

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

«Формальное описание процесса разработки модуля
кодирования и декодирования графа кодом Прюфера с
помощью методологии IDEF0»

Студент,
группы 5130201/20102

_____ Гаар В.С.

Преподаватель,
к.т.н., доц.

_____ Курочкин М.А.

«_____» _____ 2024 г.

Санкт-Петербург, 2024

Содержание

Введение	3
1 Постановка задачи	4
2 Основные понятия методологии и языка IDEF0	5
3 Концепия IDEF0	6
4 Модель IDEF0	8
4.1 Контекстная диаграмма A-0	8
4.2 Диаграммы декомпозиций	10
Заключение	16
Список источников	17

Введение

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.

В нашей работе используется методология IDEF0. Она позволяет исследовать функции организации, не связывая их с объектами, обеспечивающими их реализацию.

IDEF0 используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции.

Модель — искусственный объект, представляющий собой отображение системы и её компонентов.

В данном случае объектом моделирования является процесс разработки модуля кодирования и декодирования графа кодом Прюфера.

1 Постановка задачи

Для выполнения работы были поставлены следующие задачи:

- Изучить методологию IDEF0.
- В рамках этой методологии создать функциональную модель модуля кодирования и декодирования графа кодом Прюфера.

2 Основные понятия методологии и языка IDEF0

Основу методологии IDEF0 составляет графический язык описания процессов.

Графический язык — полное и выразительное средство, способное наглядно представлять широкий спектр деловых, производственных и других процессов и операций предприятия на любом уровне детализации.

Язык обеспечивает точное и лаконичное описание моделируемых объектов, удобство использования и интерпретации этого описания.

Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе.

Контекст — окружающая среда, в которой действует функция (или комплект функций на диаграмме).

Контекстная диаграмма — диаграмма, имеющая узловой номер A-n ($n \geq 0$), которая представляет контекст модели.

Диаграмма A-0 — специальный вид контекстной диаграммы IDEF0, состоящей из одного блока, описывающего функцию верхнего уровня, ее входы, выходы, управления, и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель. [1]

Декомпозиция — разделение моделируемой функции на функции-компоненты.

Блоки представляют основные функции моделируемого объекта. Эти функции могут быть разбиты (декомпозированы) на составные части и представлены в виде более подробных диаграмм. Процесс декомпозиции продолжается до тех пор, пока объект не будет описан на уровне детализации, необходимом для достижения целей конкретного проекта.

Дочерняя диаграмма — диаграмма, создаваемая при декомпозиции, охватывающая ту же область, что и родительский блок, но описывающая ее более подробно.

Таким образом, единственная функция, представленная на контекстной диаграмме верхнего уровня A-0, может быть разложена на основные подфункции посредством создания дочерней диаграммы A0.

Базовые требования стандарта IDEF0:

1. В левом верхнем углу всегда — главный элемент.
2. Все элементы должны иметь входящие и исходящие стрелки, так как для выполнения необходимо что-то получить на входе (заказ, поставленную задачу), а после обработки на выходе необходимо передать готовый продукт. Входящие стрелки всегда слева, исходящие — справа.
3. Сверху — управляющие элементы, снизу — механизмы, необходимые для выполнения процесса.
4. Если на одном листе (экране) располагается несколько блоков, каждый последующий располагается справа и ниже предыдущего.
5. Необходимо стремиться создавать схемы таким образом, чтобы пересечение стрелок было сведено к необходимому минимуму. [1]

3 Концепция IDEF0

Методология IDEF0 основана на следующих концепциях:

1. **Графическое и текстовое представление моделируемой деятельности.**

Графическая и текстовая нотация блочного моделирования в IDEF0-диаграммах показывает производственные операции (блок) и взаимосвязи с операциями (стрелки, входящие/покидающие блок). Наличие четко описанных нотаций обеспечивает корректность встроенных в иерархическую структуру модели диаграмм.

В IDEF0 все, что происходит в системе и ее элементах, принято называть **функциями**. Каждой функции ставится в соответствие **блок**. Для того чтобы представить реальные производственные операции, блоки могут быть интерпретированы как деятельность, связанная с другими блоками, с интерфейсными стрелками, определяющими, когда и как переключаются или управляются операции. Взаимодействие блоков друг с другом описывается посредством **интерфейсных стрелок**, выражающих «ограничения», которые в свою очередь определяют, когда и каким образом функции выполняются и управляются.

