

УДК 004.4'2

Юрков В.А., старший преподаватель, ФГБОУ ВО «Луганский государственный университет имени Владимира Даля»

Юрков Д.А., доцент, ФГБОУ ВО «Луганский государственный университет имени Владимира Даля»

Горбунов В.А., ассистент, ФГБОУ ВО «Луганский государственный университет имени Владимира Даля»

ДИАГРАММА КЛАССОВ КАК ЭФФЕКТИВНЫЙ ИНСТРУМЕНТ РАЗРАБОТКИ ИНФОРМАЦИОННЫХ СИСТЕМ

Аннотация: рассматриваются различные аспекты использования диаграммы классов при разработке информационных систем и программного обеспечения. Проанализированы различные возможности использования данного инструмента, даны общие рекомендации для его эффективного использования.

Ключевые слова: программирование, программное обеспечение, диаграмма классов, информационная система

Современные информационные системы состоят из множества различных компонентов, взаимодействующих между собой, и требуют тщательного планирования и проектирования. Разрабатывая информационную систему, важно использовать эффективные инструменты визуализации, которые позволяют упростить и ускорить процесс разработки.

Парадигма ООП (объектно–ориентированное программирование) широко и активно используется при разработке современного программного обеспечения различного уровня сложности. Модель, используемая в данной парадигме, способна достаточно точно описывать свойства и поведение объектов реального мира, которые взаимодействуют друг с другом для достижения поставленной цели разрабатываемой системы.

Однако ООП не является универсальной технологией для решения всех существующих проблем при разработке информационных систем. Отсутствие четкого представления, какие классы нужно реализовать в системе, как их потом использовать и как они должны взаимодействовать друг с другом, может

привести к очень сложной, нестабильной и трудно расширяемой системе с большим количеством потенциальных ошибок, которые будут неожиданно проявляться по мере эксплуатации системы.

Чтобы избежать подобной ситуации, на этапе программирования разрабатываемой информационной системы или отдельных ее компонентов, существенную помощь может оказать такой инструмент UML (Unified Modeling Language, унифицированный язык моделирования), как диаграмма классов [1-6].

Диаграмма классов используется для визуального представления внутренней структуры разрабатываемой системы или отдельных ее компонентов в виде классов и связей между ними, что подчеркивает их функциональные зависимости и взаимодействия. Использование диаграммы классов, дает возможность проанализировать систему, принимая во внимание различные аспекты ее эксплуатации, и откорректировать потенциальные ошибки. Подобный подход является актуальным и помогает упростить разработку системы или отдельных ее частей всеми участниками проектирования.

Рассмотрим фрагмент системы, который реализует иерархию взаимосвязей домашних животных. Имеется следующее текстовое описание: иерархия классов начинается с класса Pet, внутри которого определяется следующее перечисление {UNKNOWN = 0, CAT = 1, DOG = 2, PARROT = 3}. Класс Pet должен содержать поле kind, значением которого может быть только один из элементов данного перечисления и поле name, содержащее имя домашнего питомца. Кроме этого, класс Pet должен определять следующие два метода:

- getName – метод, возвращающий имя домашнего животного;
- getKind – метод, возвращающий вид домашнего животного.

По умолчанию, имя домашнего животного отсутствует, а вид домашнего животного неизвестен.

На данном этапе, создается базовый класс с именем Pet, соответствующий заданному описанию. Но даже из этого небольшого фрагмента видно, что текстовое описание не самый лучший способ задания исходных данных для программирования данной системы, поскольку опущены существенные детали, например, не указана видимость полей и методов, возникает вопрос, что именно (какое значение) должен возвращать метод getKind и какой тип данных должно иметь поле name. Кроме этого, текстовые описания приходится тщательно изучать, чтобы не упустить важные моменты. На диаграмме классов данное описание можно представить следующим образом (рис.1).

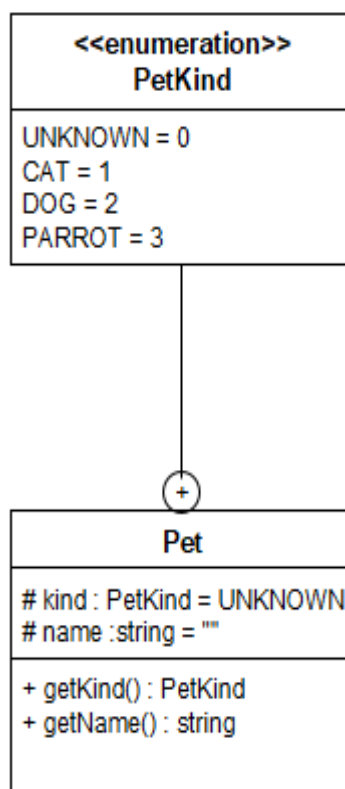


Рис.1. Диаграмма классов для класса Pet

Использование семантики элементов диаграммы классов, легко устраняет все вышеперечисленные недостатки текстового описания и дает четкое визуальное представление исходных данных. Можно продолжить текстовое описание построения иерархии классов домашних животных, например, описать классы, наследуемые от класса Pet, их поля и методы,

функционал и т.д., однако чем система сложнее, тем более сложное текстовое описание она будем иметь.

