Reúso de software
Librerías y frameworks OO

Reúso

- ¿qué es lo que reusamos?
- ¿por qué el reúso es un tema importante en Ing. De Software?
- ¿qué dificultades encontramos?
- ¿qué alternativas de reúso tenemos?

¿qué reusamos?

- Conceptos e ideas que funcionan
  - P.e., Compartir en las redes sociales; ayuda de contexto…
- Diseños o estrategias de diseño (que imitamos, implementamos)
  - P.e., Patrones de diseño, algoritmos de búsqueda, arquitecturas…
- Funciones y estructuras de datos
  - P.e., de Colecciones, de Fechas, de Archivos,
- Componentes y Servicios (que invocamos)
  - P.e., Twitter API, OpenAI API, Google Maps, …
- Aplicaciones completas que adaptamos e integramos
  - P.e., Drupal/Wordpress; Mindcraft, ...

¿qué reusamos?

- Conceptos e ideas que funcionan
  - P.e., Compartir en las redes sociales; ayuda de contexto…
- Diseños o estrategias de diseño (que imitamos, implementamos)
  - P.e., Patrones de diseño, algoritmos de búsqueda, arquitecturas…
- Funciones y estructuras de datos
  - P.e., de Colecciones, de Fechas, de Archivos,
- Componentes y Servicios (que invocamos)
  - P.e., Twitter API, OpenAI API, Google Maps, …
- Aplicaciones completas que adaptamos e integramos
  - P.e., Drupal/Wordpress; Mindcraft, ...

¿qué reusamos?

- Conceptos e ideas que funcionan
  - P.e., Compartir en las redes sociales; ayuda de contexto…
- Diseños o estrategias de diseño (que imitamos, implementamos)
  - P.e., Patrones de diseño, algoritmos de búsqueda, arquitecturas…
- Funciones y estructuras de datos
  - P.e., de Colecciones, de Fechas, de Archivos,
- Componentes y Servicios (que invocamos)
  - P.e., Twitter API, OpenAI API, Google Maps, …
- Aplicaciones completas que adaptamos e integramos
  - P.e., Drupal/Wordpress; Mindcraft, ...

¿qué reusamos?

- Conceptos e ideas que funcionan
  - P.e., Compartir en las redes sociales; ayuda de contexto…
- Diseños o estrategias de diseño (que imitamos, implementamos)
  - P.e., Patrones de diseño, algoritmos de búsqueda, arquitecturas…
- Funciones y estructuras de datos
  - P.e., de Colecciones, de Fechas, de Archivos,
- Componentes y Servicios (que invocamos)
  - P.e., Twitter API, OpenAI API, Google Maps, …
- Aplicaciones completas que adaptamos e integramos
  - P.e., Drupal/Wordpress; Mindcraft, ...

¿qué reusamos?

- Conceptos e ideas que funcionan
  - P.e., Compartir en las redes sociales; ayuda de contexto…
- Diseños o estrategias de diseño (que imitamos, implementamos)
  - P.e., Patrones de diseño, algoritmos de búsqueda, arquitecturas…
- Funciones y estructuras de datos
  - P.e., de Colecciones, de Fechas, de Archivos,
- Componentes y Servicios (que invocamos)
  - P.e., Twitter API, OpenAI API, Google Maps, …
- Aplicaciones completas que adaptamos e integramos
  - P.e., Drupal/Wordpress; Mindcraft, ...

¿por qué?

- Para aumentar la productividad
  - Reducir los costos de desarrollo y mantenimiento
  - Reducir los tiempos de entrega y puesta en el mercado
  - Agilidad de mercado y customización en masa

- Para aumentar la calidad
  - El reúso aprovecha mejoras y correcciones – el tiempo da madurez y confiabilidad
  - Se reaprovecha el conocimiento de los especialistas (que queda encapsulado en componentes reusables)
  - Ayuda a la implementación e imposición de estándares

¿por qué?