2. **Компактность.**

Документация с описанием производственной архитектуры должна быть компактной для простого ориентирования в предмете. Двухмерная форма, описанная на языке диаграмм, достигает компактности без потери возможности выражения отношений, таких как интерфейсы и обратная связь. IDEF0-диаграммы позволяют представить любую изучаемую и/или описываемую систему в виде обеспечивающей компактность информации иерархии взаимодействующих и взаимосвязанных блоков, отображающих процессы, операции, действия.

3. **Функциональная декомпозиция.**

Одной из наиболее важных особенностей методологии IDEF0 является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель.

Функциональная декомпозиция — способ моделирования ситуации, когда любое действие, операция, функция могут быть декомпозированы на более простые действия, операции, функции. Другими словами, сложный процесс может быть представлен в виде совокупности элементарных функций. Представляя функции графически, в виде блоков, можно заглянуть внутрь блока и детально рассмотреть его структуру.

4. **Обмен информацией.**

Диаграммы IDEF0 базируются на простой графике, состоящей из блоков и стрелок, легко читаемы и понимаемы. Средства IDEF0 облегчают передачу информации от одного участника разработки модели (разработчика или рабочей группы) к другому. К числу таких средств относятся:

- метки на естественном языке для описания блоков и стрелок, а также глоссарий и сопроводительный текст для точного определения понятий элементов диаграммы;

- последовательная декомпозиция диаграмм модели, использование иерархии с главной функцией на верху модели и дальнейшее разбитие на подфункции при углублении вниз;
- древовидные схемы иерархии диаграмм и блоков, обеспечивающие обзорность модели в целом и входящих в нее деталей;
- индексирование диаграмм и блоков, позволяющее однозначно обращаться к ним в ие рархической структуре модели;
- ограничения (не более 6 блоков на диаграмму) введены для простого восприятия диаграмм.

5. Строгость, точность, формализм и однозначность.

Разработка моделей IDEF0 требует соблюдения ряда строгих формальных правил, обеспечивающих преимущества методологии в отношении однозначности, точности и целостности сложных многоуровневых моделей. Качество модели обеспечивается соблюдением следующих требований:

- 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;
- Разделение входов и управлений (правило определения роли данных);
- Обязательное наличие управления (все блоки требуют как минимум одного управляющего входа) ;
- Подробное описание на каждом уровне (3-6 блоков).
- Ограниченный контекст (только то, что относится к делу и ничего лишнего, ничего не упущено).
- Требования к меткам дуг данных (правила минимальных меток).

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.

7. Отделение «организации» от «функций».

Отделение организации от функции включено в цель модели и осуществляется отбором имен функций и связей в процессе разработки модели. 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 применения модели. [2]

4 Модель IDEF0

4.1 Контекстная диаграмма A-0

Моделирование процесса начинается именно с построения контекстной диаграммы. Эта диаграмма с единственным блоком определяет контекст всей модели и образует основу для дальнейшей декомпозиции.

На Рис. 1 представлена диаграмма A-0. Контекстная диаграмма устанавливает область моделирования и границу области моделирования. Данная диаграмма задает основные параметры процесса разработки модуля кодирования и декодирования графа кодом Прюфера с точки зрения разработчика-программиста.

Входные данные: техническое задание.

Выходные данные: разработанный модуль кодирования и декодирования графа кодом Прюфера.

Управляющие данные: нормативные документы.

Механизмы: программист, язык программирования C++ и среда разработки Visual Studio 2022.