Воспользуемся возможностями, предоставляемыми диаграммой классов, и расширим данную систему на предлагаемых домашних животных из заданного перечисления, а именно {CAT, DOG, PARROT}. Изобразим ее на диаграмме классов (рис.2).

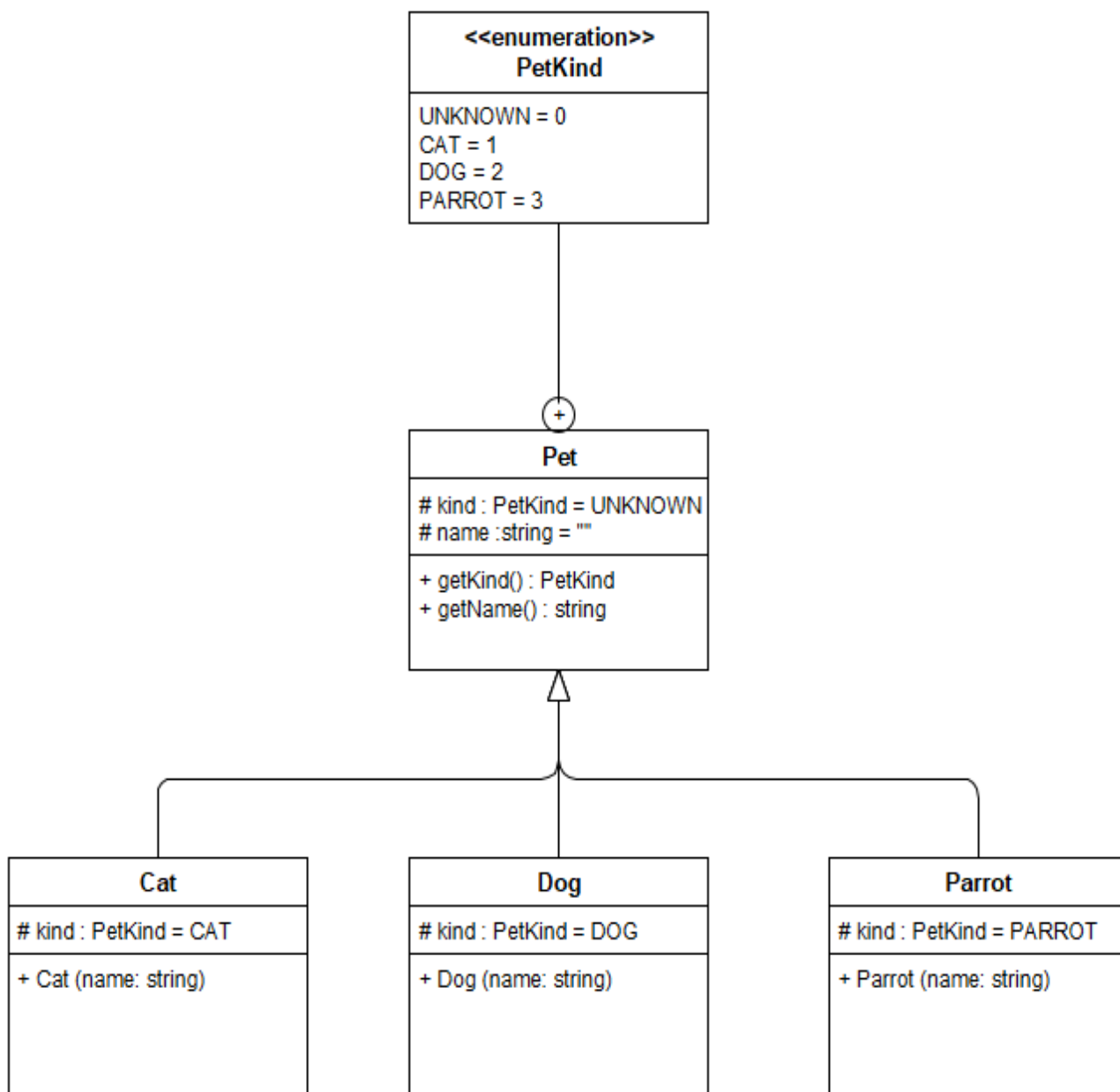


Рис.2. Диаграмма классов иерархии домашних животных

Диаграмма классов иерархии домашних животных показывает, что имя домашнего питомца и его вид задаются при создании экземпляра класса,

который наследуется от класса Pet, при этом, имя задается через конструктор соответствующего класса. Как можно видеть, диаграмма классов очень четко и лаконично позволяет расширить заданную систему и показать взаимосвязи между ее элементами.

