

Содержание:

Введение

В век высоких технологий как никогда раньше в современном мире необходима автоматизация производств, в любой сфере от медицины до логистики без информационных систем не обойтись, они позволяют избегать человеческих фактор и увеличивать мощности предприятий.

В данной курсовой я не буду углубляться в такое обширное понятие как автоматизация, а лишь ограничусь её технической составляющей в момент проектирования. Примером информационной системы проектирования в моем случае будет UML - Unified Modeling Language (унифицированный язык моделирования). UML позволяет, легко проектировать информационные системы используя всю мощь объектно-ориентированного подхода.

Используя UML при проектировании мы визуализируем будущую систему одновременно дополняя её документацией, по сути, мы используем конструктор проектирования, отображая данные в объектно-ориентированном стиле, именно поэтому эта система набрала большую популярность, за свою легкость в использовании.

Язык моделирования UML состоит из словаря и правил пользования, комбинирующих входящие в него слова и выдавая осмысленные конструкции в объектно-ориентированном стиле. Такое концептуальное проектирование информационных систем позволяет составлять системы с большим количеством взаимосвязанных моделей. В применении к информационным системам это означает, что с помощью UML можно с разных сторон анализировать представления архитектуры системы на протяжении всего цикла ее разработки.

Цель моей работы в том чтобы описать основные принципы проектирования с применением объектно-ориентированного подхода используя унифицированный язык моделирования.

Задачи работы

Обзор основных применений и определений объектно-ориентированного стиля:

- Исследование процесса разработки информационных систем с применением UML
- Использование основных инструментов унифицированного языка моделирования
- Обзор основных понятий UML
- Применение UML на практике.

Выбранная мной тема разделена на две главные части:

- 1) основы объектно-ориентированного подхода, а также проведен обзор унифицированного языка моделирования как составляющей процесса проектирования
- 2) моделирование информационной системы на примере службы доставки

В заключении я сделал выводы по эффективности UML системы в настоящее время и её перспективы дальнейшего развития.

Анализ и проектирования информационных систем с применением UML

Объектно-ориентированный стиль

Объектно-ориентированный стиль описывает точную структуру объектов взаимосвязанных между собой и составляющих общую архитектуру приложения, построенную на атрибутах и операциях. В объектной модели системы описаны понятия, и объекты реального мира из которых состоит разрабатываемая системы.

В такой информационной системе отражается практическая составляющая разрабатываемой системы, что характеризуется использованием терминологии прикладной области. Основная задача проектируемой системы является построение бизнес-модели [\[1\]](#)организации в её объектном стиле.

Базовыми определениями объектного подхода являются объект и класс, причем одно не может существовать без другого, точнее объект не может быть создан, если он не использует класс. Сама концепция объектно-ориентированного подхода строиться на четырех китах: Инкапсуляции, Наследовании, Полиморфизма и Абстракции;

Инкапсуляция

Инкапсуляция — возможность позволяющая спрятать всю сложность реализации используемого программного интерфейса внутри системы и взаимодействовать с ним посредством простого интерфейса, предоставляющего спецификацию объекта.

Наследование

Наследование — возможность описывать новый класс на основе родительского с передачей всего функционала от ребенка к родителю.

Полиморфизм

Полиморфизм — объектов с единой спецификацией иметь разную реализацию. В упрощенной форме это можно описать, как возможность разных объектов могут выполнять одни и те же действия.

Абстракция

Абстракция — придание объекту свойств, отличающих его от всех других объектов, четко разделяя его концептуальные границы. Основная идея абстракции состоит в том, чтобы разделить способ использования комбинированных объектов данных от деталей их реализаций.

Унифицированный процесс как методология разработки

UML[2] - это система обозначений, которую можно применять для объектно-ориентированного проектирования используя вышеописанные возможности объектной модели - (Инкапсуляции, Наследования, Полиморфизма, Абстракции). Каждый унифицированный процесс описывает этапы системы в визуальной составляющей диаграмме объектов.

Унифицированный процесс (Unified Process) построения процессов реализации жизненного цикла программной разработки представляющей методологию, содержащую ряд этапов разработке, от внедрению и до эксплуатации информационных систем. Она отвечает "на вопросы когда, как, кто, что и с помощью чего реализуется проект[30], а именно содержит описание:

Технологический процесс

Технологический процесс отвечает на вопрос “когда?” и представляет собой последовательности различных типов работ, способных решить поставленную

задачу. Технологический процесс в контексте UML отображается в виде диаграммы классов и объектов, на которых видны все этапы работ и их последовательность для каждой стадии разработки информационной системы;

Виды деятельности

Виды деятельности отвечают на вопрос “как?” и представляют собой задачи, осуществляемые исполнителями (рис. 1).

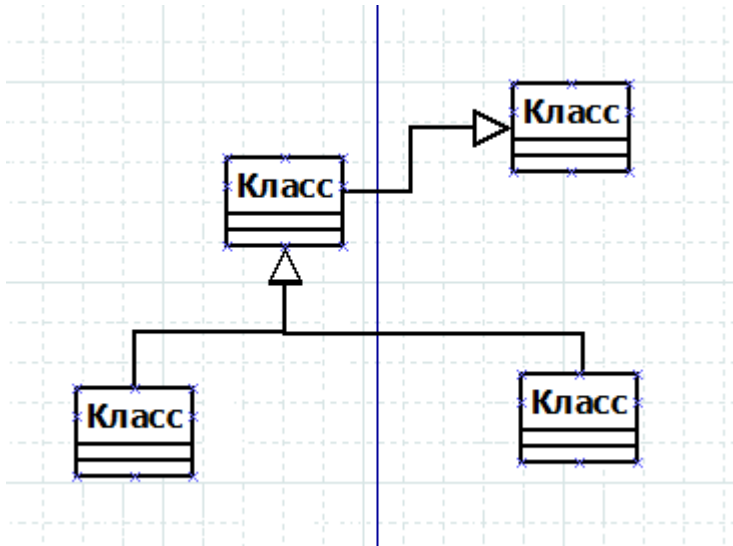


рис. 1. – Вид деятельности (Dia - редактор схем и диаграмм).

