## Técnicas de Diseño

M.C. Carlos Rojas Sánchez<sup>1</sup>

<sup>1</sup>Licenciatura en Informática Universidad del Mar :: Puerto Escondido

.: Diseño Estructurado de Algoritmos :.

#### Técnicas de diseño

Top-down y Bottom-up son estrategias de procesamiento de información, especialmente en lo relativo al software. Por extensión se aplican también a otras ciencias sociales y exactas.



Técnicas de Diseño

1 Top down



## Top down ('de arriba abajo')

- El diseño top-down fue promovido en la década de 1970 por los investigadores de IBM Harlan Mills y Niklaus Wirth.
- Mills desarrolló los conceptos de programación estructurada.
- Niklaus Wirth, que entre sus logros está el desarrollo del lenguaje de programación Pascal.



## Top down ('de arriba abajo')

- En el modelo Top-down se formula un resumen del sistema, sin especificar detalles. Cada parte del sistema se refina diseñando con mayor detalle.
- Cada parte nueva es entonces redefinida, cada vez con mayor detalle, hasta que la especificación completa es lo suficientemente detallada para validar el modelo.

# Top down ('de arriba abajo')

■ El modelo "Top-down" se diseña con frecuencia con la ayuda de "cajas negras" que hacen más fácil cumplir requerimientos aunque estas cajas negras no expliquen en detalle los componentes individuales



Técnicas de Diseño

2 Bottom up



# Buttom up ('de abajo arriba')

- En el diseño Bottom-up las partes individuales se diseñan con detalle y luego se enlazan para formar componentes más grandes, que a su vez se enlazan hasta que se forma el sistema completo.
- Las estrategias basadas en el flujo de información "bottom-up", se basan en el conocimiento de todas las variables que pueden afectar los elementos del sistema.

Buttom up ('de abajo arriba')

- Bottom-up hace énfasis en la programación y pruebas tempranas, que pueden comenzar tan pronto se ha especificado el primer módulo.
- Este enfoque tiene el riesgo de programar cosas sin saber como se van a conectar al resto del sistema, y esta conexión puede no ser tan fácil como se creyó al comienzo.
- La reutilización del código es uno de los mayores beneficios del enfoque bottom-up.



Técnicas de Diseño

3 Ejemplos





Planteamiento Determinar si un numero es par o impar



Planteamiento Determinar si un numero es par o impar

Definición ¿tipo de numero?, ¿qué es un numero par?



Planteamiento Determinar si un numero es par o impar

Definición ¿tipo de numero?, ¿qué es un numero par?

Análisis Entrada - Numero entero Salida - Leyenda "par" o "impar"



Planteamiento Determinar si un numero es par o impar

Definición ¿tipo de numero?, ¿qué es un numero par?

Análisis Entrada - Numero entero Salida - Leyenda "par" o "impar"

Método Uso del operador módulo (%), el operador de módulo, o resto, divide dos números y devuelve solamente el resto





Planteamiento Determinar si un año es bisiesto o no



Planteamiento Determinar si un año es bisiesto o no

Definición ¿tipo de valor?, ¿qué es un año bisiesto?



Planteamiento Determinar si un año es bisiesto o no

Definición ¿tipo de valor?, ¿qué es un año bisiesto?

Análisis Entrada - Numero entero Salida - Leyenda "Es bisiesto" o "No es bisiesto"



Planteamiento Determinar si un año es bisiesto o no

Definición ¿tipo de valor?, ¿qué es un año bisiesto?

Análisis Entrada - Numero entero Salida - Leyenda "Es bisiesto" o "No es bisiesto"

Método Uso del operador módulo (%), el operador de módulo, o resto, divide dos números y devuelve solamente el resto





Planteamiento Determinar si un numero es primo o no



Planteamiento Determinar si un numero es primo o no

Definición ¿tipo de valor?, ¿qué es un numero primo?



Planteamiento Determinar si un numero es primo o no

Definición ¿tipo de valor?, ¿qué es un numero primo?

Análisis Entrada - Numero entero mayor a 1

Salida - Leyenda "Es primo" o "No es primo"



Planteamiento Determinar si un numero es primo o no

Definición ¿tipo de valor?, ¿qué es un numero primo?

Análisis Entrada - Numero entero mayor a 1 Salida - Leyenda "Es primo" o "No es primo"

Método Uso del operador módulo (%), el operador de módulo, o resto, divide dos números y devuelve solamente el resto





Planteamiento Contar los valores entre 1 y un número dado



Planteamiento Contar los valores entre 1 y un número dado

Definición ¿tipo de valor?, ¿valor máximo?



Planteamiento Contar los valores entre 1 y un número dado

Definición ¿tipo de valor?, ¿valor máximo?

Análisis Entrada - Numero entero mayor a 1 Salida - Valor de la suma



Planteamiento Contar los valores entre 1 y un número dado

Definición ¿tipo de valor?, ¿valor máximo?

Análisis Entrada - Numero entero mayor a 1 Salida - Valor de la suma

Método 
$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$





Planteamiento Determinar si una palabra es o no palíndromo



Planteamiento Determinar si una palabra es o no palíndromo

Definición ¿Qué es un palíndromo?



Planteamiento Determinar si una palabra es o no palíndromo

Definición ¿Qué es un palíndromo?

Análisis Entrada - Palabra y su longitud Salida - Leyenda "Es palíndromo" o "No es palíndromo"



Planteamiento Determinar si una palabra es o no palíndromo

Definición ¿Qué es un palíndromo?

Análisis Entrada - Palabra y su longitud Salida - Leyenda "Es palíndromo" o "No es palíndromo"

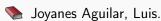
Método Comparaciones

$$par - \frac{n}{2}$$

$$impar - \frac{n-1}{2}$$



## Bibliografía I



Fundamentos de programación, algoritmos y estructura de datos.

McGraw-Hill. 2003, 3<sup>a</sup> Edición.

Cairó Battistutti, Osvaldo Gabriel.

Metodología de la programación: algoritmos, diagramas de flujo y programas.

Alfaomega. 2005, 3<sup>a</sup> Edición.

