

DDD glosario:

"Diseño dirigido por dominios: abordando la complejidad en el corazón del software por Evans, Eric.

En este color, los conceptos del taller de DDD Ollivander's

Glosario

Aquí se presentan definiciones breves de términos seleccionados, nombres de patrones y otros conceptos utilizados en el libro.

AGGREGATE (AGREGADO)

Un conjunto de objetos asociados que se tratan como una unidad con el propósito de cambios de datos. Las referencias externas están restringidas a un miembro del AGGREGATE, designado como la raíz. Un conjunto de reglas de consistencia se aplica dentro de los límites del AGGREGATE.

analysis pattern análisis de patrones

Un grupo de conceptos que representa una construcción común en la modelización empresarial. Puede ser relevante para un solo dominio o puede abarcar muchos dominios (Fowler 1997, p. 8).

ASSERTION AFIRMACIÓN

Una declaración del estado correcto de un programa en algún punto, independientemente de cómo lo haga. Típicamente, una AFIRMACIÓN especifica el resultado de una operación o un invariante de un elemento de diseño.

BOUNDED CONTEXT (CONTEXTO DELIMITADO)

La aplicabilidad delimitada de un modelo particular. Los BOUNDING CONTEXTS dan a los miembros del equipo una comprensión clara y compartida de lo que debe ser consistente y de lo que puede desarrollarse de manera independiente.

Client cliente

Un elemento de programa que llama al elemento en diseño, utilizando sus capacidades.

Cohesion cohesión

Acuerdo y dependencia lógica.

command comando (también conocido como modificador)

Una operación que efectúa algún cambio en el sistema (por ejemplo, estableciendo una variable). Una operación que crea intencionalmente un efecto secundario.

CONCEPTUAL CONTOUR CONTORNO CONCEPTUAL

Una consistencia subyacente del propio dominio, que, si se refleja en un modelo, puede ayudar al diseño a adaptarse al cambio de manera más natural.

Cohesion contexto

El entorno en el que aparece una palabra o afirmación que determina su significado.

Ver también [BOUNDED CONTEXT] (CONTEXTO DELIMITADO)

CONTEXT MAP MAPA DE CONTEXTO

Una representación de los BOUNDED CONTEXTS involucrados en un proyecto y las relaciones reales entre ellos y sus modelos.

CORE DOMAIN DOMINIO CENTRAL

La parte distintiva del modelo, central para los objetivos del usuario, que diferencia la aplicación y la hace valiosa.

declarative design diseño declarativo

Una forma de programación en la que una descripción precisa de las propiedades controla realmente el software. Una especificación ejecutable.

deep model modelo profundo

Una expresión incisiva de las preocupaciones principales de los expertos en el dominio y su conocimiento más relevante. Un modelo profundo desecha los aspectos superficiales del dominio y las interpretaciones ingenuas.

design pattern patrón de diseño

Una descripción de objetos y clases que se comunican y que están personalizados para resolver un problema de diseño general en un contexto particular. (Gamma et al. 1995, p. 3)

Distillation destilación

Un proceso de separación de los componentes de una mezcla para extraer la esencia en una forma que la hace más valiosa y útil. En el diseño de software, la abstracción de aspectos clave en un modelo, o la partición de un sistema más grande para llevar el DOMINIO CENTRAL al frente.

Domain dominio

Un ámbito de conocimiento, influencia o actividad.

domain expert experto en dominio

Un miembro de un proyecto de software cuyo campo es el dominio de la aplicación, en lugar del desarrollo de software. No solo cualquier usuario del software, el experto en dominio tiene un conocimiento profundo del tema.

domain layer capa de dominio

Aquella parte del diseño e implementación responsable de la lógica del dominio dentro de una ARQUITECTURA EN CAPAS. La capa de dominio es donde vive la expresión de software del modelo de dominio.

ENTITY ENTIDAD

Un objeto definido fundamentalmente no por sus atributos, sino por un hilo de continuidad e identidad.

FACTORY FÁBRICA

Un mecanismo para encapsular lógica de creación compleja y abstraer el tipo de un objeto creado por el bien de un cliente.

Function función

Una operación que calcula y devuelve un resultado sin efectos secundarios observables.

Immutable inmutable

La propiedad de nunca cambiar el estado observable después de la creación."

implicit concept
concepto implícito

Un concepto que es necesario para entender el significado de un modelo o diseño pero que nunca se menciona.