Исполнители

Исполнители отвечают на вопрос “кто?” и представляют собой отдельные лица или группы лиц, выполняющие реализацию проекта. Исполнитель характеризуется строгими обязательствами (ролями) и набором функций (видов работ), который определяет его поведение. Во время разработки проекта исполнитель может выступать в разных ролях (рис. 2).

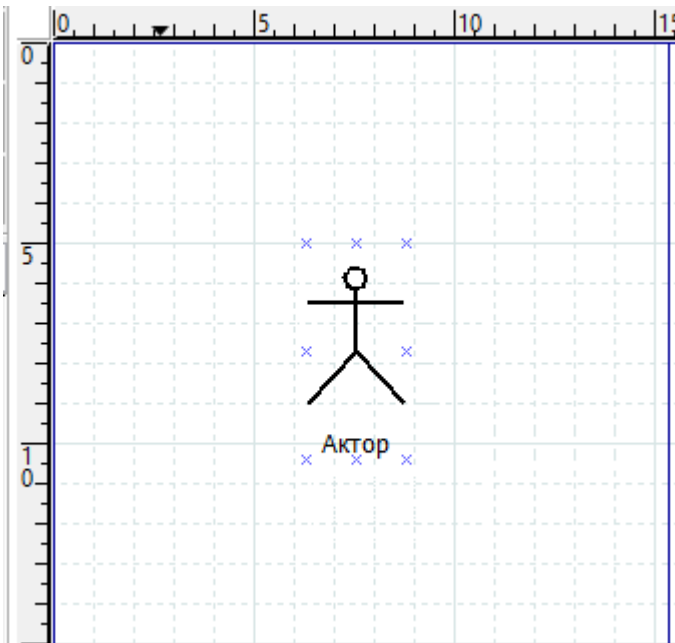


Рисунок 2 – исполнитель (Dia - редактор схем и диаграмм).

Артефакты

Артефакты отвечают на вопрос что и представляют из себя набор информации, изменяемой, создаваемой или используемой исполнителями в реализовываемом проекте. Следует уточнить, что артефакт – это не только то, что образуется в результате выполнения задач (рис. 3).

- технические артефакты – модели систем, исходники программ, готовый технический продукт, документация к проекту и т.п), но и то, что направляет эту деятельность
- артефакты управления – план разработки, тех. задание, инструкции и т. п.;

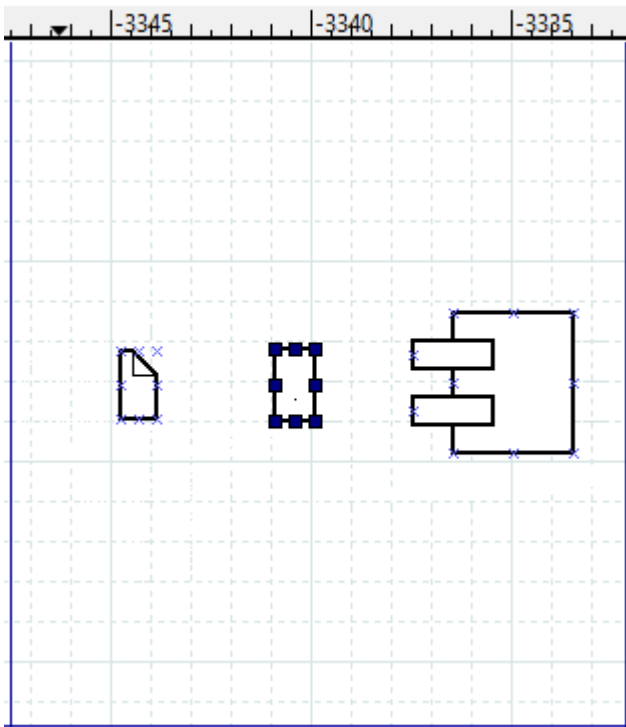


Рисунок 3 – артефакты (Dia - редактор схем и диаграмм).

Используемые утилиты

Используемые утилиты, отвечающие на вопрос “с помощью чего?” – программных софт, который применяется при выполнении работ.

Архитектура унифицированного процесса

Архитектура унифицированного процесса состоит из следующих информационных составляющих:

- интерактивное проектирование;
- проектирование основывается на требованиях, реализация и модификация которых постоянно мониториться[3]. Главные требования являются функциональные особенности системы;
- модели системы осуществляют модульную архитектуру, позволяющую продуктивно распределять задания между исполнителями, при этом достигая максимальной степени повторного использования сделанных компонентов в сопутствующих проектах; Такую архитектуру часто применяют в Фреймворках [4].
- использование визуального моделирования при создании согласованной концепции всех направлений разрабатываемой системы;

- технологические процессы разрабатываются с применением множества утилит которые не должны различаться у исполнителей

Спиральная модель жизненного цикла

Унифицированный процесс можно представить в виде спиральной модели жизненного цикла используемого программного обеспечения.

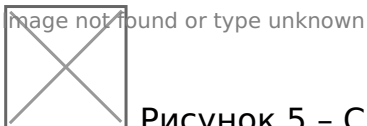


Рисунок 5 – Спиральная модель жизненного цикла

Любой из витков расширяющейся спирали обозначает итеративное увеличение возможностей функциональности системы при схожем наборе технологических фаз (рис. 5).

Внутри каждой фазы реализация также происходит по спиральной модели. В начале каждого старта каждого этапа вычисляется количество итераций, с заданным увеличением степени преимуществ поставленной задачи. Если рассмотреть итерацию изнутри, то в ней осуществляются все процессы, начиная от формирования и заканчивая внедрением требований к системе.

Этапы жизненного цикла

В унифицированном процессе применяется временное разбиение жизненного цикла на несколько основных этапов по порядку:

1. начало;
2. уточнение;
3. конструирование;
4. переход.

Каждая фаза оканчивается достижением результата используемого для управления следующими процессами или же завершает проектирование оптимизированного процесса (рис. 6). На UML диаграмме технологического процесса можно посмотреть весь жизненный цикл бизнес процесса (рис. 7).