- Para aumentar la productividad
  - Reducir los costos de desarrollo y mantenimiento
  - Menores tiempos de entrega y puesta en el mercado
  - Agilidad de mercado y customización en masa

- Para aumentar la calidad
  - El reúso aprovecha mejoras y correcciones – el tiempo da madurez y confiabilidad
  - Se reaprovecha el conocimiento de los especialistas (que queda encapsulado en componentes reusables)

¿qué dificultades tenemos?

- Problemas de mantenimiento si no tenemos control de los componente que reusamos
- Algunos programadores prefieren hacer todo ellos mismos (confianza en lo propio, búsqueda de desafíos)
- Hacer software reusable es mas difícil y costoso (paga a la larga y requiere procesos especiales)
- Encontrar, aprender a usar y adaptar componentes reusables requiere esfuerzo adicional

¿qué dificultades tenemos?

- Dificultades de mantenimiento si no tenemos control de los componente que reusamos
- Algunos programadores prefieren hacer todo ellos mismos (confianza en lo propio, búsqueda de desafíos)
- Hacer software reusable es mas difícil y costoso (paga a la larga y requiere procesos especiales)
- Encontrar, aprender a usar y adaptar componentes reusables requiere esfuerzo adicional

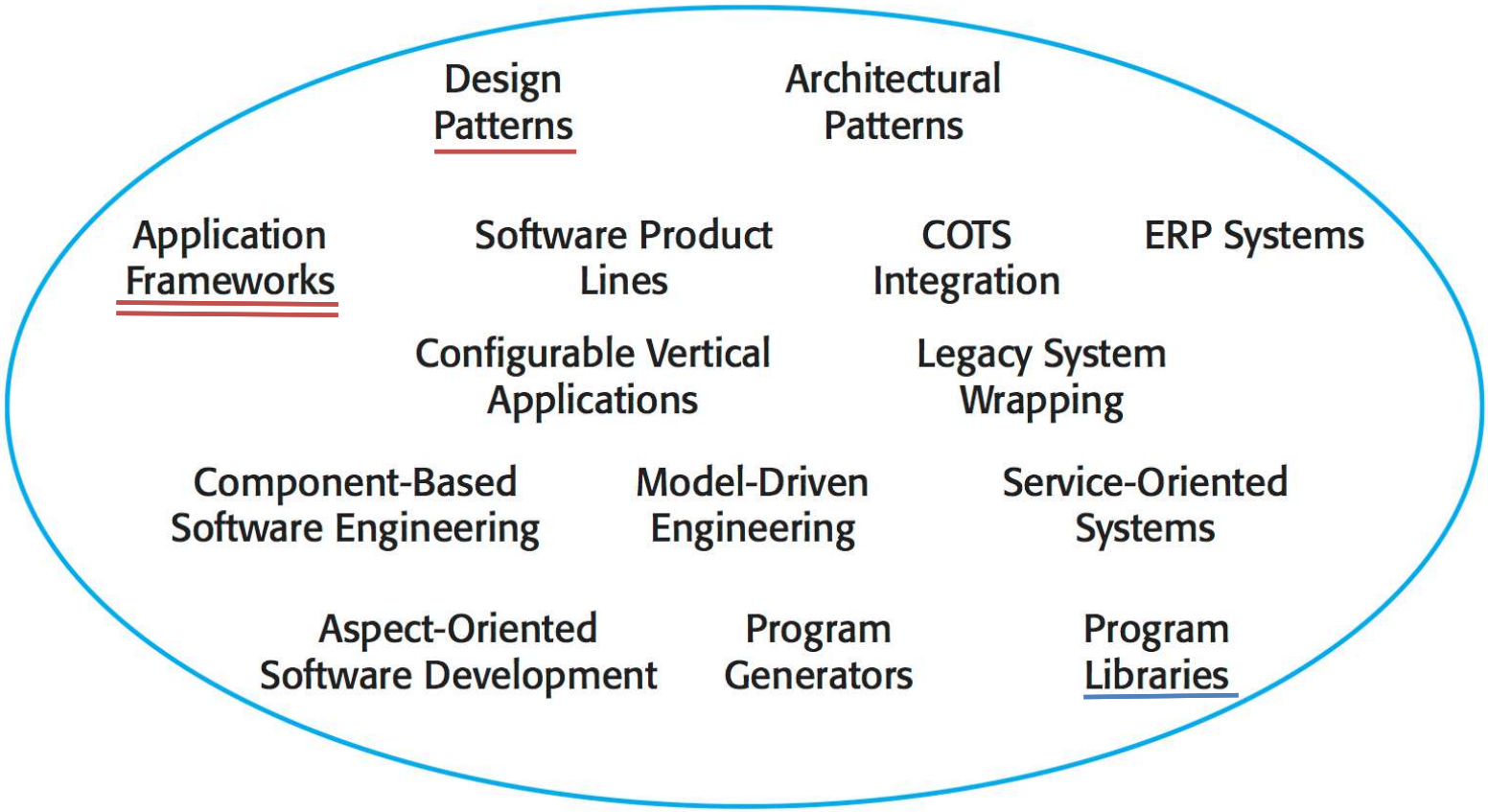
¿qué dificultades tenemos?