INTENTION-REVEALING INTERFACE

INTERFAZ REVELADORA DE INTENCIÓN

Un diseño en el que los nombres de clases, métodos y otros elementos transmiten tanto el propósito original del desarrollador al crearlos como su valor para un desarrollador cliente.

Invariant invariante

Una AFIRMACIÓN sobre algún elemento de diseño que debe ser verdadera en todo momento, excepto durante situaciones específicamente transitorias como en medio de la ejecución de un método o en medio de una transacción de base de datos no comprometida.

Iteration iteración

Un proceso en el que un programa se mejora repetidamente en pequeños pasos. También, uno de esos pasos.

large-scale structure estructura a gran escala

Un conjunto de conceptos de alto nivel, reglas o ambos que establecen un patrón de diseño para todo un sistema. Un lenguaje que permite discutir y entender el sistema en términos generales.

LAYERED ARCHITECTURE ARQUITECTURA EN CAPAS

Una técnica para separar las preocupaciones de un sistema de software, aislando una capa de dominio, entre otras cosas.

life cycle ciclo de vida

Una secuencia de estados que un objeto puede adoptar entre la creación y la eliminación, típicamente con restricciones para garantizar la integridad al cambiar de un estado a otro. Puede incluir la migración de una ENTIDAD entre sistemas y diferentes CONTEXTOS DELIMITADOS.

Model modelo

Un sistema de abstracciones que describe aspectos seleccionados de un dominio y puede utilizarse para resolver problemas relacionados con ese dominio.

MODEL-DRIVEN DESIGN DISEÑO DIRIGIDO POR MODELOS

Un diseño en el que algún subconjunto de elementos de software se corresponde estrechamente con elementos de un modelo. También, un proceso de co-desarrollo de un modelo y una implementación que se mantienen alineados entre sí.

modeling paradigm paradigma de modelado

Un estilo particular de definir conceptos en un dominio, combinado con herramientas para crear análogos de software de esos conceptos (por ejemplo, programación orientada a objetos y programación lógica).

REPOSITORY REPOSITORIO

Un mecanismo para encapsular el almacenamiento, la recuperación y el comportamiento de búsqueda que emula una colección de objetos.

Responsibility responsabilidad

Una obligación de realizar una tarea o conocer información (Wirfs-Brock et al. 2003, p. 3).

SERVICE SERVICIO

Una operación ofrecida como una interfaz que se mantiene por sí sola en el modelo, sin estado encapsulado.

side effect efecto secundario

Cualquier cambio observable de estado que resulte de una operación, ya sea intencional o no, incluso una actualización deliberada.

FUNCIÓN LIBRE DE EFECTOS SECUNDARIOS

Ver [función]

STANDALONE CLASS CLASE INDEPENDIENTE

Una clase que puede entenderse y probarse sin referencia a otras, excepto a primitivas del sistema y bibliotecas básicas.

stateless sin estado

La propiedad de un elemento de diseño que permite a un cliente usar cualquiera de sus operaciones sin tener en cuenta el historial del elemento. Un elemento sin estado puede usar información accesible globalmente e incluso cambiar esa información global (es decir, puede

tener efectos secundarios) pero no tiene ningún estado privado que afecte su comportamiento.

strategic design diseño estratégico

Decisiones de modelado y diseño que se aplican a grandes partes del sistema. Tales decisiones afectan a todo el proyecto y deben decidirse a nivel de equipo.

supple design diseño flexible

Un diseño que pone el poder inherente en un modelo profundo en manos de un desarrollador cliente para hacer expresiones claras y flexibles que den resultados esperados de manera robusta. Igualmente, importante, aprovecha ese mismo modelo profundo para hacer que el diseño en sí mismo sea fácil para el implementador de moldear y remodelar para acomodar nuevas ideas.

UBIQUITOUS LANGUAGE LENGUAJE UBICUO

Un lenguaje estructurado en torno al modelo de dominio y utilizado por todos los miembros del equipo para conectar todas las actividades del equipo con el software.

Unification unificación

La consistencia interna de un modelo de tal manera que cada término es inequívoco y ninguna regla contradice.

VALUE OBJECT OBJETO DE VALOR

Un objeto que describe alguna característica o atributo pero no tiene ningún concepto de identidad.

WHOLE VALUE VALOR COMPLETO

Un objeto que modela un concepto único y completo.