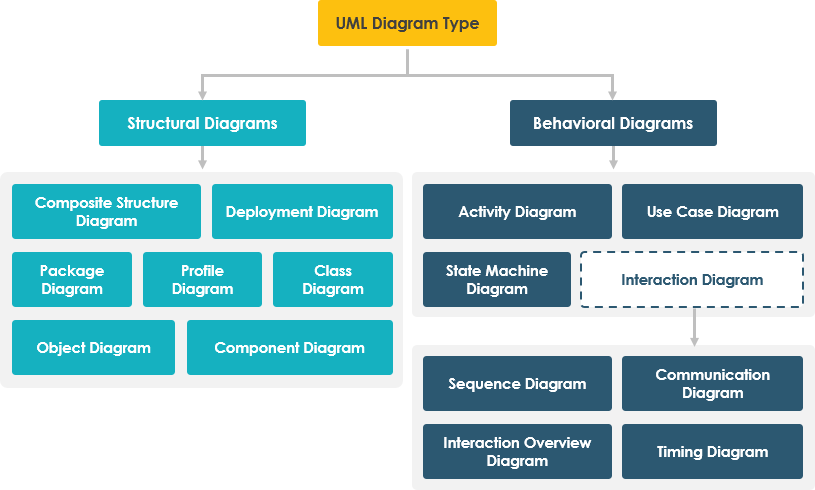
Дoклад 02

По Софтуерно инженерство

Изготвил: Валентин Кьосев 12 ,,а‘‘ клас

1. Видове UML диаграми – избройте и опишете за какво служат различните видове structural и behavioral диаграми. Дайте примери.



1. **Видове UML диаграми:**
2. **Какво е Unified Modeling Language (UML)?**

UML , съкращение от Unified Modeling Language, е стандартизиран език за моделиране, състоящ се от интегриран набор от диаграми, разработени да помогнат на разработчиците на системи и софтуер за специфициране, визуализиране, конструиране и документиране на артефактите на софтуерни системи, както и за бизнес моделиране и други несофтуерни системи.

UML представлява колекция от най-добри инженерни практики, които са се доказали като успешни при моделирането на големи и сложни системи.

UML е много важна част от разработването на обектно-ориентиран софтуер и процеса на разработка на софтуер.

UML използва предимно графични обозначения, за да изрази дизайна на софтуерните проекти. Използването на UML помага на проектните екипи да комуникират, да изследват потенциални дизайни и да валидират архитектурния дизайн на софтуера.

1. **Цел**

Целта на UML е да предостави стандартна нотация, която може да се използва от всички обектно-ориентирани методи и да избере и интегрира най-добрите елементи от предшестващите нотации.

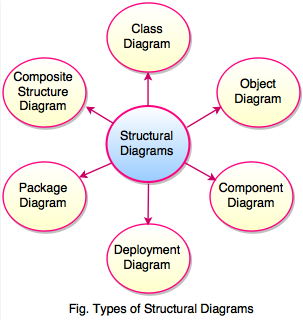
UML е проектиран за широк спектър от приложения. Следователно, той предоставя конструкции за широк спектър от системи и дейности (напр. разпределени системи, анализ, системно проектиране и внедряване).

1. **Structural diagrams / Структурни диаграми**

Структурните диаграми показват статичната структура на системата и нейните части на различни нива на абстракция и изпълнение и как те са свързани помежду си. Елементите в структурна диаграма представляват смислените концепции на система и могат да включват абстрактни, реални концепции и концепции за изпълнение.

* Class Diagram
* Component Diagram
* Deployment Diagram
* Object Diagram
* Package Diagram
* Composite Structure Diagram
* Profile Diagram

1. **Видове структурни UML диаграми**

****

**Class Diagram / Диаграма на класа**

Диаграмите на класове са основните градивни елементи на всеки обектно-ориентиран метод.

Диаграмата на класа може да се използва за показване на класовете, връзките, интерфейса, асоциирането и сътрудничеството. UML е стандартизиран в класови диаграми.

Тъй като класовете са градивният елемент на приложение, базирано на ООП, класовата диаграма има подходяща структура за представяне на класовете, наследството, връзките и всичко, което ООП имат в своя контекст.

