Abstracción de datos

Ricardo Pérez López

IES Doñana, curso 2019/2020

1. Tipos abstractos de datos

1.1. Concepto, terminología y ejemplos

1.1.1 Ejemplo: números racionales

1.1. Concepto, terminología y ejemplos

- Al considerar el amplio conjunto de cosas en el mundo que nos gustaría representar en nuestros programas, descubrimos que la mayoría de ellas tienen una estructura compuesta.
- Por ejemplo, una posición geográfica tiene coordenadas de latitud y longitud. Para representar posiciones, nos gustaría que nuestro lenguaje de programación tenga la capacidad de juntar una latitud y longitud para formar un par, un valor de datos compuesto que nuestros programas pueden manipular como una sola unidad conceptual, pero que también tiene dos partes que pueden ser considerados individualmente por separado.
- El uso de datos compuestos nos permite aumentar la modularidad de nuestros programas.
- Si podemos manipular posiciones geográficas como valores completos, entonces podemos proteger partes de nuestro programa que calculan usando posiciones de los detalles de cómo se representan esas posiciones.
- La técnica general de aislar las partes de un programa que se ocupan de cómo se representan los datos de las partes que se ocupan de cómo se manipulan los datos es una poderosa metodología de diseño llamada abstracción de datos.
- La abstracción de datos hace que los programas sean mucho más fáciles de diseñar, mantener y modificar.
- La abstracción de datos es de carácter similar a la abstracción funcional.
- Cuando creamos una abstracción funcional, los detalles de cómo se implementa

Ejemplo: números racionales

- Un número racional es una relación de enteros, y los números racionales constituyen una subclase importante de números reales. Un número racional como 1/3 o 17/29 generalmente se escribe como:

  /

  donde tanto como son marcadores de posición para valores enteros. Ambas partes son necesarias para caracterizar exactamente el valor del número racional. La división de enteros en realidad produce una aproximación flotante , perdiendo la precisión exacta de los enteros.

1 / 3 ,3333333333333333 1 / 3 == 0.33333333333333300000 # La división de números enteros, se obtiene una aproximación verdadera

- Sin embargo, podemos crear una representación exacta para números racionales combinando juntos el numerador y el denominador.
- Por el uso de abstracciones funcionales, sabemos que podemos comenzar a programar productivamente antes de implementar algunas partes de nuestro programa. Comencemos suponiendo que ya tenemos una manera de construir un número racional a partir de un numerador y un denominador. También suponemos que, dado un número racional, tenemos una forma de seleccionar su numerador y su componente denominador. Supongamos además que el constructor y los selectores están disponibles como las siguientes tres funciones:

racional (n, d) devuelve el número racional con numerador n y denominador d . numer (x) devuelve el numerador del número racional x . denom (x) devuelve el denominador del número racional x . Estamos utilizando una estrategia poderosa para diseñar

1.2. Programación con tipos abstractos de datos

2. Especificación

2.1. Especificaciones algebraicas

Géneros

Operaciones

Constructoras

Selectoras

2.2. Construcción de especificaciones

2.3. Verificación con especificaciones algebraicas

3. Implementación

3.1. Pilas

3.2. Colas

3.3. Listas