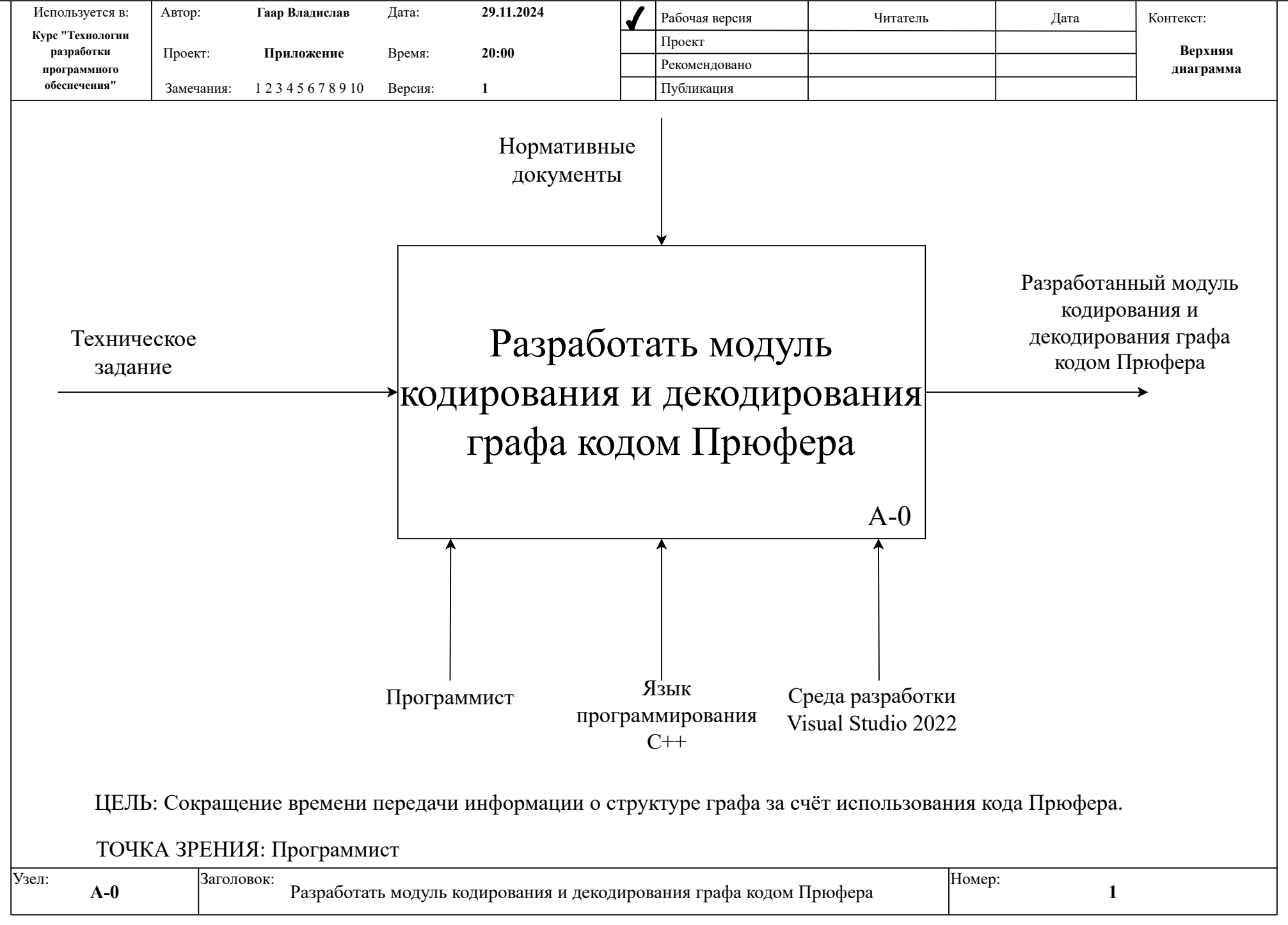


Рис. 1. Контекстная диаграмма А-0

4.2 Диаграммы декомпозиций

4.2.1 Диаграмма A0

На Рис. 2 представлена дочерняя диаграмма A0. Данная диаграмма представляет собой декомпозицию общего блока на элементы, которые связаны между собой.

Были выделены следующие блоки:

- A1. Реализовать функции ввода данных и выбора способа задания графа;
- A2. Реализовать функцию генерации графа;
- A3. Реализовать функцию кодирования графа кодом Прюфера;
- A4. Реализовать функцию декодирования графа из кодов Прюфера;
- A5. Объединить реализованные функции.

Управляющим воздействием всех блоков являются нормативные документы, механизмами являются язык программирования C++ и среда разработки Visual Studio 2022.

На вход блокам подается техническое задание на модуль кодирования и декодирования графа кодом Прюфера. На каждый блок поступают конкретные требования из технического задания, необходимые для разработки конкретных функций системы.