**Package Diagram / Диаграма на опаковката**

Пакетната диаграма е вид диаграма на Unified Modeling Language (UML), използвана главно за представяне на организацията и структурата на система под формата на пакети.

Пакетът се използва като контейнер за организиране на присъстващите в системата елементи в по-управляема единица.

Много е полезно да се представи архитектурата и дизайна на системата като сплотена единица и по кратък начин.

**Object Diagram / Диаграма на обекта**

Обектна диаграма може да се нарече екранна снимка на екземплярите в системата и връзката, която съществува между тях.

Тъй като диаграмите на обекти изобразяват поведение, когато обектите са инстанцирани, ние сме в състояние да изучаваме поведението на системата в определен момент.

Обектните диаграми са жизненоважни за представяне и разбиране на функционалните изисквания на една система. С други думи, „Диаграма на обект в Unified Modeling Language (UML) е диаграма, която показва пълен или частичен изглед на структурата на моделирана система в определен момент.

**Component Diagram / Диаграма на компонентите**

Диаграмите на компонентите се използват, за да представят как са организирани физическите компоненти в една система. Ние ги използваме за моделиране на подробности за изпълнението.

Диаграмите на компонентите изобразяват структурната връзка между елементите на софтуерната система и ни помагат да разберем дали функционалните изисквания са били покрити от планираното развитие.

Диаграмите на компонентите стават съществени за използване, когато проектираме и изграждаме сложни системи.

Интерфейсите се използват от компонентите на системата за комуникация помежду си.

**Composite structure diagram / Композитна структурна диаграма**

Композитната структурна диаграма представлява връзката между части и тяхната конфигурация, които определят как се държи класификаторът (клас, компонент или възел за разгръщане).

Те представляват вътрешна структура на структуриран класификатор, който използва части, портове и конектори.

Те са подобни на диаграмите на класове, с изключение на това, че представят отделни части в детайли в сравнение с целия клас.

**Deployment diagram / Схема на разполагане**

Диаграмите за разполагане се използват за представяне на системния хардуер и неговия софтуер. Те ни казват какви хардуерни компоненти съществуват и какви софтуерни компоненти работят върху тях.

Ние илюстрираме системната архитектура като разпределение на софтуерни артефакти върху разпределени цели.

Те се използват предимно, когато даден софтуер се използва, разпространява или внедрява на множество машини с различни конфигурации.

1. **Ползи от използването на структурни UML диаграми**

Структурните диаграми са като визуални ръководства, които помагат на всеки от софтуерния екип да види как различните части на програмата си пасват. Те правят планирането, изграждането и поправянето на нещата много по-лесно.

* **Ясна картина за всеки:**

Структурните UML диаграми са като визуални карти, които показват как са свързани различни части от компютърна програма или система.

* **Помага при проектирането и планирането:**

Тези диаграми са като чертежи за построяване на къща.

Те помагат на хората, които проектират софтуера, да видят как различните части ще работят заедно.

* **Лесно за обяснение и споделяне:**

Структурните диаграми улесняват обяснението как работи една компютърна програма и споделят идеи с екипа.

* **Поддържа нещата организирани:**

Тези диаграми помагат за организирането на различните части на компютърна програма.

Той поддържа всичко в ред, което го прави по-лесно за разбиране и актуализиране.

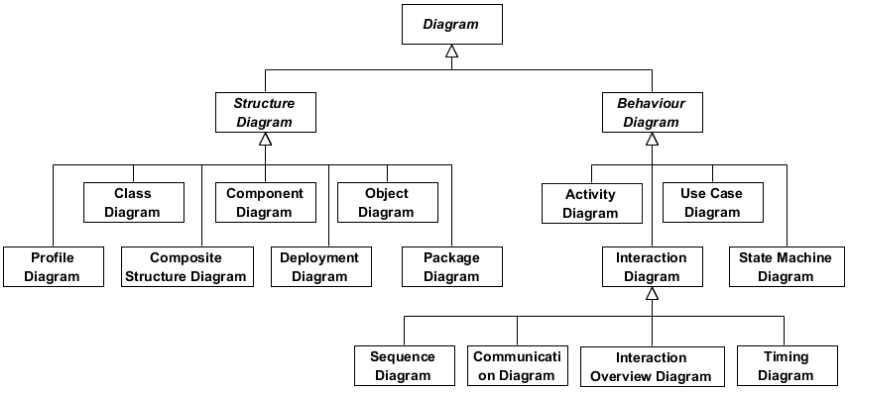
* **Спестява време и отстранява проблемите навреме:**

Преди да построят нещо голямо, като мост, инженерите използват чертежи, за да уловят проблемите навреме. Структурните диаграми правят същото за компютърните програми.