Таким образом, с помощью диаграммы классов можно ответить на многие вопросы и проанализировать различные аспекты, возникающие при разработке информационных систем, что было бы затруднительно реализовать другими способами.

Рассмотрим фрагмент другой системы, в которой устанавливаются взаимосвязи между преподавателями и ассистентами высшего учебного заведения, а также предметами, которые они совместно преподают. Имеем следующие исходные данные:

- ассистент может получать задания и лабораторные работы для студентов от преподавателя, с которым он ведет занятия;
- преподаватель может получать отчеты по лабораторным работам и по заданиям от ассистента, с которым он ведет занятия;
- преподаватель вместе с ассистентом могут вести несколько предметов.

Для построения диаграммы классов, формализуем исходные данные следующим образом: имеется три сущности ассистент, преподаватель и предмет. Преподаватель и ассистент могут быть описаны одним классом (класс Teacher), который реализует функционал как преподавателя (интерфейс ILect) так и ассистента (интерфейс IAssist). Таким образом, класс Teacher может играть две роли: преподавателя или ассистента, каждая из которых имеет свой собственный функционал. Кроме этого, класс Teacher имеет ассоциацию с классом, описывающим предмет (класс Subject). Покажем данную информацию на диаграмме классов и проанализируем ее (рис. 3).