Image not found or type unknown

Рисунок 6 – Распределение процессов

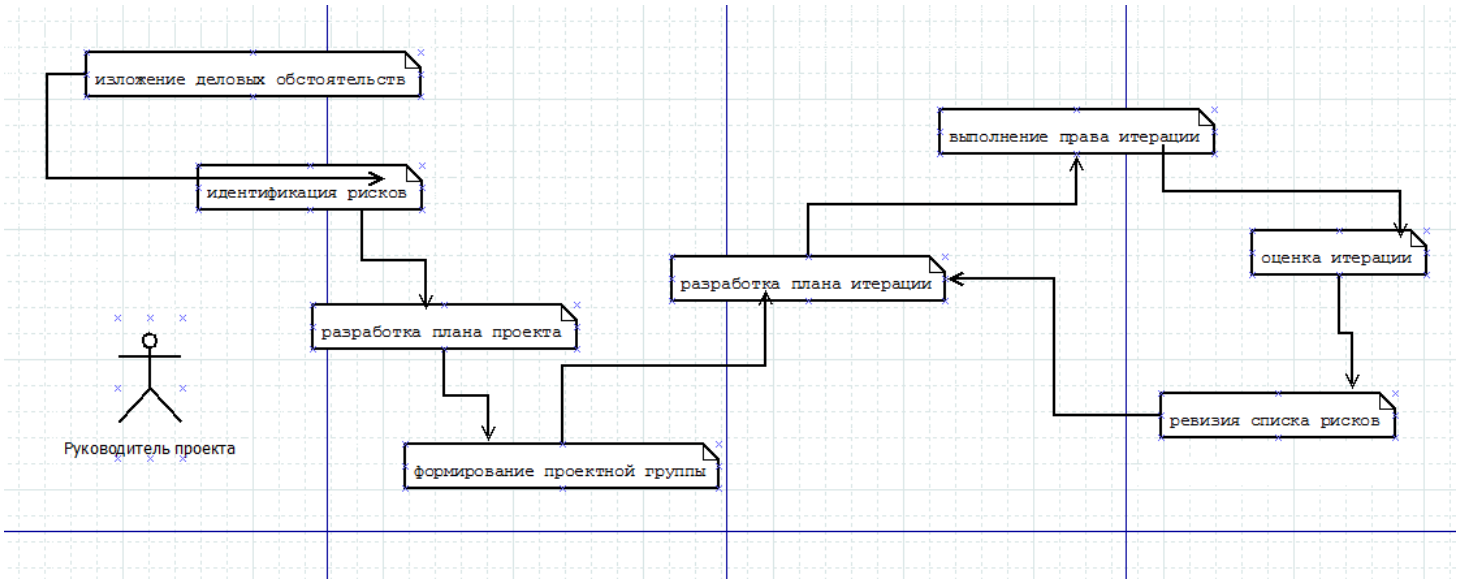


Рисунок 7 – Диаграмма технологического процесса(Dia - редактор схем и диаграмм).

Все виды функционирования направлены на создание артефактов, основным из которых и является сама система, но здесь надо уточнить, что со стороны исполнителя разработанные модели информационных систем также очень важны потому что, они выступают в качестве управляющей и направляющей информации, а также фиксируют результат всех видов отношений.

В унифицированном процессе модели, отображаются главные технологические процессы, каждый из которых характеризуется набором согласованных Unified Modeling Language диаграмм и документов, представленных в таблице ниже.

Модели унифицированного процесса

Таблица 1

Процесс	Модель	Описание
---------	--------	----------

Формирование требований	Варианты использования	Отображает значимые функциональные требования к разрабатываемой системе в форме, наглядной для всех заинтересованных лиц. Под значимыми имеются в виду требования, реализация которых принесет пользователям существенный и ощутимый. Модель обладает минимальной формализованностью, однако именно она управляет всем процессом проектирования
Анализ требований	анализ	Конкретизирует варианты использования с позиции формирования внутренней архитектуры системы: состава классов анализа и отношений между ними.
Проектирование	проектирование	Детализирует представление алгоритмов работы системы и внутренней архитектуры. Используется внутри организации, проектирующей систему
Реализация	реализация	Описывает исполняемую систему: компоненты (исходные тексты, исполняемые модули пр.) и схемы развертывания системы
Тестирование	тестирование	Используется для проверки соответствия полученной информационной системы исходным требованиям

Данные модели описывают проектируемую систему на разных уровнях абстракции и с разными представлениями. При этом одни и те же элементы могут вместе применяться в разных моделях имеющих разную степень детализации (рис. 8).

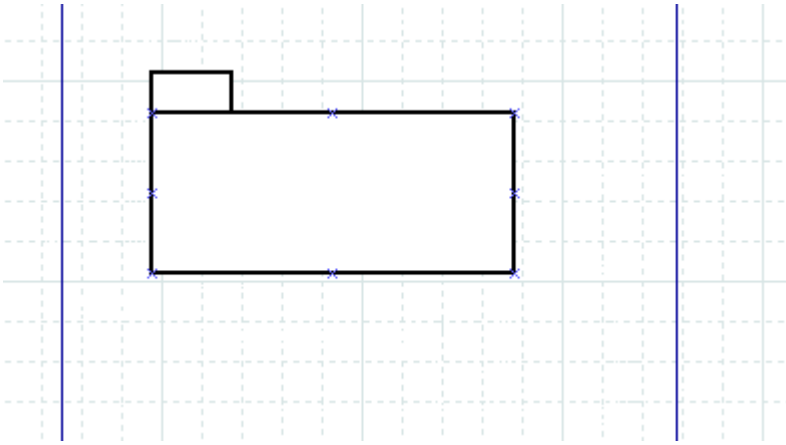


рисунок 8 - графическое обозначение модели(Dia - редактор схем и диаграмм).

Унифицированные модели могут с лёгкостью входить друг в друга, на картинке ниже отображена такая схема двумя способами (рис. 9).



рисунок 9 - унифицированные модели(Dia - редактор схем и диаграмм).

