

Paradigmas Avanzados de Programación

Principios de la P00 y Herencia

Juan Francisco Díaz Frias

Maestría en Ingeniería, Énfasis en Ingeniería de Sistemas y Computación
Escuela de Ingeniería de Sistemas y Computación,
home page: <http://eisc.univalle.edu.co>
Universidad del Valle - Cali, Colombia

Plan

- 1 Principios de la P00
- 2 Herencia
- 3 Conceptos para definir clases

Plan

- 1 Principios de la P00
- 2 Herencia
- 3 Conceptos para definir clases

Plan

- 1 Principios de la P00
- 2 Herencia
- 3 Conceptos para definir clases

Principios de la P00

El modelo de computación

El modelo de computación de la P00 es el modelo con estado.

¿Qué es un programa?

Los programas son colecciones de abstracciones de datos que interactúan entre ellas. Dos principios para construirlas:

- Las abstracciones de datos deben tener estado por defecto.
- El estilo objeto, por defecto, debería ser el estilo APD, el cual fomenta el polimorfismo y la herencia.

En resumen...

La P00 es la programación con abstracción de datos en forma de objetos, estado explícito, polimorfismo, y herencia.

Principios de la P00

El modelo de computación

El modelo de computación de la P00 es el modelo con estado.

¿Qué es un programa?

Los programas son colecciones de abstracciones de datos que interactúan entre ellas. Dos principios para construirlas:

- Las abstracciones de datos deben tener estado por defecto.
- El estilo objeto, por defecto, debería ser el estilo APD, el cual fomenta el polimorfismo y la herencia.

En resumen...

La P00 es la programación con abstracción de datos en forma de objetos, estado explícito, polimorfismo, y herencia.

Principios de la P00

El modelo de computación

El modelo de computación de la P00 es el modelo con estado.

¿Qué es un programa?

Los programas son colecciones de abstracciones de datos que interactúan entre ellas. Dos principios para construirlas:

- Las abstracciones de datos deben tener estado por defecto.
- El estilo objeto, por defecto, debería ser el estilo APD, el cual fomenta el polimorfismo y la herencia.

En resumen...

La P00 es la programación con abstracción de datos en forma de objetos, estado explícito, polimorfismo, y herencia.

Herencia

Observación

Muchas abstracciones de datos tienen mucho en común, v. gr. los conjuntos.
¿Podemos implementarlas sin duplicar las partes comunes?

Problema: duplicación de código

- Programas más largos.
- Se dificulta el mantenimiento: modificación de todas las copias.
- Copias ligeramente diferentes: relación entre cambios no obvia.

Herencia

Observación

Muchas abstracciones de datos tienen mucho en común, v. gr. los conjuntos.
¿Podemos implementarlas sin duplicar las partes comunes?

Problema: duplicación de código

- Programas más largos.
- Se dificulta el mantenimiento: modificación de todas las copias.
- Copias ligeramente diferentes: relación entre cambios no obvia.

Herencia

¿Para qué el concepto de herencia?

Para reducir el problema de la duplicación de código y para clarificar las relaciones entre las abstracciones de datos.

¿Cómo?

- Una abstracción “hereda” de otra(s), teniendo esencialmente la misma funcionalidad que las otras, posiblemente con algunas extensiones y modificaciones.
- Sólo se especifican las diferencias entre la abstracción de datos y sus ancestros.
- A tal definición incremental de una abstracción de datos se le denomina una *clase*.

Herencia

¿Para qué el concepto de herencia?

Para reducir el problema de la duplicación de código y para clarificar las relaciones entre las abstracciones de datos.

¿Cómo?

- Una abstracción “hereda” de otra(s), teniendo esencialmente la misma funcionalidad que las otras, posiblemente con algunas extensiones y modificaciones.
- Sólo se especifican las diferencias entre la abstracción de datos y sus ancestros.
- A tal definición incremental de una abstracción de datos se le denomina una *clase*.

