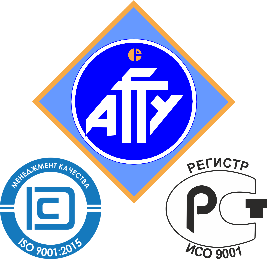
***Федеральное агентство по рыболовству***

***Федеральное государственное бюджетное образовательное***

***учреждение высшего образования***

***«Астраханский государственный технический университет»***

**Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована**

**ООО «ДКС РУС» по международному стандарту ISO 9001:2015**

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**ОТЧЁТ**

**о выполнении практической работы №1 (по профилю специальности)**

**По теме : «UML диаграммы и техническое задание»**

Специальность: 09.02.07 «Информационные системы и программирование»

Отчёт выполнен студентом:

\_\_\_\_\_\_\_\_\_\_\_ Смирновым Р.А.

(подпись) Ф.И.О.

Группа: ДКМО - 42

«05» сентября 2024г.

| Руководитель:  \_\_\_\_\_\_\_\_\_ Бондаренко И.А.. |
| --- |

Астрахань 2024

**1.UML ДИАГРАММА**

**1.1. UML**

UML (Unified Modeling Language) — унифицированный язык моделирования. Это графический язык, который с помощью диаграмм и схем описывает разнообразные процессы и структуры. UML не является языком программирования, но чаще всего применяется в IT — с его помощью можно автоматически генерировать код. Кроме IT, UML используется в проектировании, документировании и построении бизнес-процессов. Плюсы UML:

- упрощает сложности при разработке ПО;

- автоматизирует производство программного обеспечения и процессов;

- помогает решить постоянные проблемы с архитектурой;

- улучшает качество работы;

- сокращает затраты и время выхода на рынок;

**1.1 Виды диаграмм в UML**

Итак, приступим к изучению и обзору диаграмм UML. Все UML диаграммы по своей сущности делятся на два вида (см. рисунок 1.1).

Структурные диаграммы - описывают структуру сложных объектов и систем, показывают статическую структуру системы и ее частей на разных уровнях абстракции и реализации, а также их взаимосвязь

Диаграммы поведения - иллюстрируют взаимодействие с системой и процесс её работы, основное внимание здесь уделяется динамическим аспектам до системы программного обеспечения или процесса

К структурным диаграммам относят следующие 7 типов диаграмм:

- диаграмма составной структуры

- диаграмма развертывания

- диаграмма пакетов

- диаграмма профилей

- диаграмма классов

- диаграмма объектов

- диаграмма компонентов

А к диаграммам поведения относят следующие типы диаграмм:

- диаграмма деятельности

- диаграмма прецедентов

- диаграмма состояний

- диаграмма последовательности

- диаграмма коммуникаций

- диаграмма обзора взаимодействия

- временная диаграмма

**1.2. Диаграмма классов**

Диаграмма классов описывает типы объектов системы и различного рода статические отношения, которые существуют между ними. На диаграммах классов отображаются свойства классов, операции классов и ограничения, которые накладываются на связи между объектами.

На рисунке ниже изображена модель класса обработки заказов клиентов. Прямоугольники на диаграмме представляют классы и разделены на три части: имя класса (жирный шрифт), его атрибуты и его операции. На рисунке также показаны два вида связей между классами: ассоциации и обобщения. (см. рисунок 1.2).

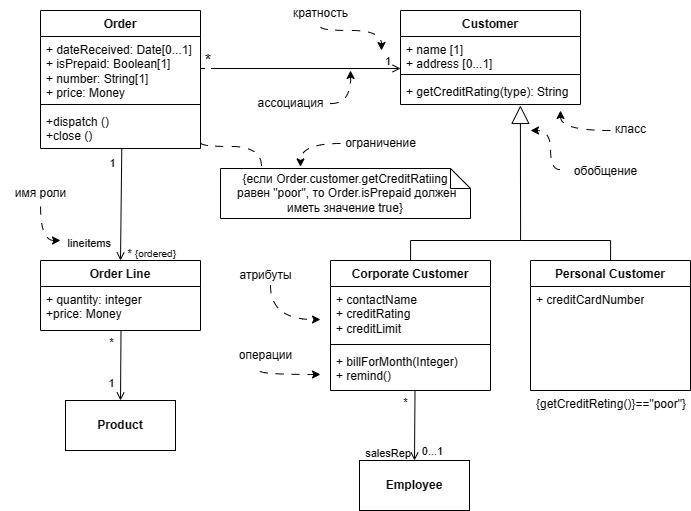


Рисунок 1.2 – «Диаграмма классов»

Свойства**:**

- представляют структурную функциональность класса. Можно рассматривать свойства как поля класса.

- свойства представляют единое понятие, воплощающееся в двух совершенно различных сущностях: в атрибутах и в ассоциациях. Хотя на диаграмме они выглядят совершенно по-разному, в действительности это одно и то же.