Все сущности, получаемые в унифицированном процессе можно изобразить в виде диаграмм артефактов, по аналогии с технологическим процессом. На (рис. 10) ниже изображена такая схема.

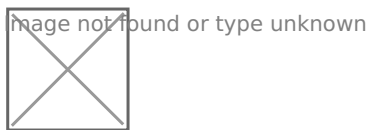


рисунок 10 - унифицированные модели.

Специальные средства автоматизации

Последовательная, аккуратная и быстрая реализация проекта невозможна без использования сторонних утилит – специальных средств автоматизации используемых исполнителями. К таким утилитам относятся технологические средства, или специализированный СОФТ[5], предназначенный для разработки, обеспечения, поддержки и документирования жизненного цикла приложения. Для примера рассмотрим продукцию компании International Business Machines Rational, которая является хорошим примером объектно-ориентированного стиля применяемого при разработке и моделировании информационных систем:

- International Business Machines Rational Rose [6]– визуальное моделирование и генерация объектного кода;

- International Business Machines Rational Team Test – авто-тестирование.
- International Business Machines Rational ClearCase [\[7\]](#)– конфигурационное управление;
- International Business Machines Rational Rapid Developer [\[8\]](#)– разработка кода;
- International Business Machines Rational RequisitePro [\[9\]](#) – управление требованиями;
- International Business Machines Rational ClearQuest [\[10\]](#) – управление и сохранение изменений;
- International Business Machines Rational SoDA [\[11\]](#) – авто-документирование;

IBM Rational Rose

IBM Rational Rose - это классическая Case-система являющаяся основным продуктом автоматизации при проектирования и разработки информационных систем.

Продукт применяют не только разработчики, IBM Rational Rose активно используют проектировщики и аналитики, при чем его можно синхронизовать с другими утилитами проектирования. IBM Rational Rose имеет ряд важных инструментов:

- проектирование с использованием модели в виде UML диаграмм;
- генерация кода – авто-генерация кода будущей программы на основе спроектированной модели;
- генерация скриптов DDL3, схем баз данных для Oracle и разметки XML;
- реинжиниринг (перепроектирование) – построение модели на основе скрипта программы, это обратный процесс от генерации кода;
- синхронизация модели с ее физическим объектом для осуществления удобного проектирования системы.

Dia Diagram Editor

DIE DIAGRAM EDITOR - редактор диаграмм, который умеет рисовать диаграммы: статических структур UML, баз данных, диаграмм сущность-связь, радиоэлектронных элементов, потоковых диаграмм, сетевых диаграмм и других. Продукт полностью переведен на русский язык.

Возможности:

- Поддержка диаграмм потоков, структурных диаграмм и т. д.;
- Экспорт в Postscript[\[12\]](#);

- Загрузка и сохранение в формате XML[13];
- Возможность описания новых объектов.

Хочу отметить, что в данном проекте я буду по большей части использовать данный редактор так как он распространяется по бесплатной модели.

Структура UML

Унифицированный язык моделирования UML можно назвать неофициальным стандартом документирования информационных систем проектирования.

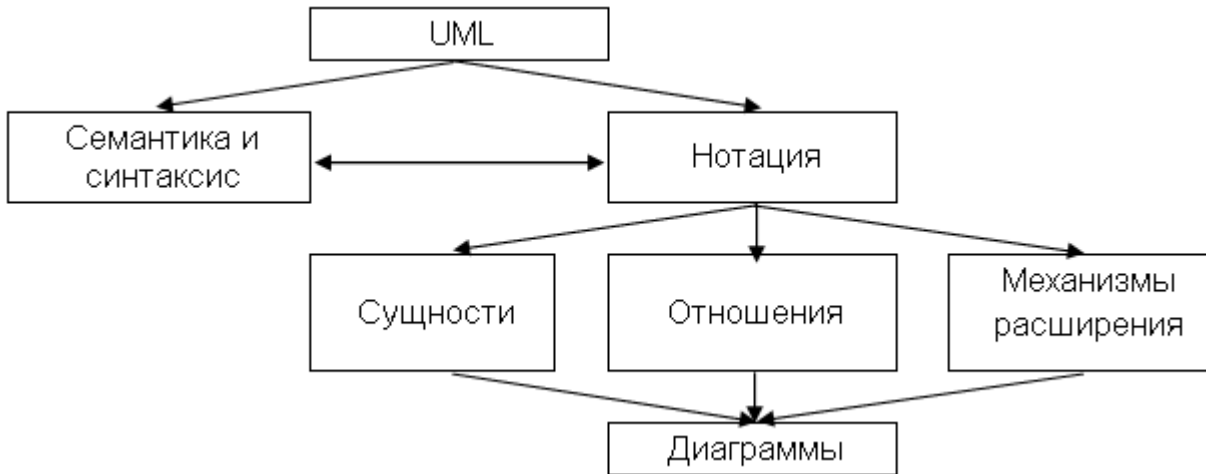


рисунок 10 - основные сегменты UML

Основные сегменты UML

Семантика – раздел языка, изучающий значение лингвистических единиц из слов и словосочетаний.

Синтаксис – способы сочетания слов и предложений.

Нотация – это визуальная интерпретация семантики и синтаксиса для построения графического представления для быстрого восприятия.

В контексте UML синтаксис и семантика формируют шаблон построения моделей, соединяющий обыденные и естественные языки для визуализации элементов модели и способов их расширения.

Сущности и отношения в UML

- структурный – абстракция обобщающая физический или визуальный объект

- группирующий – смысловое объединения диаграммы объектов
- поясняющий – документирование элементов диаграмм

На (рис. 11) ниже показаны все виды отношений, используемые в UML с их описанием в графическом виде.

Image not found or type unknown



рисунок 11 - отношения UML

Обозначения в UML

Таблица 2

Название	Обозначение	Описание
Стереотип	« »	Уточняет семантику элемента.
Сторожевое условие	[]	Логическое условие.
Ограничение	{ }	Ограничивает семантику элемента модели.
Помеченное значение	{ }	Новое или уточняющее свойство элемента.