Те помагат да се намерят и коригират проблеми, преди да отделите много време за писане на действителния код.

1. **Behavioral diagrams / Диаграмите на поведение**

Диаграмите на поведение показват динамичното поведение на обектите в една система, което може да се опише като поредица от промени в системата във времето.



**Activity Diagrams / Диаграми на активността**

Диаграмите на дейност моделират поведението на системата и начина, по който тези поведения са свързани в цялостния поток на системата.

**Use Case Diagrams / Диаграми на случаи на използване**

Диаграмите на случаите на употреба улавят случаите на употреба и взаимоотношенията между участниците и системата; те описват функционалните изисквания на системата, начина, по който външните оператори взаимодействат на границата на системата и реакцията на системата.

**StateMachine Diagrams / Диаграми на StateMachine**

Диаграмите на StateMachine илюстрират как даден елемент може да се движи между състояния, класифицирайки поведението му според тригерите за преход и ограничаващите предпазители.

**Timing Diagrams / Времеви диаграми**

Времевите диаграми определят поведението на различни обекти в рамките на времева скала, осигурявайки визуално представяне на обекти, променящи състоянието си и взаимодействащи във времето.

**Sequence Diagrams / Диаграми на последователности**

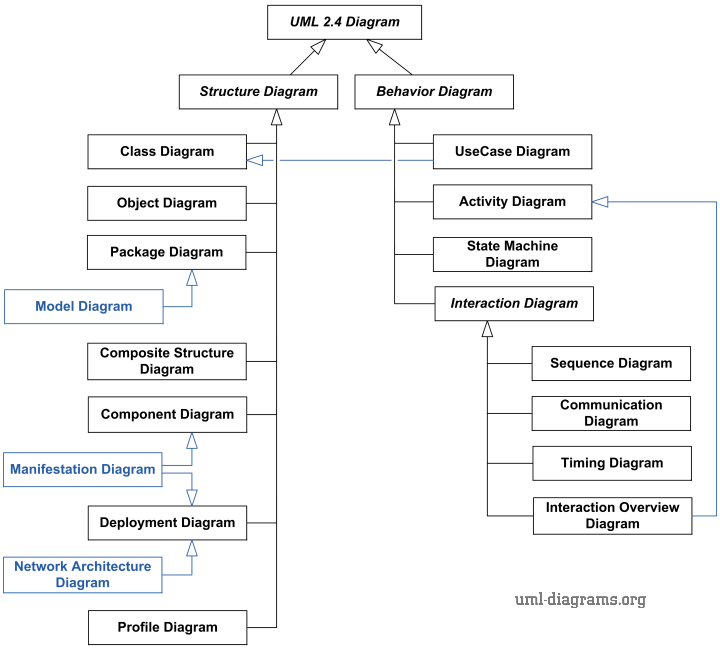
Диаграмите на последователността са структурирани представяния на поведението като поредица от последователни стъпки във времето. Те се използват за изобразяване на работния процес, предаването на съобщения и как елементите като цяло си сътрудничат с течение на времето, за да постигнат резултат.

**Communication Diagrams / Комуникационни диаграми**

Комуникационните диаграми показват взаимодействията между елементите по време на изпълнение, визуализирайки връзките между обектите.

**Interaction Overview Diagrams / Диаграми за преглед на взаимодействие**

Диаграмите за общ преглед на взаимодействието визуализират сътрудничеството между диаграми на взаимодействие (Диаграми за време, последователност, комуникация и други диаграми за преглед на взаимодействие), за да илюстрират контролен поток, обслужващ всеобхватна цел.



**Използвана литература / Източници:**

<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>

<https://www.geeksforgeeks.org/structural-diagrams-unified-modeling-languageuml/>

<https://sparxsystems.com/enterprise_architect_user_guide/16.1/modeling_languages/behavioraldiagrams.html>