На выходе блоков получают разработанные функции для работы с вводом данных, генерации графа, кодирования и декодирования графа кодом Прюфера.

Стрелки, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются теми же самыми, что и стрелки, входящие в диаграмму нижнего уровня и выходящие из нее, потому что блок и диаграмма представляют одну и ту же часть системы. Как следствие этого, границы функции верхнего уровня – это то же самое, что и границы диаграммы декомпозиции.

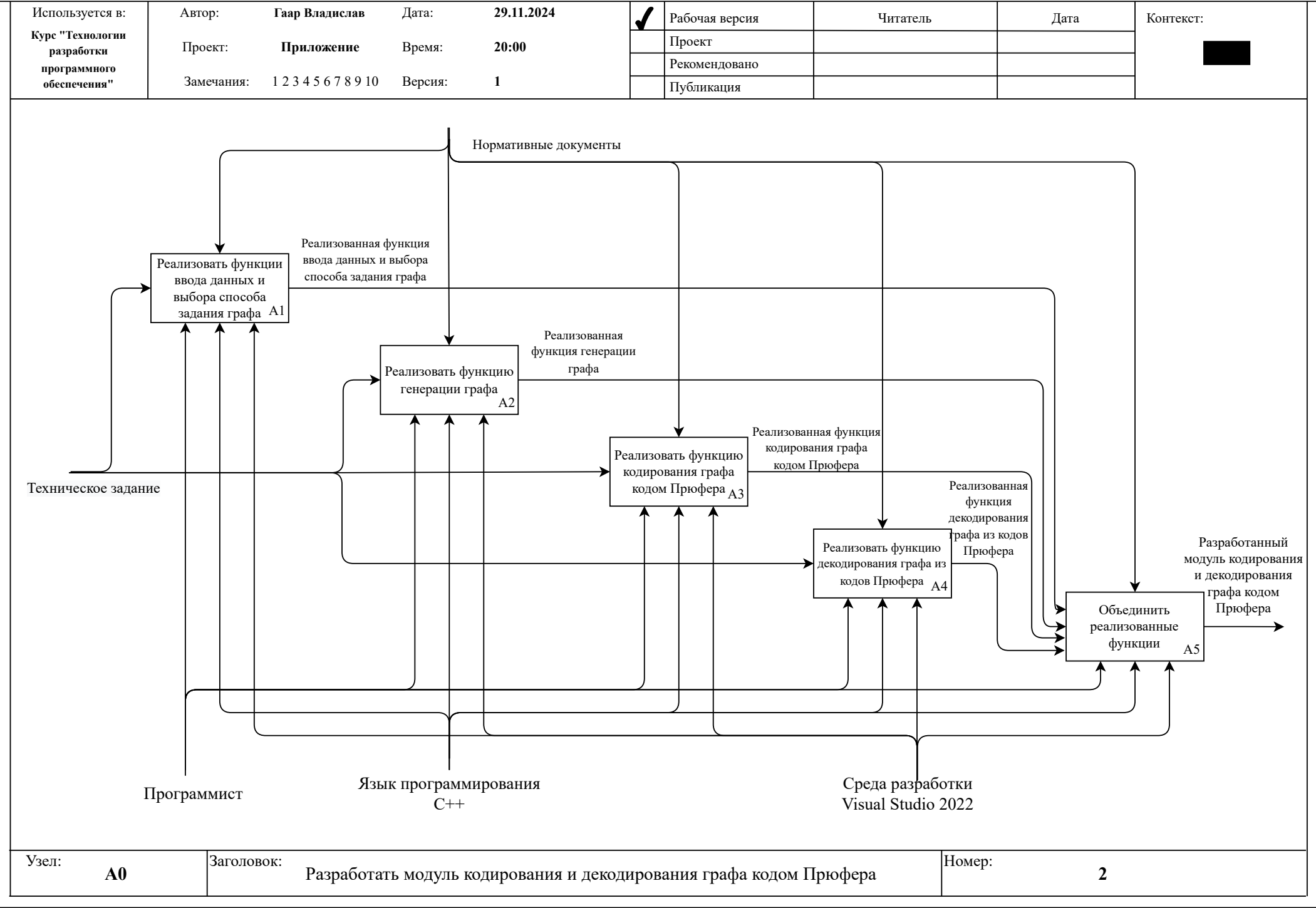


Рис. 2. Диаграмма A0

4.2.2 Диаграмма A1

Блок «Реализовать функции ввода данных и выбора способа задания графа» дочерней диаграммы A0 (см. Рис. ??) может быть разложен на составные части посредством создания дочерней диаграммы следующего, более низкого уровня.

