



UNIVERSIDAD SIMÓN BOLÍVAR



Ingeniería de Software I (CI3715)

Patrones de Diseño de Software

(Clase XIV)

Profesor:

Jean Carlos Guzmán

Mail:

soportepregradousb2014@gmail.com

by @JeanCGuzman



Sartenejas, Marzo 2017

Agenda I

- Motivación
- Definiciones de Patrón
- Clasificación de Patrones
 - Estilo o Patrón Arquitectónico
 - Patrones de Diseño
 - Idiomas

Motivación

- Llegar a ser un maestro de ajedrez
 - Aprender reglas y requerimientos físicos
 - Nombres de piezas, movimientos válidos, geometría del tablero y orientación...
 - Aprender principios
 - Valor relativo de ciertas piezas, valor estratégico de posiciones centrales, poder de amenaza...
 - Estudiar los juegos de maestros
 - Entenderlos, memorizarlos y aplicarlos



Motivación

- **Llegar a ser un maestro de diseño**
 - Aprender reglas y requerimientos físicos
 - Algoritmos, estructuras de datos, lenguajes...
 - Aprender principios
 - Paradigma + principios del paradigma...
 - Estudiar los juegos de maestros
 - Entenderlos y aplicarlos



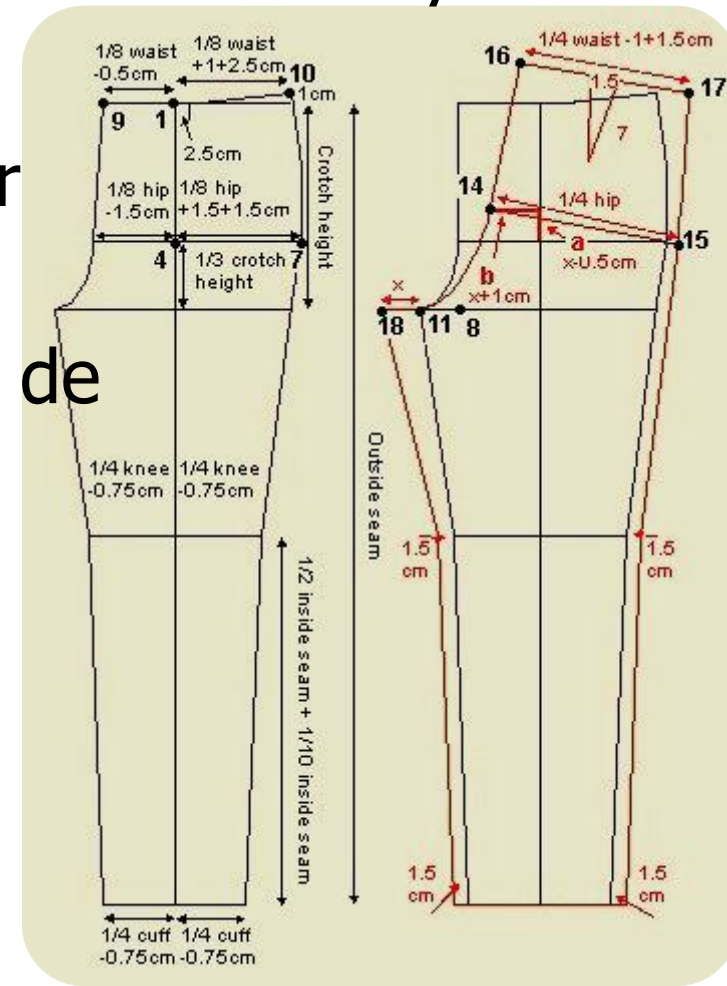
Motivación

- Una de las metas del diseño arquitectónico del software es el uso de patrones o soluciones reutilizables que reflejen la organización del sistema.
- Patrones de Diseño dan soporte a la reutilización de arquitecturas de software:
 - Presentan estructuras estáticas y dinámicas así como colaboraciones en soluciones exitosas a problemas en un dominio particular