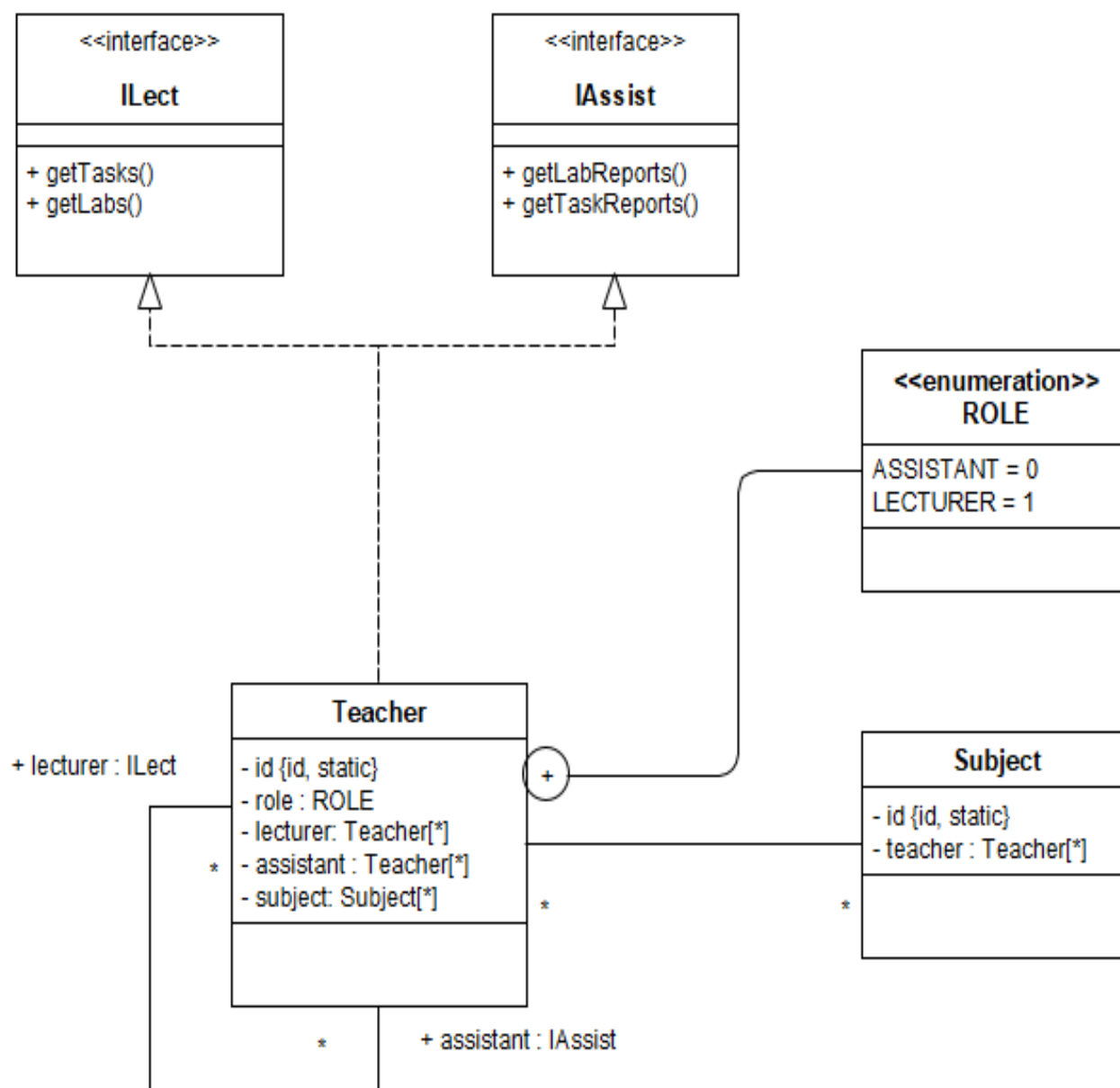


Рис. 3. Диаграмма классов взаимосвязей преподавателей, ассистентов, предметов (начальный вариант)

Анализируя начальный вариант диаграммы, можно видеть, что в классе **Teacher** нет однозначного соответствия между преподавателем ведущим занятия и ассистентом или ассистентами, с которыми он ведет занятия (если класс **Teacher** имеет роль **LECTURER**) и наоборот, нет однозначного соответствия между ассистентом и преподавателями, с которыми он ведет занятия (если класс **Teacher** имеет роль **ASSISTANT**). В данном случае, на диаграмме, мы имеем отношение «многие–ко–многим». Точно такое же отношение существует между классами **Teacher** и **Subject**. На данном примере хорошо видно, как диаграмма

классов позволяет увидеть и проанализировать различные аспекты разработки информационных систем. Для исправления данной ситуации (устранения отношения «многие–ко–многим»), необходимо ввести две новые сущности (два класса–ассоциации): LectAssist (задающий однозначное отношение между преподавателем и ассистентом) и TeacherSubject (задающий однозначное отношение между преподавателем и предметом). Откорректированный вариант диаграммы классов показан на рис.4.