Первые три типа характеризуют отношение сущностей, которая присуща общему количеству экземпляров объекта, находящихся в отношении. Отношение указывается с обеих сторон выбранной сущности. Отношение обычно указываться одним из пяти способов:

1. любое количество экземпляров;
2. любое неотрицательное целое число, кратность которого строго фиксирована;
3. диапазон неотрицательных целых чисел;
4. диапазон чисел с «открытым» конечным значением;
5. •перечисление целых неотрицательных диапазонов через запятую.

На (рис. 12) ниже наглядно продемонстрировано описание механизмов расширения, используемых для уточнения семантики отношений текстовой строки.

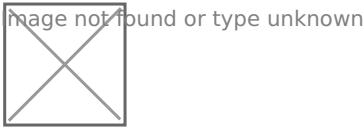


рисунок 12 - механизмы расширения UML

Кроме разметки, которая представляет собой текстовую строку в кавычках, на диаграммах можно применить графическую разметку.

Диаграммы UML

Диаграмма – это группировка элементов, используемая для отображения значений проектируемой информационной системы проектирования. Они представлены в виде графического рисунка, вершина которого отображается сущностями, а ребра отображаются отношениями (рис. 13) .

При проектировании конкретной модели в унифицированном процессе часто используют несколько типичных диаграмм, и, чем более сложная система, тем больше одинаковых диаграмм будет использовано при ее моделировании. Проектирование с помощью UML позволяет создавать диаграммы разных видов, но использовать их всех исполнитель не обязан.

Модель	Диаграмма							
	Вариантов использования	Состояний	Классов	Кооперации	Последовательности	Деятельности	Компонентов	Развертывания
Вариантов использования	+	+	+	+	+			
Анализа	+	+	+	+	+			
Проектирования		+	+	+	+	+		
Реализации			+				+	+

рисунок 13 - диаграмм UML

После начального этапа проектирования для некоторых диаграмм может потребоваться модификация и дополнительная реализация в рамках технологического процесса.

Синтаксис языка[14], порядок построения и информативность моделей UML довольно просты, предоставляя исполнителям широкие возможности для точного описания предметной области объекта. За лёгкость в освоение UML используют в различных аспектах проектирования, не ограничиваясь лишь созданием информационных систем, их применяют в различных архитектурных проектах и прочих программных комплексах.

Использование унифицированного языка моделирования на примере курьерской службы

Рассмотрим использование UML для проектирования автоматизированной системы на примере службы доставки.

Основным бизнес-процессом службы доставки является транспортировка посылок и грузов.

Цель информационной системы

Цель информационной системы является оптимизация работы отдела логистики и получение платы с клиента.

1. Менеджер отдела логистики ежедневно оформляет заявки клиентов на доставку посылок согласно прайс-листу компании. Он регистрирует заявки в логистическом листе.
2. После оформления и подтверждения заявки менеджер отдела логистики выставляет счёт клиенту на оплату и регистрирует его в журнале заявок.
3. Бухгалтер получает и регистрирует выписки банка и приходные кассовые ордера, содержащие информацию о поступлениях денежных средств в кассу предприятия или на расчётный счёт логистической фирмы. Бухгалтер, получив нужные документы, делает пометки оплаты счета в реестре счетов.
4. Менеджер отдела логистики контролирует поступление платежных счетов от клиента. По истечению срока оплаты менеджер помечает недействительные заявки в логистическом листе.
5. Менеджер отправляет счёта о доставке курьеру.
6. Курьер выписывает товарную накладную на оплаченную доставку на основании счета, после этого проверяет груз на сохранность, заполняет накладную на доставку и выдает посылку клиенту.
7. По истечении месяца на основании журнала заявок менеджером отдела логистики формируется сводка о количестве пришедших заявок.

Все основные модели данного бизнес-процесса наглядно проиллюстрированы на (рис. 14) .



рисунок 14 - бизнес процесс курьерской службы

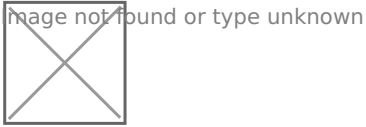


рисунок 15 - диаграмма действий “Курьерской службы”

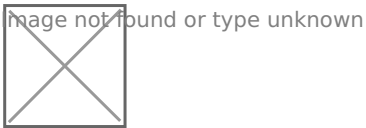


рисунок 16 - диаграмма композиций “Курьерской службы”

На рисунке ниже я попытаюсь воссоздать проектирование информационной системы службы доставки, используя веб приложение (рис. 17)

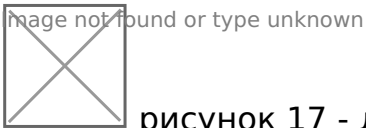


рисунок 17 - диаграмма действий “Курьерской службы”

Объектно-ориентированное проектирование системы на языке UML

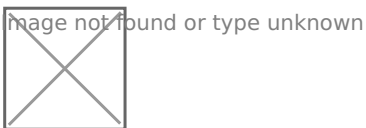


рисунок 17 - диаграмма вариантов использования

Описание выполнения процедуры учета заказов

Таблица 3

Прецедент Учет заказов

Актеры Клиента, информационная система

Цель	Регистрация и направление заказа на доставку
Краткое описание	Покупатель регистрируется в системе, выбирает город отправки и получения, оформляет заказ, выбирает скидку, оплачивает, информационная система регистрирует заказ и составляет счет-фактуру заказа, направляет ее курьеру, для последующей доставки
Тип	Базовый
Ссылки	«Оформление заказа», «передача заказа в курьеру»

Описание выполнения процедуры учета заказов клиента

Таблица 4

Действия актеров	Отклик
Покупатель выбирает город доставки и отправления и просчитывает доставку	Информационная система добавляет город отправки и получения в ЛК Клиента
Покупатель оформляет заказ	Информационная система регистрирует номер заказа, дату и др.реквизиты, подает сигнал о заявке курьеру
Покупатель вводит номер посылки и приложение проверяет номер и видит процент скидки для себя	предоставляет скидку
Покупатель оплачивает доставку	Информационная система проверяет платеж и отправляет email с подтверждением платежа покупателю

