La estrategia de diseño
Propósito
Ejemplos
Cuerpo
Pruebas

Elementos de un programa, Programas Simples

Universidad del Valle

Contenido

1. La estrategia de diseño
   - Propósito
   - Ejemplos
   - Cuerpo
   - Pruebas

Diseñando Programas

Ejemplo:

La empresa de alimentos Coma-rico debe surtir el pedido de sus clientes llevando cierta cantidad de cajas de su producto Super-perro a distintos almacenes. Para poder surtir a los clientes, debe contratar algunos camiones. El administrador de la empresa Coma-rico debe solicitar el servicio de transportes a una empresa transportadora; para esto debe pedir un número de camiones. Los camiones de la empresa seleccionada pueden llevar máximo 30 cajas de producto.

Realice una función en Scheme, que dado el número de cajas del producto que debe entregar en el día, retorne el número de camiones que el administrador de Coma-rico debe solicitar a la empresa transportadora.

La cantidad de camiones es un entero

Diseñando Programas

Al leer el anterior problema surgen dudas como:

- Cuáles son los parámetros?
- Cuáles son los resultados?
- Cómo obtenemos el resultados dados los datos de entrada?
- Necesito operaciones especiales?
- Scheme las tiene o yo tengo que hacerlas?.
- Cómo se que mi programa está correcto?

Estrategia de diseño

Permite llevar a cabo el proceso de programar de forma ordenada.

Es una secuencia de pasos que facilitan el diseño de programas, porque nos indican lo que se debe hacer y el orden que debemos seguir

La estrategia de diseño conta de 4 pasos clave:

1. Entender el propósito del programa:
   resultado: contrato, propósito y plantilla.
2. Crear ejemplos de la ejecución del programa:
   resultado: proponer mínimo 3 ejemplos con entradas y resultados correctos
3. Escribir el programa.
   resultado: siguiendo la plantilla se termina de escribir el cuerpo del programa
4. Realizar pruebas para verificar que esté correcto.
   resultado: expresiones que permiten evaluar el programa.

Estrategia de diseño

Permite llevar a cabo el proceso de programar de forma ordenada.

Es una secuencia de pasos que facilitan el diseño de programas, porque nos indican lo que se debe hacer y el orden que debemos seguir

La estrategia de diseño conta de 4 pasos clave:

1. Entender el propósito del programa:
   resultado: contrato, propósito y plantilla.
2. Crear ejemplos de la ejecución del programa:
   resultado: proponer mínimo 3 ejemplos con entradas y resultados correctos
3. Escribir el programa.
   resultado: siguiendo la plantilla se termina de escribir el cuerpo del programa
4. Realizar pruebas para verificar que esté correcto.
   resultado: expresiones que permiten evaluar el programa.

Estrategia de diseño

Permite llevar a cabo el proceso de programar de forma ordenada.

Es una secuencia de pasos que facilitan el diseño de programas, porque nos indican lo que se debe hacer y el orden que debemos seguir

La estrategia de diseño conta de 4 pasos clave:

1. Entender el propósito del programa:
   resultado: contrato, propósito y plantilla.
2. Crear ejemplos de la ejecución del programa:
   resultado: proponer mínimo 3 ejemplos con entradas y resultados correctos
3. Escribir el programa.
   resultado: siguiendo la plantilla se termina de escribir el cuerpo del programa
4. Realizar pruebas para verificar que esté correcto.
   resultado: expresiones que permiten evaluar el programa.

Estrategia de diseño

Permite llevar a cabo el proceso de programar de forma ordenada.

Es una secuencia de pasos que facilitan el diseño de programas, porque nos indican lo que se debe hacer y el orden que debemos seguir

La estrategia de diseño conta de 4 pasos clave:

1. Entender el propósito del programa:
   resultado: contrato, propósito y plantilla.
2. Crear ejemplos de la ejecución del programa:
   resultado: proponer mínimo 3 ejemplos con entradas y resultados correctos
3. Escribir el programa.
   resultado: siguiendo la plantilla se termina de escribir el cuerpo del programa
4. Realizar pruebas para verificar que esté correcto.
   resultado: expresiones que permiten evaluar el programa.

Estrategia de diseño

Permite llevar a cabo el proceso de programar de forma ordenada.