На Рис. 3 представлена дочерняя диаграмма A1 более низкого уровня. Данная диаграмма представляет собой декомпозицию блока реализации функций ввода данных и выбора способа задания графа на элементы, которые связаны между собой.

Были выделены следующие блоки:

A11. Релизовать функцию считывания количества вершин графа;

A12. Реализовать функцию выбора способа задания графа;

A13. Реализовать функцию подтверждения данных;

A14. Объединить реализованные функции.

На вход блокам поступают требования заказчика из ТЗ, относящиеся именно к функциям ввода данных и способу задания графа.

На выходе блоков получают разработанные функции считывания количества вершин графа, выбора способа генерации графа, подтверждения данных.

Используется в: Курс "Технологии разработки программного обеспечения"	Автор: Гаар Владислав	Дата: 29.11.2024	✓	Рабочая версия	Читатель	Дата	Контекст:
	Проект: Приложение	Время: 20:00		Проект			
	Замечания: 1 2 3 4 5 6 7 8 9 10	Версия: 1		Рекомендовано			
				Публикация			

```

graph TD
    NTZ[Техническое задание] --> T3_1[TЗ для разработки функции считывания количества вершин графа]
    NTZ --> T3_2[TЗ для разработки функции выбора способа задания графа]
    NTZ --> T3_3[TЗ для разработки функции подтверждения данных]
    
    T3_1 --> A11[Реализовать функцию считывания количества вершин графа A11]
    T3_2 --> A12[Реализовать функцию выбора способа задания графа A12]
    T3_3 --> A13[Реализовать функцию подтверждения данных A13]
    
    ND[Нормативные документы] --> A11
    ND --> A12
    ND --> A13
    ND --> A14[Объединить реализованные функции A14]
    
    P[Программист] --> A11
    P --> A12
    P --> A13
    P --> A14
    
    C[C++ Язык программирования] --> A11
    C --> A12
    C --> A13
    C --> A14
    
    S[Среда разработки Visual Studio 2022] --> A11
    S --> A12
    S --> A13
    S --> A14
    
    A11 -- "Реализованная функция считывания количества вершин графа" --> A14
    A12 -- "Реализованная функция выбора способа задания графа" --> A14
    A13 -- "Реализованная функция подтверждения данных" --> A14
    A14 -- "Реализованная функция ввода данных и выбора способа задания графа" --> OUT[Выход]
  
```

Узел: A1	Заголовок: Реализовать функции ввода данных и выбора способа задания графа	Номер: 3
-----------------	----------------------------------------------------------------------------	-----------------

Рис. 3. Диаграмма A1

4.2.3 Диаграмма A2

На Рис. 4 представлено более подробное представление блока A2 «Реализовать функцию генерации графа».

Диаграмма A2 является дочерней диаграммой для A0 (см. Рис. 1). Данная диаграмма представляет собой декомпозицию блока реализации функции генерации графа на элементы, которые связаны между собой.

Были выделены следующие блоки:

A21. Реализовать функцию генерации шаблона случайного графа

A22. Релизовать функцию назначения весов рёбрам

A23. Реализовать функцию проверки связности и ацикличности графа

A24. Объединить реализованные функции

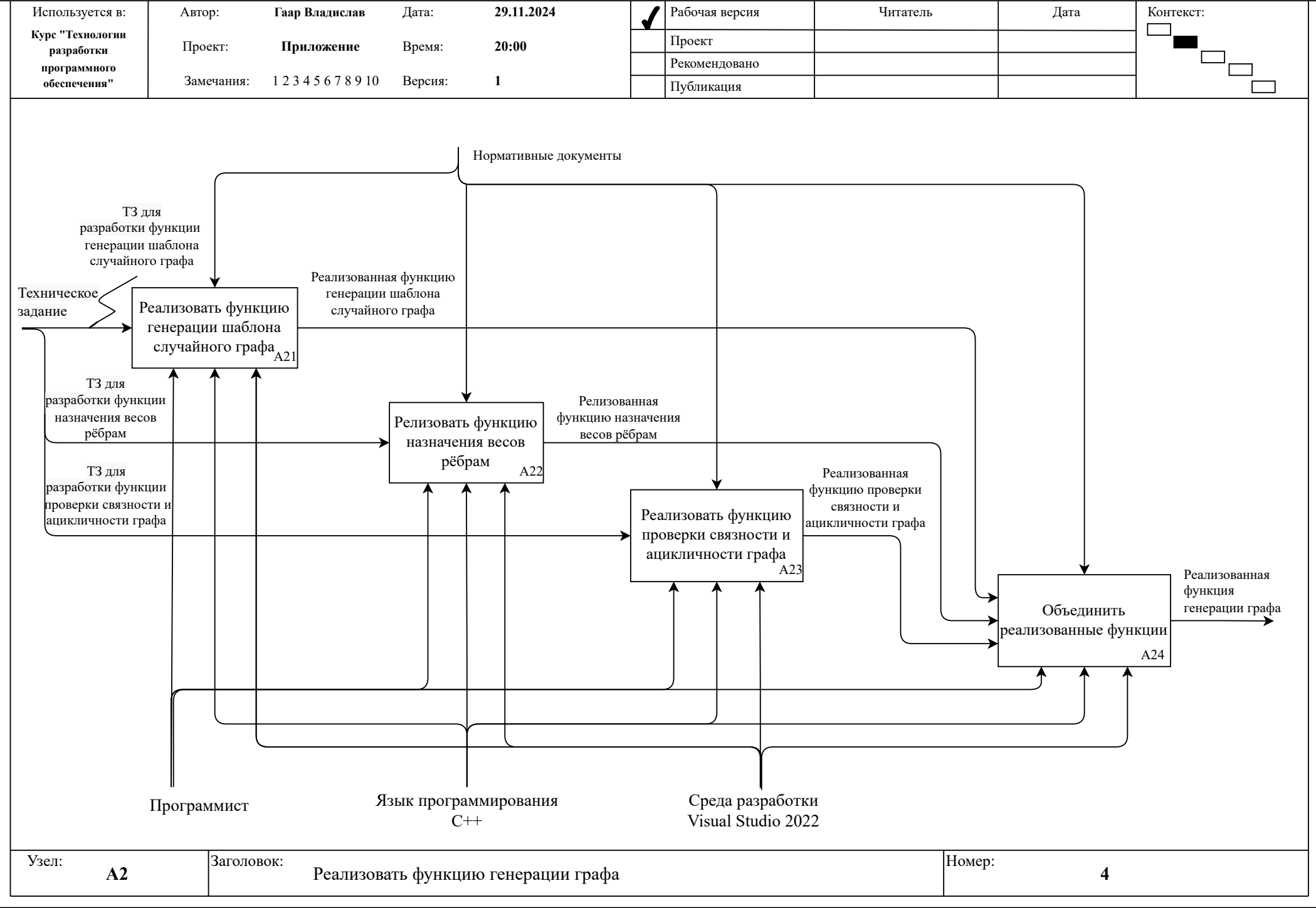


Рис. 4. Диаграмма A2

Заключение

Таким образом, для разработки модели, демонстрирующей процесс создания модуля кодирования и декодирования графа кодом Прюфера, была использована методология функционального моделирования IDEF0.

Благодаря данной методологии были выявлены и наглядно показаны основные этапы разработки приложения с точки зрения разработчика-программиста. Была построена контекстная диаграмма (родительская диаграмма), проведена ее декомпозиция и построены четыре дочерние диаграммы: A0, A1, A2.

Список источников

- [1] ГОССТАНДАРТ РОССИИ. Руководящий документ. МЕТОДОЛОГИЯ ФУНКЦИОНАЛЬНОГО МОДЕЛИРОВАНИЯ IDEF0. Москва: ИПК Издательство стандартов, 2000г., 75с.
- [2] Методология IDEF0. Стандарт. Русская версия. МетаТехнология 1993, 91 с.