Информационная система оформляет счет-фактуру заказа и отправляет ее курьеру

Информационная система регистрирует документ и направляет его курьеру для последующей доставки покупателю



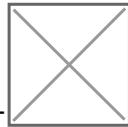
Описание процедуры оформления заказа

Таблица 5

Прецедент Оформление заказа

Актеры

Клиент



Цель

Составление заказа и оплата стоимости доставки

Краткое описание

Клиент оформляет доставку указав адрес, возможно ему предоставляется скидка, далее он оплачивает стоимость доставки

Тип

Включающий

Ссылки

«регистрация покупателя», «формирование заказа», «оплата доставки», «скидка»

Описание выполнения процедуры действия пользователей

Таблица 6

Действия актеров

Отклик

Покупатель выбирает, адреса и отмечает город доставки и город отправки	Информационная система добавляет код доставки к заказу покупателя;
Клиент оформляет заказ	Информационная система регистрирует номер заказа, дату и др.реквизиты, подает сигнал о заявке на склад или сразу курьеру;
Клиент вводит номер карты скидок службы доставки	Информационная система выдает номер и предоставляет скидку;
Клиент оплачивает заказ	Информационная система проверяет платеж и отправляет email письмо с подтверждением платежа покупателю;

Описание выполнения процедуры передачи заказа курьеру

Таблица 7

Прецедент	курьеру
Актеры	Информационная система
Цель	Контроль оплаты
Краткое описание	После оплаты заказа информационная система составляет счет-фактуру зарегистрированного заказа и направляет её курьеру
Тип	Включающий
Ссылки	-

Описание выполнения действий передачи заказа курьеру

Таблица 8

Действия актеров

Отклик

Информационная система оформляет счет-фактуру заказа и отправляет ее курьеру	Информационная система заполняет документ и направляет курьеру для последующей доставки клиенту
--	---

Разновидности UML диаграмм на примере курьерской службы

На рисунках ниже можно увидеть объектно-ориентированное проектирование системы в виде диаграммы UML и их разновидностей на примере курьерской службы:

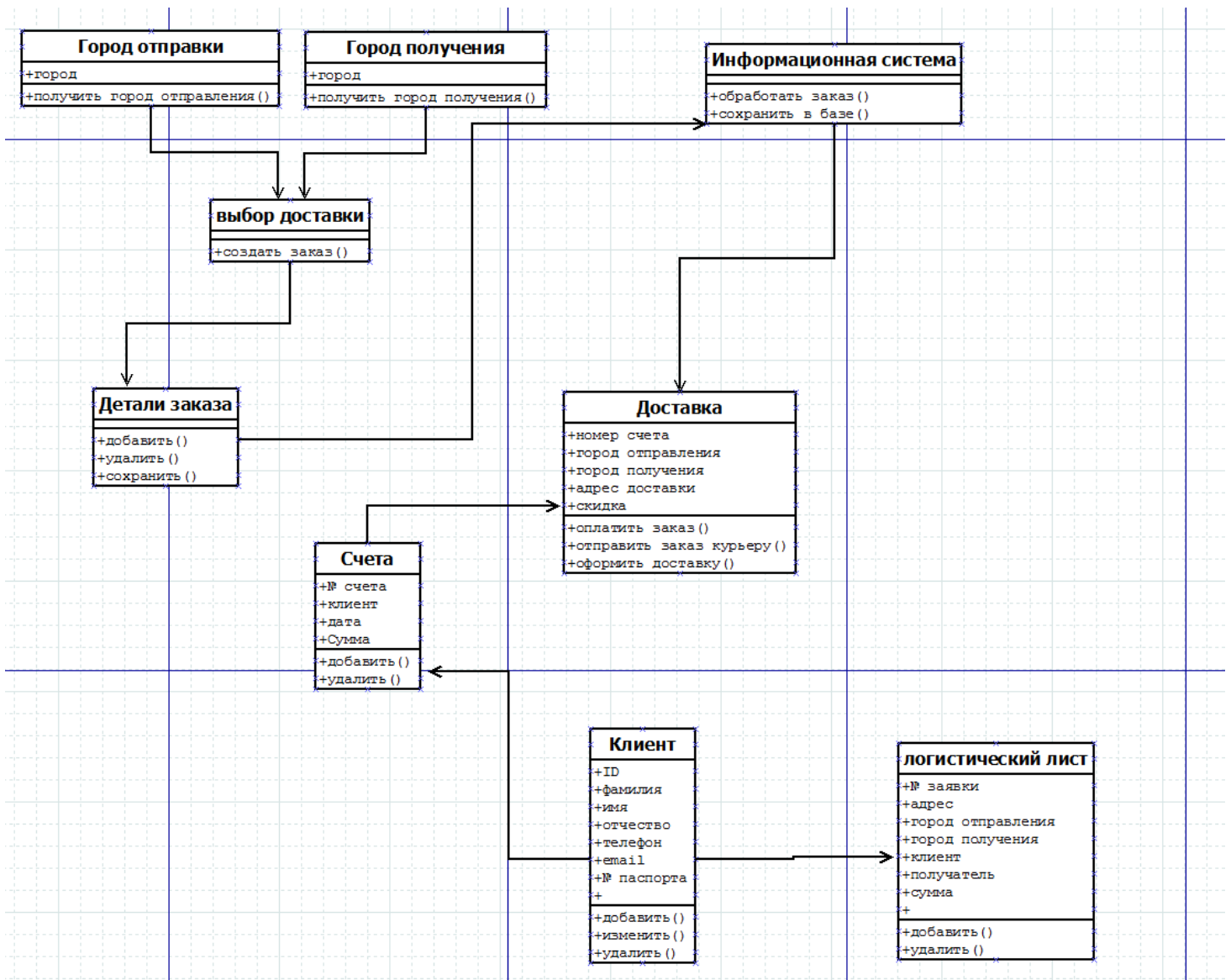


рисунок 18 - общая диаграмма классов

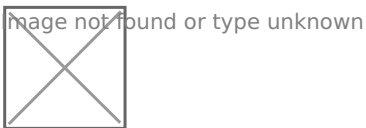


рисунок 19 - диаграмма последовательности

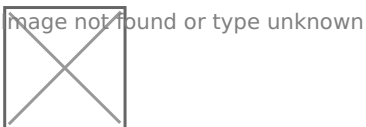


рисунок 20 - диаграмма кооперации

image not found or type unknown



рисунок 21 - диаграмма состояний

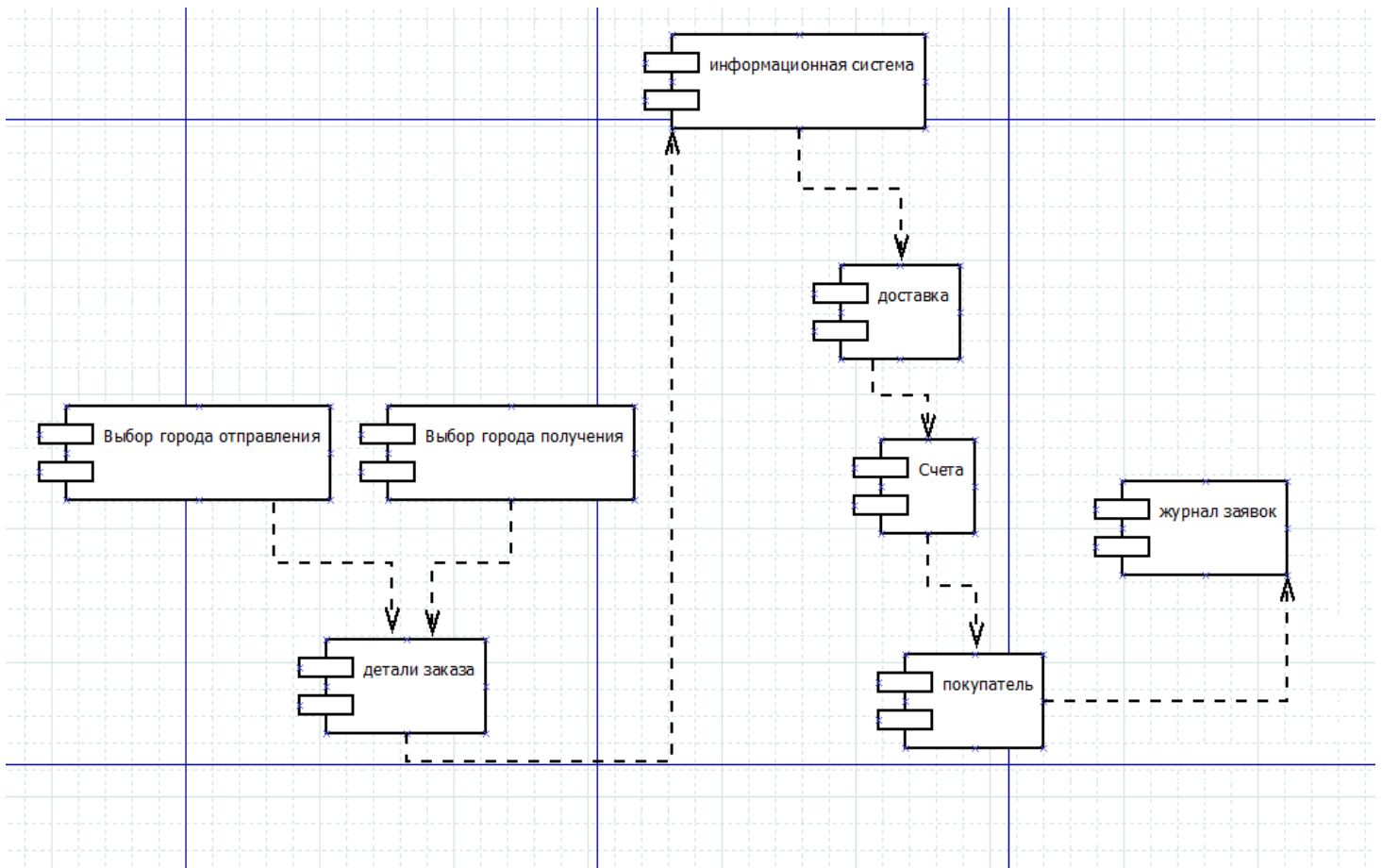


рисунок 22 - диаграмма компонентов

image not found or type unknown



рисунок 23 - диаграмма развертывания

Заключение

В процессе написания работы были изучены основы проектирования информационных систем, с применением объектно-ориентированного стиля используя унифицированный язык моделирования. В свете вышеизложенного можно сделать заключение, в котором необходимо описать как основные плюсы, так и несколько характерных для объектно-ориентированного подхода минусов:

Плюсы объектно-ориентированной концепции проектирования:

- Объектно-ориентированный подход дает возможность генерировать модели за счет использования встроенных механизмов, это позволяет из визуальной части получить работающий прототип системы.
- Восприятие архитектуры построенной системы легко переносится в физическую среду и наоборот
- Case-средства, поддерживающие ООП позволяют достичь большей степени автоматизации кода-генерации, что опять же, упрощает разработку ИС.
- Легкое восприятие моделей использующих параллельные вычисления.
- Принцип инкапсуляция защищает самые критичные данные от несанкционированного доступа.
- Реализация родительских и дочерних компонентов могут работать с одним видом объекта

Минусы объектно-ориентированной концепции проектирования:

- Понимание принципа ООП требует первичных начальных затрат на профессиональное проектирование.
- Реализация объектной модели наиболее успешно достигается к пониманию через повторное использование компонентов
- Диаграммы визуализирующие операции объектно-ориентированного подхода крайне трудно воспринимаются без первоначальных знаний.

Стоит отметить, что следующим этапом развития объектно-ориентированного подхода является агента-ориентированный подход (АОП), в котором основополагающими концепциями являются понятия агента и его ментальное поведение, зависящее от среды, в которой он находится. Однако на текущий момент эта концепция развита слабо.