Es una secuencia de pasos que facilitan el diseño de programas, porque nos indican lo que se debe hacer y el orden que debemos seguir

La estrategia de diseño conta de 4 pasos clave:

1. Entender el propósito del programa:
   resultado: contrato, propósito y plantilla.
2. Crear ejemplos de la ejecución del programa:
   resultado: proponer mínimo 3 ejemplos con entradas y resultados correctos
3. Escribir el programa.
   resultado: siguiendo la plantilla se termina de escribir el cuerpo del programa
4. Realizar pruebas para verificar que esté correcto.
   resultado: expresiones que permiten evaluar el programa.

Estrategia de diseño

Permite llevar a cabo el proceso de programar de forma ordenada.

Es una secuencia de pasos que facilitan el diseño de programas, porque nos indican lo que se debe hacer y el orden que debemos seguir

La estrategia de diseño conta de 4 pasos clave:

1. Entender el propósito del programa:
   resultado: contrato, propósito y plantilla.
2. Crear ejemplos de la ejecución del programa:
   resultado: proponer mínimo 3 ejemplos con entradas y resultados correctos
3. Escribir el programa.
   resultado: siguiendo la plantilla se termina de escribir el cuerpo del programa
4. Realizar pruebas para verificar que esté correcto.
   resultado: expresiones que permiten evaluar el programa.

Estrategia de diseño

Permite llevar a cabo el proceso de programar de forma ordenada.

Es una secuencia de pasos que facilitan el diseño de programas, porque nos indican lo que se debe hacer y el orden que debemos seguir

La estrategia de diseño conta de 4 pasos clave:

1. Entender el propósito del programa:
   resultado: contrato, propósito y plantilla.
2. Crear ejemplos de la ejecución del programa:
   resultado: proponer mínimo 3 ejemplos con entradas y resultados correctos
3. Escribir el programa.
   resultado: siguiendo la plantilla se termina de escribir el cuerpo del programa
4. Realizar pruebas para verificar que esté correcto.
   resultado: expresiones que permiten evaluar el programa.

Entender el propósito del programa

Para entender un problema y convertirlo un programa debemos primero encontrar las entradas y las salidas

La empresa de alimentos Coma-rico debe surtir el pedido de sus clientes llevando cierta cantidad de cajas de su producto Super-perro a distintos almacenes. Para poder surtir a los clientes, debe contratar algunos camiones. El administrador de la empresa Coma-rico debe solicitar el servicio de transportes a una empresa transportadora; para esto debe pedir un número de camiones. Los camiones de la empresa seleccionada pueden llevar máximo 30 cajas de producto.

Con estos elementos clave escribimos el contrato del programa:

;; nro_Camiones: número → número

Entender el propósito del programa

Para entender un problema y convertirlo un programa debemos primero encontrar las entradas y las salidas

La empresa de alimentos Coma-rico debe surtir el pedido de sus clientes llevando cierta cantidad de cajas de su producto Super-perro a distintos almacenes. Para poder surtir a los clientes, debe contratar algunos camiones. El administrador de la empresa Coma-rico debe solicitar el servicio de transportes a una empresa transportadora; para esto debe pedir un número de camiones. Los camiones de la empresa seleccionada pueden llevar máximo 30 cajas de producto.

Con estos elementos clave escribimos el contrato del programa:

;; nro_Camiones: número $\longrightarrow$ número

Entender el propósito del programa

Para entender un problema y convertirlo un programa debemos primero encontrar las entradas y las salidas

La empresa de alimentos Coma-rico debe surtir el pedido de sus clientes llevando cierta cantidad de cajas de su producto Super-perro a distintos almacenes. Para poder surtir a los clientes, debe contratar algunos camiones. El administrador de la empresa Coma-rico debe solicitar el servicio de transportes a una empresa transportadora; para esto debe pedir un número de camiones. Los camiones de la empresa seleccionada pueden llevar máximo 30 cajas de producto.

Con estos elementos clave escribimos el contrato del programa:

;; nro_Camiones: número $\longrightarrow$ número

El Contrato

El contrato consta de dos partes: el nombre del programa y el tipo de los datos de entrada y salida.

- **El nombre** debe identificar el problema que se desea resolver.
- **Los datos** están separados por una flecha.

Luego del contrato escribimos el propósito del programa que son un par de líneas en las que escribimos qué es lo que calcula el programa.

;; propósito: calcular el número de camiones necesario para llevar N cajas.