Herencia: clase

¿Cómo se define una clase?

Por combinación de una o más clases existentes con una descripción de las extensiones y modificaciones requeridas.

¿Concepto nuevo?

NO! Abstracción lingüística.

- Manipulación sintáctica, donde la sintaxis de la clase nueva se deriva a partir de las clases originales.
- Función que recibe como entrada valores de tipo clase y devuelve como salida la clase nueva.

Herencia: clase

¿Cómo se define una clase?

Por combinación de una o más clases existentes con una descripción de las extensiones y modificaciones requeridas.

¿Concepto nuevo?

NO! Abstracción lingüística.

- Manipulación sintáctica, donde la sintaxis de la clase nueva se deriva a partir de las clases originales.
- Función que recibe como entrada valores de tipo clase y devuelve como salida la clase nueva.

Use la herencia con cuidado (1)

Conocimiento profundo de las clases ancestro

Pues ellas pueden romper fácilmente un invariante de clase.

Interfaz adicional a mantener

La capacidad de extender una clase se puede ver como una manera adicional de interactuar con esa clase. Esta interfaz tiene que mantenerse a lo largo de la vida del componente.



Las clases son finales (no se pueden extender) por defecto. Para extenderlas, se requiere la acción explícita del programador.

Use la herencia con cuidado (1)

Conocimiento profundo de las clases ancestro

Pues ellas pueden romper fácilmente un invariante de clase.

Interfaz adicional a mantener

La capacidad de extender una clase se puede ver como una manera adicional de interactuar con esa clase. Esta interfaz tiene que mantenerse a lo largo de la vida del componente.



Las clases son finales (no se pueden extender) por defecto. Para extenderlas, se requiere la acción explícita del programador.

Use la herencia con cuidado (2)

Factorización de una aplicación vs. dispersión de la implementación

La implementación no existe en un sitio; todas las abstracciones de las que se hereda tienen que considerarse juntas.

Esto dificulta entender la abstracción, y, paradójicamente, puede dificultar su mantenimiento.

Lección

La herencia debe utilizarse en "pequeñas cantidades."

Use la herencia con cuidado (2)

Factorización de una aplicación vs. dispersión de la implementación

La implementación no existe en un sitio; todas las abstracciones de las que se hereda tienen que considerarse juntas.

Esto dificulta entender la abstracción, y, paradójicamente, puede dificultar su mantenimiento.

Lección

La herencia debe utilizarse en “pequeñas cantidades.”

Alternativa a la herencia...

Programación Basada en componentes

La idea es definir un componente que encapsula a otro componente y provee una funcionalidad modificada.

Compromiso Herencia - Componentes

La herencia es más flexible pero puede romper un invariante de clase, mientras que la composición de componentes es menos flexible pero no puede romper un invariante de componente.

Alternativa a la herencia...

Programación Basada en componentes

La idea es definir un componente que encapsula a otro componente y provee una funcionalidad modificada.

Compromiso Herencia - Componentes

La herencia es más flexible pero puede romper un invariante de clase, mientras que la composición de componentes es menos flexible pero no puede romper un invariante de componente.

Conjunto de conceptos para definir clases

Para construir abstracciones de datos completas

- Los métodos, los atributos, y las propiedades.
- Mensajes de primera clase y atributos de primera clase: para lograr formas poderosas de polimorfismo.

Para construir abstracciones de datos incrementalmente

Los conceptos relacionados con la herencia, i.e., los que definen cómo se relaciona una clase con las clases existentes.

Conjunto de conceptos para definir clases

Para construir abstracciones de datos completas

- Los métodos, los atributos, y las propiedades.
- Mensajes de primera clase y atributos de primera clase: para lograr formas poderosas de polimorfismo.

Para construir abstracciones de datos incrementalmente

Los conceptos relacionados con la herencia, i.e., los que definen cómo se relaciona una clase con las clases existentes.