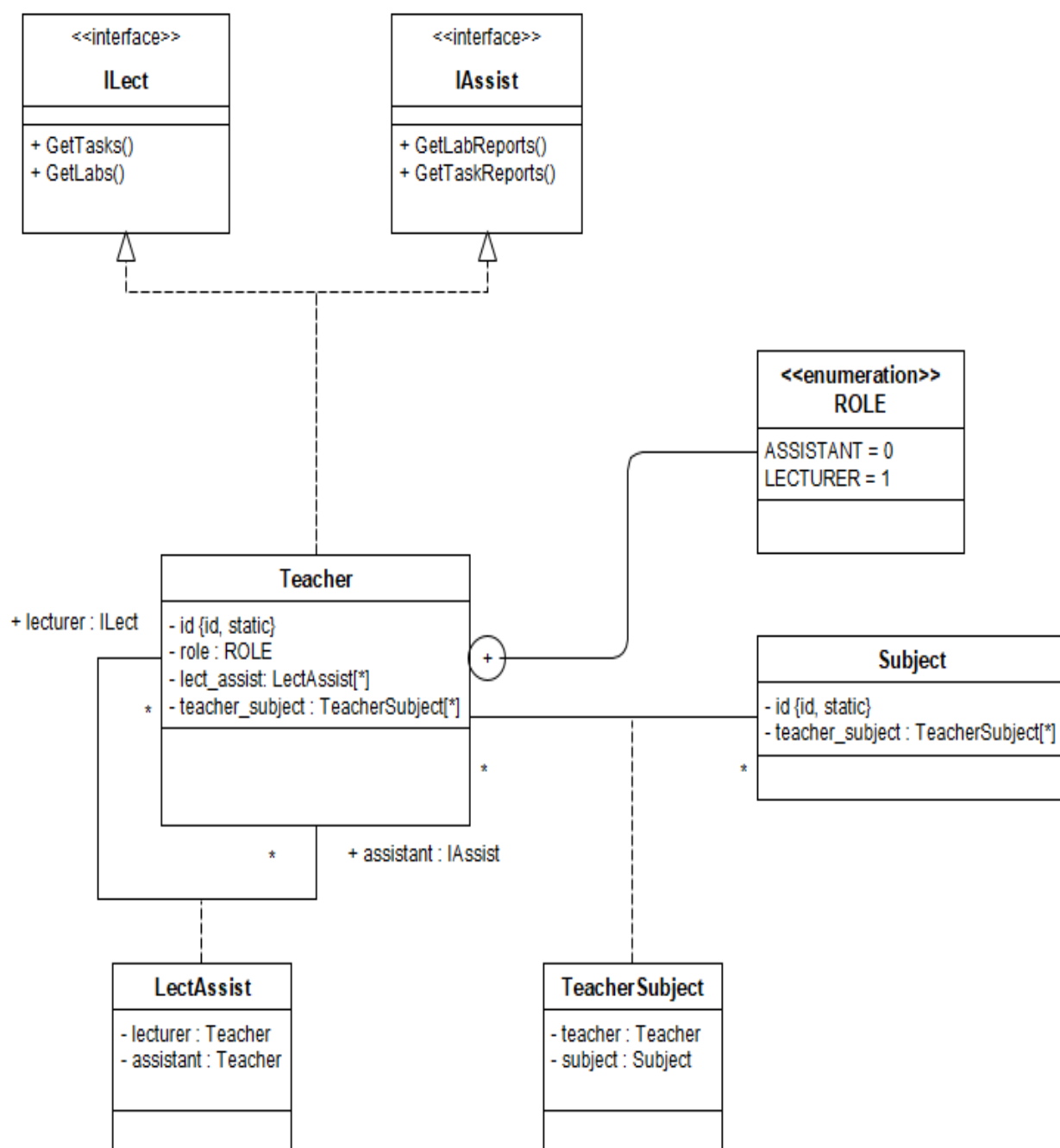


Рис. 4. Диаграмма классов взаимосвязей преподавателей, ассистентов, предметов (откорректированный вариант)

Работая с диаграммами классов, всегда необходимо помнить, что на диаграмме классов не обязана отображаться абсолютно вся информация о каждом классе (все атрибуты и операции). Это зависит от уровня детализации диаграммы, но хорошая диаграмма классов всегда должна точно и четко отражать главную идею, которую хочет донести разработчик.

Для эффективного использования такого мощного инструмента UML как диаграмма классов при разработке информационных систем и программного обеспечения в целом, необходимо уделять внимание следующим аспектам:

1. Определите классы, которые описывают сущности, их атрибуты и операции, необходимые для построения диаграммы классов.

2. Определите уровень детализации диаграммы (в любом случае, диаграмма должна содержать все необходимые компоненты, которые важны для понимания предназначения диаграммы).

3. Определите отношения между классами. При этом необходимо помнить, что различные типы отношений указывают на специфические формы взаимодействия между классами и экземплярами классов.

4. Обращайте особое внимание на кратность отношений. Необходимо избегать отношений «многие–ко–многим».

5. Придерживайтесь принципа «максимальной простоты» при построении диаграммы классов, не упуская при этом важные для понимания диаграммы моменты.

Список использованных источников

1. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд.: Пер. с англ. Мухин Н.-М.: ДМК Пресс, 2006.-496 с.: ил.
2. Леоненков А.В. Самоучитель UML. – 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2004. - 432 с.: ил.
3. Фаулер М. UML. Основы, 3-издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с., ил.

4. Моделирование на UML [Электронный ресурс] URL:
http://book.uml3.ru/sec_1_5 (дата обращения: 15.04.2025 г.)
5. Моделирование на UML [Электронный ресурс] URL:
http://book.uml3.ru/sec_3_2 (дата обращения: 15.04.2025 г.)
6. Моделирование на UML [Электронный ресурс] URL:
http://book.uml3.ru/sec_3_3 (дата обращения: 15.04.2025 г.)
-

Yurkov V.A., senior lecturer, The Federal State Budgetary Educational Institution of Higher Education «Lugansk State University named after Vladimir Dal»

Yurkov D.A., candidate of technical sciences, associate professor, The Federal State Budgetary Educational Institution of Higher Education «Lugansk State University named after Vladimir Dal»

Gorbunov V.A., assistant, The Federal State Budgetary Educational Institution of Higher Education «Lugansk State University named after Vladimir Dal»

CLASS DIAGRAM AS AN EFFECTIVE INSTRUMENT FOR INFORMATION SYSTEM DEVELOPMENT

Abstract: The article describes different aspects related to using a class diagram for the information system and software development. The different possibilities of using this tool are analyzed. The recommendations for this tool to be effectively used are formulated.

Keywords: programming, software, class diagram, information system