La Plantilla es la siguiente:

```
(define (nro_Camiones N)
...)
```

Clave: Si el enunciado del problema tiene una fórmula matemática, el número de variables de la formula es el número de datos de entrada.

El Contrato

El contrato consta de dos partes: el nombre del programa y el tipo de los datos de entrada y salida.

- **El nombre** debe identificar el problema que se desea resolver.
- **Los datos** están separados por una flecha.

Luego del contrato escribimos el propósito del programa que son un par de líneas en las que escribimos qué es lo que calcula el programa.

;; propósito: calcular el número de camiones necesario para llevar N cajas.
La Plantilla es la siguiente:

```
(define (nro_Camiones N)
...)
```

Clave: Si el enunciado del problema tiene una fórmula matemática, el número de variables de la formula es el número de datos de entrada.

El Contrato

El contrato consta de dos partes: el nombre del programa y el tipo de los datos de entrada y salida.

- **El nombre** debe identificar el problema que se desea resolver.
- **Los datos** están separados por una flecha.

Luego del contrato escribimos el propósito del programa que son un par de líneas en las que escribimos qué es lo que calcula el programa.

;; propósito: calcular el número de camiones necesario para llevar N cajas.

La Plantilla es la siguiente:

```
(define (nro_Camiones N)
...)
```

Clave: Si el enunciado del problema tiene una fórmula matemática, el número de variables de la formula es el número de datos de entrada.

El Contrato

El contrato consta de dos partes: el nombre del programa y el tipo de los datos de entrada y salida.

- **El nombre** debe identificar el problema que se desea resolver.
- **Los datos** están separados por una flecha.

Luego del contrato escribimos el propósito del programa que son un par de líneas en las que escribimos qué es lo que calcula el programa.

;; propósito: calcular el número de camiones necesario para llevar N cajas.

La Plantilla es la siguiente:

```
(define (nro_Camiones N)
...)
```

Clave: Si el enunciado del problema tiene una fórmula matemática, el número de variables de la formula es el número de datos de entrada.

Crear Ejemplos

Proponer ejemplos requiere que dados los datos de entrada usted realize el cálculo de el resultado o la salida correcta.
Usted va a entender mejor el proceso para obtener un resultado, así podrá escribirlo en scheme de forma más sencilla.

```
;; (nro_Camiones 65) debe retornar 3)
;; (nro_Camiones 120) debe retornar 4)
;; (nro_Camiones 1200) debe retornar 40)
```

Crear Ejemplos

Proponer ejemplos requiere que dados los datos de entrada usted realize el cálculo de el resultado o la salida correcta. Usted va a entender mejor el proceso para obtener un resultado, así podrá escribirlo en scheme de forma más sencilla.

```
;; (nro_Camiones 65) debe retornar 3)
;; (nro_Camiones 120) debe retornar 4)
;; (nro_Camiones 1200) debe retornar 40)
```

Crear Ejemplos

Proponer ejemplos requiere que dados los datos de entrada usted realize el cálculo de el resultado o la salida correcta. Usted va a entender mejor el proceso para obtener un resultado, así podrá escribirlo en scheme de forma más sencilla.

```
;; (nro_Camiones 65) debe retornar 3)
;; (nro_Camiones 120) debe retornar 4)
;; (nro_Camiones 1200) debe retornar 40)
```

Escribir el programa

El programa consta de dos partes: encabezado y cuerpo.

El encabezado es la parte donde declaramos el nombre del programa y los parámetros.

El cuerpo del programa es la parte en donde escribimos en scheme el procedimiento que se debe realizar para obtener el resultado.

Para empezar a escribir el programa tomamos la plantilla:

```
(define (nro_Camiones N)
...)
```

El encabezado ya está escrito en la plantilla ahora se reemplazan los . . . (puntos) por una expresión en scheme que nos permita dar una solución.

```
(define (nro_Camiones N)
(/ N 30) )
```

Escribir el programa

El programa consta de dos partes: encabezado y cuerpo.
El encabezado es la parte donde declaramos el nombre del programa y los parámetros.
El cuerpo del programa es la parte en donde escribimos en scheme el procedimiento que se debe realizar para obtener el resultado.
Para empezar a escribir el programa tomamos la plantilla:

```
(define (nro_Camiones N)
...)
```

El encabezado ya está escrito en la plantilla ahora se reemplazan los . . . (puntos) por una expresión en scheme que nos permita dar una solución.

```
(define (nro_Camiones N)
(/ N 30) )
```

Escribir el programa

El programa consta de dos partes: encabezado y cuerpo.
El encabezado es la parte donde declaramos el nombre del programa y los parámetros.
El cuerpo del programa es la parte en donde escribimos en scheme el procedimiento que se debe realizar para obtener el resultado.