- Dificultades de mantenimiento si no tenemos control de los componente que reusamos
- Algunos programadores prefieren hacer todo ellos mismos (confianza en lo propio, búsqueda de desafíos)
- Hacer software reusable es mas difícil y costoso (paga a la larga y requiere procesos especiales)
- Encontrar, aprender a usar y adaptar componentes reusables requiere esfuerzo adicional

¿qué dificultades tenemos?

- Dificultades de mantenimiento si no tenemos control de los componente que reusamos
- Algunos programadores prefieren hacer todo ellos mismos (confianza en lo propio, búsqueda de desafíos)
- Hacer software reusable es mas difícil y costoso (paga a la larga y requiere procesos especiales)
- Encontrar, aprender a usar y adaptar componentes reusables requiere esfuerzo adicional

¿que estrategias tenemos?



Sommervielle –Software Engineering - pp 429

LIBRERIAS DE CLASES / TOOLKITS

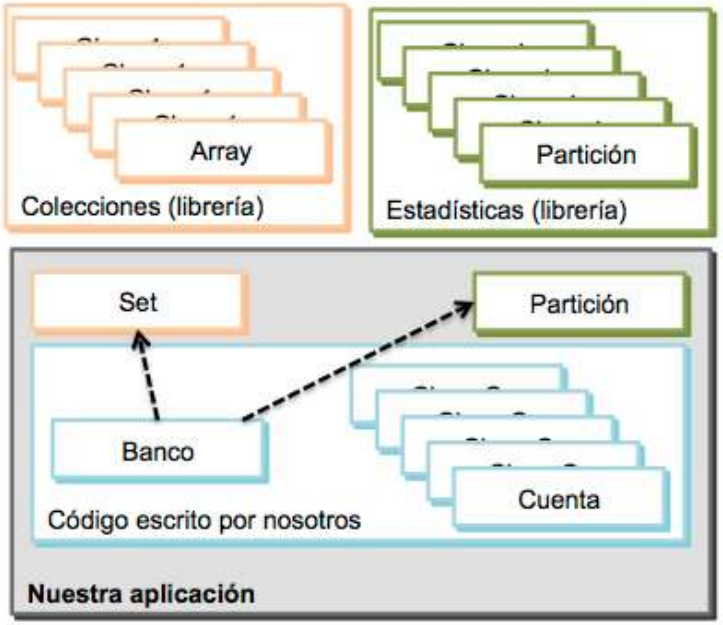
Librería de clases

- Resuelven problemas comunes a la mayoría de las aplicaciones
  - P.e., manejo de archivos, funciones aritméticas, colecciones, fechas,
- Cada clase en la librería resuelve un problema concreto, es independiente del contexto de uso, no espera nada de nuestro código, y generalmente independiente de otras clases en la librería
- Nuestro código controla/usa a los objetos de las librerías