Помимо объектно-ориентированного подхода существуют и другие менее популярные практики, как официально признанные, так и нет:

- Процедурный подход
- Аппликативный подход
- Доказательный подход
- Автоматный подход
- Порождающий подход
- Событийно-ориентированный подход
- Аспектно-ориентированный подход

- Компонентно-ориентированный подход
- Агентно-ориентированный подход

Изучаемый мной унифицированный язык моделирования позволяет реализовывать и другие технологии проектирования, но максимальную эффективность без лишних трудозатрат можно получить используя объектно-ориентированный стиль.

Плюсы UML:

- Объектно-ориентированный стиль, которым удобно можно пользоваться при проектировании информационной системы используя UML
- Возможность выстроить структуру будущего проекта учитывая все ты её разработки.
- Удобное моделирования бизнес-процессов с учетом всех тонкостей зации
- Лёгкая работа с использование визуальных UML диаграмм
- Возможность создавать свои графические интерфейсы и тем самым ирать функционал UML
- Обширная документация.
- Возможность использовать готовые паттерны проектирования

Минусы UML:

- Сложное использования системы при применении процедурного или событийного стиля проектирования
- Проблемы с совместимостью, используя разные языки программирования.

В подведении итогов хочу высказать свою точку зрения, что даже обладая некоторыми минусами, достойной альтернативы, на момент написания данной статьи при проектирование информационных систем языку моделирования UML - нет.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Ресурсы на русском языках

1. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения

2. ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания
3. ГОСТ Р ИСО/МЭК 12207-2010. Информационная технология. Системная и программная рия. Процессы жизненного цикла программных средств
4. ГОСТ Р ИСО/МЭК 15271-02. Процессы жизненного цикла программных средств.
5. Р. Баркер. CASE Method. Моделирование взаимосвязей между сущностями /Крюкова А.В, 33 с.
6. Б. Тимоти. Объектно-ориентированное программирование в действии / Питер, 1997. - 464
7. Г. Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на С++/ Бином, 2014. 560 с.
8. Г. Буч, Дж. Рамбо, А. Якобсон Язык UML. Руководство пользователя/ Питер, 2014. - 432 с.
9. А.М. Вендров. CASE-технологии. Современные методы и средства проектирования ационных систем / Финансы и статистика, 2015. 176 с.
10. М. Вайсфельд. Объектно-ориентированное мышление/ Мэтт Вайсфельд / Питер, 2014. -
11. К. Ларман. Применение UML и шаблонов проектирования: Уч. Пос / Издательский дом мс», 2014. 496 с.
12. И. Денис, Ф. Новиков. Моделирование на UML Учебно-методическое пособие / Наука и а, 2010. 640с.
13. В.И. Петров. Информационные системы / Питер, 2013. - 688 с.
14. Д. Розенберг. Применение объектного моделирования с использованием UML и анализ ентов/ ДМК-пресс, 2002. 159с.
15. А. Элиенс. Принципы объектно-ориентированной разработки программ / Издательский ильямс», 2014. 496 с.
16. А. Якобсон, Г. Буч, Дж. Рамбо. Унифицированный процесс разработки программного обеспечения / Питер, 2014. 496 с.
17. Г. Хассан. Проектирование систем реального времени, параллельных и распределенных приложений. / ДМК Пресс, 2016. 700с.
18. Хританков А.С, Полежаев В. А, Андрианов А. И. Проектирование на UML. Сборник задач / Литагент Ридеро, 2017. 163с.

Ресурсы на иностранных языках

E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns / Addison Wesley, 416 с.

Официальный сайт с докумеетацией UML – URL: <https://www.uml.org/>

Документация по IBM Rational Rose - URL: <https://www.ibm.com/ru-ru>

Документация по Dia Diagram Editor - URL: <http://www.dia-installer.de/>

1. Бизнес-модель — исследование того, какие издержки и расходы у вас есть, и сколько денег вы можете брать за продукт или услугу; URL: <https://kontur.ru/articles/5030> [↑](#)
2. Unified Modeling Language — унифицированный язык моделирования. URL: <https://www.uml.org/> [↑](#)
3. Мониторинг — система постоянного наблюдения за явлениями и процессами; URL: <https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%BD%D0%B8%D1%82%D0%BE%D1%82> [↑](#)
4. Фреймворк - программная платформа, определяющая структуру программной системы; URL: <https://ru.wikipedia.org/wiki/%D0%A4%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%B5> [↑](#)
5. Софт - комп. жарг., собир. программное или математическое обеспечение; URL: <https://ru.wiktionary.org/wiki/%D1%81%D0%BE%D1%84%D1%82> [↑](#)
6. официальная страница International Business Machines Rational Rose в интернете; URL: <https://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-ifix001> [↑](#)
7. официальная страница International Business Machines Rational ClearCase в интернете; URL: <https://www.ibm.com/us-en/marketplace/rational-clearcase> [↑](#)
8. официальная страница International Business Machines Rational Rapid Developer в интернете; URL: <https://www.ibm.com/us-en/marketplace/rad-for-websphere-software> [↑](#)

9. официальная страница International Business Machines Rational RequisitePro в интернете; URL: <https://www.ibm.com/support/pages/rational-requisitepro-714> ↑
10. официальная страница International Business Machines Rational ClearQuest в интернете; URL: <https://www.ibm.com/nl-en/marketplace/rational-clearquest> ↑
11. официальная страница International Business Machines Rational SoDa в интернете; URL: <https://www.ibm.com/support/pages/rational-soda-reporting-solution-rational-requisitepro-reaches-end-marketing> ↑
12. Документация ADOBE относительно POSTSCRIPT; URL: <https://helpx.adobe.com/ru/indesign/using/creating-postscript-eps-files.html> ↑
13. XML - расширяемый язык разметки; URL: <https://ru.wikipedia.org/wiki/XML> ↑
14. Синтаксис — сторона языка программирования, которая описывает структуру программ как наборов символов (обычно говорят — безотносительно к содержанию); URL: <https://dic.academic.ru/dic.nsf/ruwiki/357868> ↑