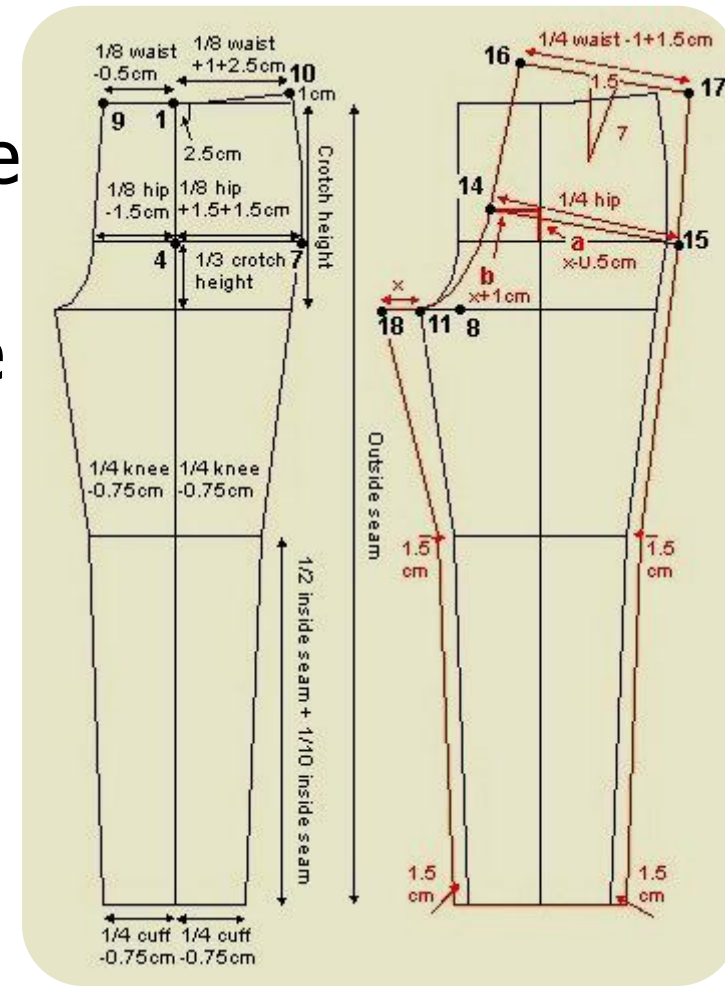
Patrón

- Enfoca un problema de diseño recurrente que surge en situaciones específicas de diseño y presenta una solución
- Define abstracciones de mayor nivel que el de las clases
- Suministra vocabulario común de diseño
- Documenta arquitecturas particulares y experiencias de diseño



Patrón

- Implementaciones diferentes de un patrón no necesariamente son iguales
- Proceso de selección e instanciación no automatizable (hasta ahora)
- Lineamientos que dan soporte



Clasificación de Patrones

- Para Gamma et al (1995) los patrones de diseño, se basa en características de uso y comportamiento:
 - Patrones de construcción (Ej. Abstract Factory)
 - Patrones estructurales (Ej. Facade, Proxy)
 - Patrones de comportamiento (Ej. Subject-Observer, Mediator, Factory Method)

Clasificación de Patrones

- Buschmann et al (1996), se basa en las relaciones de composición entre los elementos arquitectónicos:
 - Patrón arquitectónico: un conjunto de subsistemas (Ej. MVC, PAC, BROKER)
 - Patrón de diseño o subsistema (Ej. Subject-Observer, Mediator, Proxy)
 - Idioms de bajo nivel: para la implementación (Ej. Factory Method)

Clasificación de Patrones



Clasificación de Patrones

- **Estilo o Patrón Arquitectónico:**
 - El uso de estilos de diseño es común a varias disciplinas de la ingeniería.
 - Son el producto de una larga experiencia de diseñadores de sistemas.
 - Devienen del valor de algunos principios organizacionales específicos y sus estructuras, para una clase particular de software
 - Definen una familia de sistemas en términos de patrones de alto nivel de abstracción

Clasificación de Patrones

- **Estilo o Patrón Arquitectónico:**
 - El vocabulario utilizado es recopilado en manuales y material de uso profesional.

Clasificación de Patrones

- Shared Data (Datos Compartidos)
 - Repositorio
 - Pizarrón (Blackboard)
- Control Basado en Eventos
 - publisher/subscriber
 - Mediator

Clasificación de Patrones

- Arquitecturas de Sistemas Distribuidos
 - Cliente-Servidor
 - Objetos Distribuidos
 - Peer-To-Peer
 - Service Oriented Architecture (SOA)

Patrones de Diseño

- Reutilizar diseños de expertos
- Patrón describe una solución exitosa a un problema recurrente en un contexto
 - Documenta conocimiento experto de diseño,
 - Soluciones exitosas,
 - Guía de cuando y cómo usarlas
 - Consecuencias
 - Identifica clases, colaboraciones, responsabilidades

Patrones de Diseño

- No son diseños concretos para sistemas particulares
- Deben ser instanciados en un dominio particular:
 - tomar decisiones de diseño e implementación
- Involucra evaluar y considerar consecuencias
- Implementar el patrón y combinarlo con código que puede o no derivar de un patrón

Idioms

- Es un patrón de bajo nivel, específico para un lenguaje de programación.
- Describe como implementar aspectos particulares de elementos o de las relaciones entre ellos usando las características de un lenguaje particular:
 - Buschmann al et. (1996):
 - Singleton
 - Factory Method
 - Template Method
 - Counted Pointer
 - Ejemplos en C++:
 - Bloqueo de alcance
 - Interfaz de Hilos-Seguros
 - Constructor Virtual
 - Flujo de control indentado

Referencias

- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). Pattern – Oriented Software Architecture. A System of Patterns. John Wiley & Sons, England.
- Gamma E., Helm R., Johnson R. and Vlissides J. (1995). Design Patterns. Elements of Reusable Object-Oriented Software. Addison Wesley, Reading, Massachussets.
- Garlan D. and Shaw M. (1993). An Introduction to Software Architecture. To appear in Advances in Software Engineering and Knowledge Engineering. Vol. I, World Scientific Publishing Co.
- Garlan D. and Shaw M. (1996). Software Architecture: Perspectives on an Emerging Discipline. School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213-3890.