Librería de clases

- Resuelven problemas comunes a la mayoría de las aplicaciones
  - P.e., manejo de archivos, funciones aritméticas, colecciones, fechas,
- Cada clase en la librería resuelve un problema concreto, es independiente del contexto de uso, no espera nada de nuestro código y generalmente independiente de otras clases en la librería
- Nuestro código controla/usa a los objetos de las librerías

Librería de clases



munes a la mayoría de las

funciones aritméticas,

resuelve un problema
nte del contexto de uso, y
iente de otras clases en la

- Nuestro código controla/usa a los objetos de las librerías

Algunas librerías de clases

Java
Codec (apache common) Clases (hoy, 18) que implementan algoritmos de encoding/decoding algorithms (por ejemplo, phonetic, base64, URL).
Compress (apache common) Clases para trabajar con archivos tar, zip, bzip2, etc.
Math (apache common) Clases que implementan componentes autocontenidos de matemáticas y estadísticas.
java.útil.collections Clases que implementan estructuras de datos (colecciones) y algoritmos para manipularlas

FRAMEWORKS ORIENTADOS A OBJETOS

Frameworks Orientados a Objetos

- Un framework es una *aplicación "semi-completa", "reusable", que puede ser especializada para producir aplicaciones a medida…*
- *. . .un conjunto de clases concretas y abstractas, relacionadas para proveer una arquitectura reusable para una familia de aplicaciones relacionadas…*

Frameworks Orientados a Objetos

- Los desarrolladores incorporan el framework en sus aplicaciones y lo especializan para cubrir sus necesidades …
- *… el framework define el esqueleto, y el desarrollador define sus propias caracteristicas para completar el esquelero …*
- *… (a diferencia de un toolkit o librería) un framework provee una estructura coherente en lugar de un conjunto de "clases útiles" …*

Tudose, Catalin. *JUnit in Action, Third Edition*. 3rd edition. Shelter Island, NY: Manning Publications, 2020.

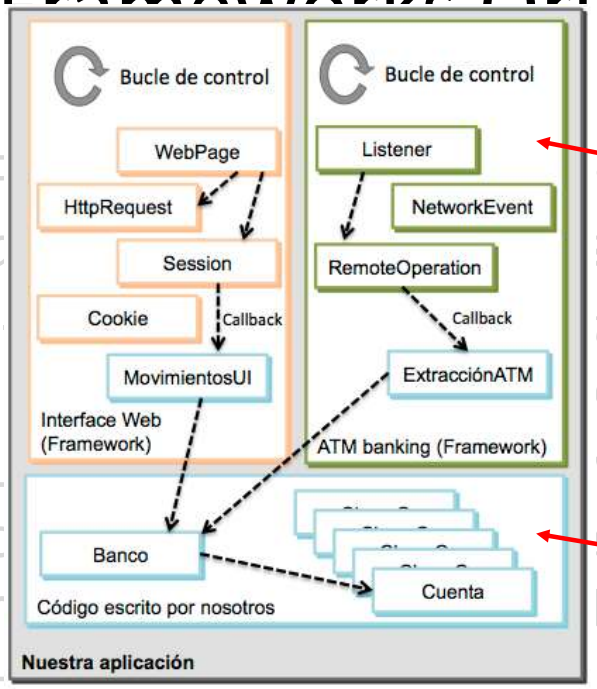
Frameworks Orientados a Objetos

- *Proveer una solución reusable para una familia de aplicaciones/problemas relacionadas*
  - *P.e., interfaces web, editores gráficos, aplicaciones colaborativas, gestores de contenidos*
- Las clases en el framework se relacionan (herencia, conocimiento, envío de mensajes) de manera que resuelven la mayor parte del problema en cuestión – conforman un todo
- El código del framework controla al nuestro