Para empezar a escribir el programa tomamos la plantilla:

```
(define (nro_Camiones N)
...)
```

El encabezado ya está escrito en la plantilla ahora se reemplazan los . . . (puntos) por una expresión en scheme que nos permita dar una solución.

```
(define (nro_Camiones N)
(/ N 30) )
```

Escribir el programa

El programa consta de dos partes: encabezado y cuerpo.
El encabezado es la parte donde declaramos el nombre del programa y los parámetros.
El cuerpo del programa es la parte en donde escribimos en scheme el procedimiento que se debe realizar para obtener el resultado.
Para empezar a escribir el programa tomamos la plantilla:

```
(define (nro_Camiones N)
...)
```

El encabezado ya está escrito en la plantilla ahora se reemplazan los ... (puntos) por una expresión en scheme que nos permita dar una solución.

```
(define (nro_Camiones N)
(/ N 30) )
```

Escribir el programa

El programa consta de dos partes: encabezado y cuerpo.
El encabezado es la parte donde declaramos el nombre del programa y los parámetros.
El cuerpo del programa es la parte en donde escribimos en scheme el procedimiento que se debe realizar para obtener el resultado.
Para empezar a escribir el programa tomamos la plantilla:

```
(define (nro_Camiones N)
...)
```

El encabezado ya está escrito en la plantilla ahora se reemplazan los ... (puntos) por una expresión en scheme que nos permita dar una solución.

```
(define (nro_Camiones N)
                (/ N 30) )
```

Pruebas

Debe probarse el programa mínimo con los ejemplos que se propuso inicialmente.

No es posible probar todos los posibles valores, pero el proceso de pruebas es muy importante para encontar errores en el programa, ya sea en su escritura o en el procedimiento.
Las pruebas deben ir en la ventana de definición, para que al momento de ejecutar, se realizen las pruebas.

(nro_Camiones 65) debe dar 3
(nro_Camiones 120) debe dar 4
(nro_Camiones 1200) debe retornar 40

Pruebas

Debe probarse el programa mínimo con los ejemplos que se propuso inicialmente.

No es posible probar todos los posibles valores, pero el proceso de pruebas es muy importante para encontar errores en el programa, ya sea en su escritura o en el procedimiento.

Las pruebas deben ir en la ventana de definición, para que al momento de ejecutar, se realizen las pruebas.

(nro_Camiones 65) debe dar 3
(nro_Camiones 120) debe dar 4
(nro_Camiones 1200) debe retornar 40

Pruebas

Debe probarse el programa mínimo con los ejemplos que se propuso inicialmente.

No es posible probar todos los posibles valores, pero el proceso de pruebas es muy importante para encontar errores en el programa, ya sea en su escritura o en el procedimiento.
Las pruebas deben ir en la ventana de definición, para que al momento de ejecutar, se realizen las pruebas.

(nro_Camiones 65) debe dar 3
(nro_Camiones 120) debe dar 4
(nro_Camiones 1200) debe retornar 40

El programa completo es el siguiente

```
;; contrato: nro_Camiones: número → número
;; propósito: calcular el número de camiones necesario para
;; llevar cierta cantidad de cajas. La capacidad de cada camion
;; es de 30 cajas
;; Ejemplos:
;; (nro_Camiones 65) debe retornar 3)
;; (nro_Camiones 120) debe retornar 4)
;; (nro_Camiones 1200) debe retornar 40
;; Programa:
(define (nro_Camiones N)
                  (/ N 30) )
;; Pruebas:
(nro_Camiones 65)
(nro_Camiones 120)
(nro_Camiones 1200)
```

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Resumen

- Tenemos una estrategia de diseño, con cuatro pasos que debemos seguir en orden
- Los pasos son:
  1. Entender el propósito del programa:
     resultado: contrato y propósito
  2. Crear ejemplos de la ejecución del programa:
     resultado: 3 ejemplos con entradas y resultados correctos
  3. Escribir el programa.
     resultado: encabezado y cuerpo del programa
  4. Realizar pruebas para verificar que esté correcto.
     resultado: expresiones que permiten evaluar el programa.

Fin de la Presentación.