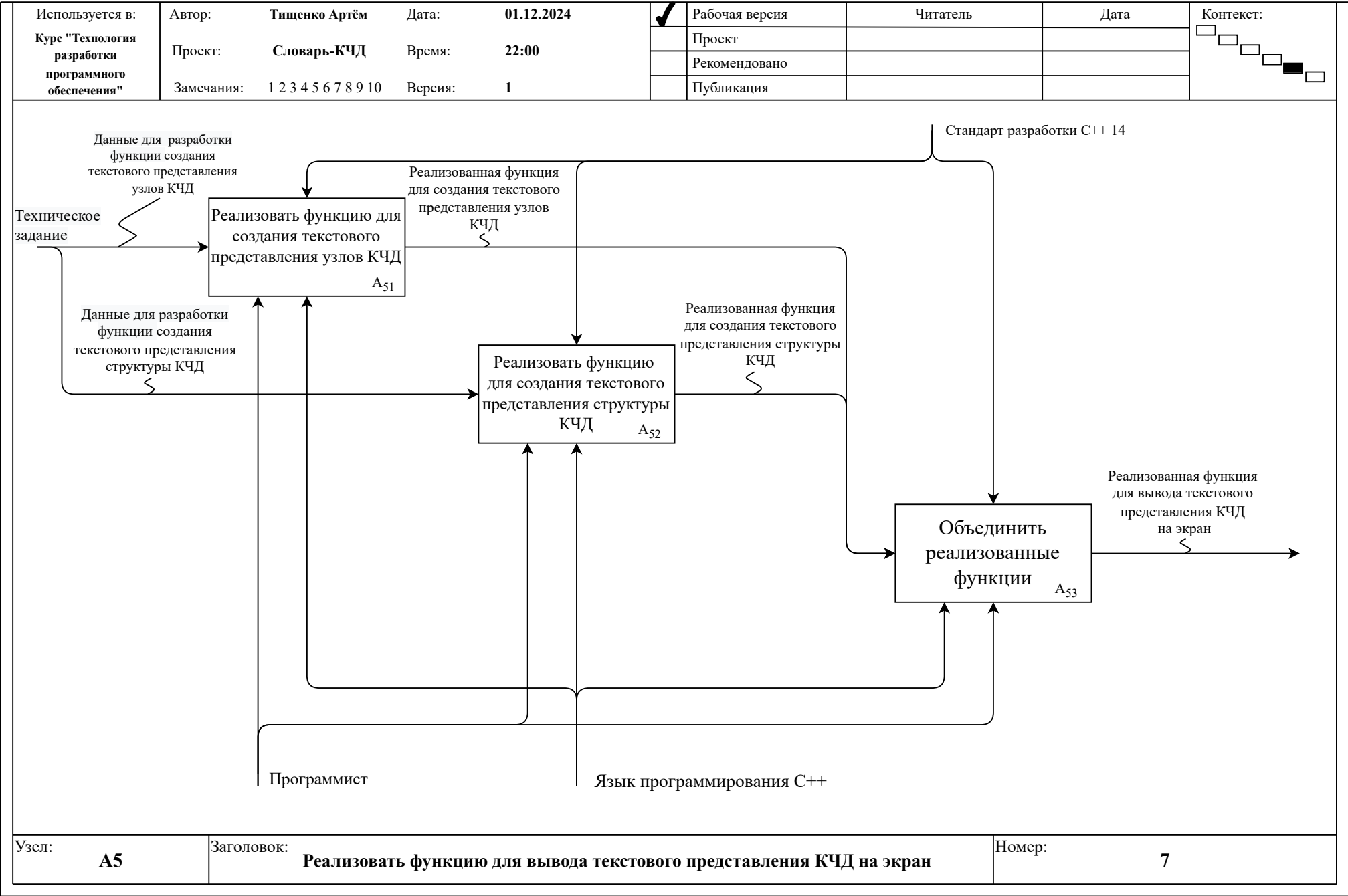
А51. Реализовать функцию для создания текстового представления узлов КЧД;

А52. Реализовать функцию для создания текстового представления структуры КЧД;

А53. Объединить реализованные функции.

На вход блокам поступают данные из технического задания приложения «Словарь-КЧД» для разработки каждой конкретной функции.

На выходе блоков получают разработанные функции для создания текстового представления узлов и структуры КЧД.



Заключение

С помощью методологии функционального моделирования IDEF0 была построена модель демонстрирующая процесс разработки приложения «Словарь-КЧД». Благодаря данной методологии были выявлены и наглядно показаны основные этапы разработки приложения с точки зрения программиста-разработчика. Была построена контекстная диаграмма (родительская диаграмма), проведена ее декомпозиция и построены пять дочерних диаграмм: A0, A1, A2, A3, A4, A5.