Frameworks Orientados a Objetos

- *Proveer una solución reusable para una familia de aplicaciones/problemas relacionadas*
  - *P.e., interfaces web, editores gráficos, aplicaciones colaborativas, gestores de contenidos*
- Las clases en el framework se relacionan (herencia, conocimiento, envío de mensajes) de manera que resuelven la mayor parte del problema en cuestión – conforman un todo
- El código del framework controla al nuestro

Frameworks Orientados a Objetos



- El código del framework controla al nuestro:

INVERSIÓN DE CONTROL

Frameworks de infraestructura

- Estos frameworks ofrecen una infraestructura portable y eficiente sobre la cual construir una gran variedad de aplicaciones
- Algunos de los focos de este tipo de frameworks son: interfaces de usuario (desktop, web, moviles), seguridad, contenedores de aplicación, procesamiento de imágenes, procesamiento de lenguaje, comunicaciónes
- Atacan problemas generales del desarrollo de software, que por lo general no son percibidos completamente por el usuario de la aplicación (es decir, resuelven problemas de los programadores, no de los usuarios)
- Es común que se los incluya como parte de plataformas de desarrollo (Java tiene los suyos, al igual de .NET)

Frameworks de integración

- Estos frameworks se utilizan comunmente para integrar componentes de aplicación distribuidos (p.e., la base de datos con la aplicación y ésta con su cliente liviano)
- Los frameworks de integración (o frameworks middleware) estan diseñados para aumentar la capacidad de los desarrolladores de modularizar, reusar y extender su infraestrcutura de software para que trabaje transparentemente en un ambiente distribuido
- El mercado de los frameworks de integración es muy activo y, al igual que ciertos frameworks de infraestructura, se han vuelto commodities (mercancia de uso común)
- Ejemplos: Object Relational Mapping, Message Oriented Middleware, Orchestration Frameworks

Frameworks de aplicación (o Enterprise)

- Estos frameworks atancan dominios de aplicación amplios que son pilares fundamentales de las actividades de las empresas
- Ejemplos de frameworks enterprise son aquellos en el dominio de los ERP (Enterprise Resource Planning), CRM (Customer Relationship Management) Gestión de documentos, Cálculos financieros.
- Los problemas que atacan derivan directamente de las necesidades de los usuarios de las aplicaciones y por tanto hacen que el retorno de la inversión en su desarrollo/adquisición sea mas evidente y justificado
- Un framework entreprise puede encapsular el conocimiento y experiencia de muchos años de una empresa, tranformandose en la clave de su ventaja competitiva y su mas preciado capital
- Son la base para la implementación de líneas de productos (SPL)

Algunos frameworks conocidos

Java
Hibernate Framework de integración que resuelve el mapeo de objetos a bases de datos relacionales
jBPM Framework de aplicación (y suite de herramientas) para construir aplicaciones en las que se modelan, ejecutan, y monitorean procesos de negocios.
jUnit Framework de aplicación/infraestructura que resuelve la automatización de tests de unidad
libGDX Framework de aplicación, para construir juegos en Java, multiplataforma.
Spring Familia de frameworks de infraestructura e integración, enfocando una amplia variedad de necesidades (Testing, Data, Web, etc.)

Frozenspot vs Hotspots

- Frozenspots de un framework
  - Todas las aplicaciones construídas con un mismo framework tienen aspectos en común que no podemos cambiar
- Hotspots de un framework
  - El framework ofrece puntos de extensión que nos permiten introducir variantes y así construir aplicaciones diferentes

Caja blanca vs Caja negra

- Los puntos de extensión pueden implementarse en base a herencia o en base a composición
- A los frameworks que utilizan herencia en sus puntos de extensión, les llamamos de Caja Blanca (Whitebox)
- A los que utilizan composición les llamamos de Caja Negra (blackbox)
- La mayoría de los frameworks están en algún lugar en el medio
  - Algunos usuarios los ven como caja negra
  - Otros como caja blanca

Material adicional

Frameworks: Guía de lectura