Атрибут:

- описывает свойство в виде строки текста внутри прямоугольника класса.

Ассоциация:

-непрерывная линия между двумя классами, направленная от исходного класса к целевому классу.

- имя свойства (вместес кратностью) располагается на целевом конце ассоциации.

- целевой конец ассоциации указывает на класс, который является типом свойства.

Двунаправленная ассоциация:

- пара свойств, связанных в противоположных направлениях. Класс Car (Автомобиль) имеет свойство owner:Person[1], а класс Person (Личность) имеет свойство cars:Car[\*].

Кратность:

- свойства обозначает количество объектов, которые могут заполнять данное свойство.

Операции:

- представляют собой действия, реализуемые некоторым классом. Существует очевидное соответствие между операциями и методами класса. Обычно термины операция и метод употребляются как взаимозаменяемые, однако иногда полезно их различать.

Обобщение:

- объединяет несколько подклассов в один класс. Так, в нашем примере обобщение объединяет индивидуального и корпоративного клиентов некоторой бизнес-системы. Несмотря на определенные различия, у них много общего. Одинаковые свойства можно поместить в базовый класс Customer (Клиент), при этом класс Personal Customer (Индивидуальный клиент) и класс Corporate Customer (Корпоративный клиент) будут выступать как подтипы.

Примечания:

- комментарии на диаграммах. Примечания могут существовать сами по себе или быть связаны пунктирной линией с элементами, которые они комментируют. Они могут присутствовать на диаграммах любого типа.

**2.ТЕХНИЧЕСКОЕ ЗАДАНИЕ.**

Техническое задание (ТЗ) — это документ, который подробно описывает требования к проекту или продукту. В нем указываются цели, задачи, функциональные и технические характеристики, сроки выполнения, критерии приемки и другие важные аспекты. ТЗ служит основой для разработки и контроля качества работы, обеспечивая понимание между заказчиком и исполнителем.

Техническое задание (ТЗ) — это официальный документ, который детально описывает все требования к проекту или продукту. Оно служит основой для всех этапов разработки.

Цели и задачи:

- определяют, что именно должно быть достигнуто.

Функциональные требования:

- описывают, какие функции должен выполнять продукт.

Технические требования:

- указывают на спецификации, такие как производительность, совместимость и т.д.

Ограничения:

- включают в себя бюджетные, временные и ресурсные ограничения.

Критерии приемки:

- условия, при которых работа будет считаться завершенной.

**2.1. Для чего нужно ТЗ**

Техническое задание необходимо для:

Планирования:

- помогает разбить проект на этапы и определить сроки выполнения.

Коммуникации:

- обеспечивает ясное понимание между всеми участниками проекта, включая заказчика и исполнителя.

Контроля качества:

- устанавливает стандарты, по которым будет оцениваться готовый продукт.

Управления рисками:

- позволяет заранее выявить и учесть потенциальные проблемы.

Документирования:

- служит официальной записью всех требований и изменений в проекте.

**2.2. ГОСТы ТЗ**

ГОСТы — это государственные стандарты, которые регулируют составление и содержание технических заданий. Основные ГОСТы для ТЗ включают:

- ГОСТ 34.602-89: «Техническое задание на создание автоматизированной системы». Этот стандарт определяет структуру и содержание ТЗ для автоматизированных систем, включая требования к функциональности, надежности, безопасности и т.д.

- ГОСТ 19.201-78: «Техническое задание на создание программных средств». Описывает требования к документированию программных продуктов, включая описание функций, интерфейсов и сред выполнения.

Эти стандарты помогают унифицировать процесс разработки, обеспечивая четкость и полноту документации, что в свою очередь способствует успешной реализации проектов.

IEEE STD 830-1998 - Достаточно хорошее определение стандарта 830-1998 — IEEE Recommended Practice for Software Requirements Specifications дано в самом его описании: Описывается содержание и качественные характеристики правильно составленной спецификации требований к программному обеспечению (SRS) и приводится несколько шаблонов SRS. Данная рекомендуемая методика имеет своей целью установление требований к разрабаты

Стандарт IEEE 29148-2011 - обеспечивает единую трактовку процессов и продуктов, используемых при разработке требований на протяжении всего жизненного цикла систем и программного обеспечения. Он приходит на смену стандартов IEEE 830-1998, IEEE 1233-1998, IEEE 1362-1998. Данный стандарт содержит два шаблона спецификации требований:

- System requirements specification (SyRS)

- Software requirements

RUP - Структура SRS в RUP(Rational Unified Process) представляет собой документ, в котором необходимо описать артефакты, полученные в процессе специфицирования требований. Шаблон SRS в RUP адаптирован из стандарта IEEE STD 830 и содержит два варианта:

- Традиционный шаблон SRS со структурированными функциональными требованиями по функциям Системы, максимально похож на 830 стандарт.

- Упрощенный шаблон