

Chapter 1 Programming: A General Overview

1.2 Revision matematicas

1.2.1 Exponentes

Para empezar de manera basica, los exponentes es una forma de expresar una multiplicacion de una expresion por si misma un numero determinado de veces. La letra "a" se usa como base, en otras palabras, es el numero que se usara para la multiplicacion, y la letra "n" es el que se usa como exponente, es decir, la cantidad de veces que se multiplicara la "a".

Ejemplos: $X^A X^B = X^{A+B}$ $X^A X^B = X^{A-B}$ $(X^A)^B = X^{AB}$ $X^N + X^N = 2X^N = X^{2N}$ $2^N + 2^N = 2^{N+1}$

1.2.2 Logaritmos

Los logaritmos son una funcion que dependen de una base y un argumento que su tasa de crecimiento va disminuyendo.

Ejemplos: $\log_A B = \log C B / \log C A$; $A, B, C > 0, A \neq 1$ $\log AB = \log A + \log B$; $A, B > 0$

1.2.3 Series

Con dudas

Las series son una suma del numero infinito de terminos de una sucesion.

Ejemplo: $S_n = \sum a_n$

1.2.4 Aritmetica modular

La aritmetica modular utiliza un solo numero fijo de varios resultados posibles en todos sus computos.

Ejemplo: Un caso particular de la aritmética modular es la llamada aritmética del reloj. Cuando a las 10 de la mañana se le agrega 5 horas se llega a las 3 de la tarde, es decir "10 + 5 = 3". También si a las 2 de la tarde se le quita 4 horas, el resultado es las 10, lo que equivale a decir que "2 - 4 = 10".

1.3 Recursividad

La recursividad es una característica de los lenguajes de programación que permite que un subprograma se invoque a sí mismo. Sirve para resolver problemas definibles en sus propios terminos. Algo en lo que se suele usar, es en la "funcion factorial".

1.4 Clases

Una clase es en general un modelo, receta o plantilla que define el estado y comportamiento de cierto tipo de objetos. Una clase puede pensarse como una colección de variables (atributos o propiedades) y funciones (métodos) que permiten representar un conjunto de datos y especificar las operaciones o procedimientos que permiten manipular tales datos. Cosnta de sus miembros, los cuales pueden ser datos o funciones.

Constructores

En las clases hay algo llamado "constructores" que es una personalización de cómo se inicializa una clase de sus miembros o invoca funciones cuando se crea un objeto de la clase. Un constructor tiene el mismo nombre que la clase y no devuelve ningún valor. Normalmente tienen accesibilidad pública.

Los constructores tienen diferentes tipos:

1. Constructores preterminados: son una de las funciones miembro especiales. En caso de que no se declare ningún constructor en una clase, el compilador proporcionará un constructor preterminado.
2. Constructores de copias: un constructor de copia inicializa un objeto copiando los valores de los miembros de un objeto del mismo tipo. Si los miembros de una clase son simples, el constructor de copia generado por el compilador es suficiente y no es necesario definir el suyo propio.
3. Constructores de movimiento: es una función de miembro especial que mueve la propiedad de los datos de un objeto existente a una nueva variable sin copiar los datos originales, pueden aumentar significativamente la eficacia del programa al pasar objetos grandes.

Vector y String

Vectores: son estructuras de datos similares a los arreglos, pero más desarrollados, ya que crecen y decrecen dinámicamente, según se necesite.

String: Son utilizados para almacenar texto y caracteres en la memoria del programa y son necesarios para la manipulación y procesamiento de datos en muchos tipos de programas.

1.5 Detalles

Punteros

Con dudas

es una variable que almacena la dirección donde reside otro objeto. Es el mecanismo fundamental utilizado en muchas estructuras de datos. Por ejemplo, para almacenar una lista de elementos, podríamos usar una matriz contigua, pero la inserción en el medio de la matriz contigua requiere la reubicación de muchos elementos. Los punteros se usan ampliamente en C y C++ para tres propósitos principales:

para asignar nuevos objetos en el montón, para pasar funciones a otras funciones para iterar sobre elementos en matrices u otras estructuras de datos.

Lvalues, Rvalues y Referencias

Lvalue: lvalue es una expresión que identifica un objeto no temporal.

Rvalue: es una expresión que identifica un objeto temporal o es un valor (como una constante literal) que no está asociado a cualquier objeto.

Referencia: almacena la dirección de un objeto que se encuentra en otra parte de la memoria.

1.6 Plantillas

Las plantillas son la base para la programación genérica en C++. Es una construcción que genera un tipo o función normal en tiempo de compilación en función de los argumentos que proporciona el usuario para los

parámetros de la plantilla. Las plantillas permiten definir las operaciones de una clase o función y permiten al usuario especificar en qué tipos concretos deben funcionar esas operaciones. Una plantilla de función no es una función real, sino que es un patrón de lo que podría convertirse en una función.

1.7 Uso de matrices

Las matrices son una estructura de datos bastante similar a los vectores o arreglos. De hecho, una matriz no es más que una serie de vectores contenidos uno en el otro (u otros), es decir, una matriz es un vector cuyas posiciones son otros vectores. Hablemos con más detalle de esto para quedar más claros.

Operator[]

La idea del operador[] es que si tenemos una matriz m , entonces $m[i]$ debería devolver un vector correspondiente a la fila i de la matriz m .

Referencias

1. Leyes de los Exponentes. (s.f.). Kentucky Center for Mathematics.
https://kcm.nku.edu/tutorials/leyes_de_exponentes/explaws_spa.xhtml
2. Rodó, P. (2019, 3 de noviembre). Logaritmo | Economipedia. Economipedia.
<https://economipedia.com/definiciones/logaritmo.html>
3. Series. (s.f.). PAPIME. <http://gmc.geofisica.unam.mx/papime2020/index.php/articulos/9-series>
4. Libretexts. (2022, 30 de octubre). 5.7: Aritmética Modular. LibreTexts Español.
[https://espanol.libretexts.org/Matematicas/Combinatoria_y_Matematicas_Discretas/Un_libro_de_trabajo_en_espiral_para_matemáticas_discretas_\(Kwong\)/05:_Teoría_básica_de_números/5.07:_Aritmética_Modular](https://espanol.libretexts.org/Matematicas/Combinatoria_y_Matematicas_Discretas/Un_libro_de_trabajo_en_espiral_para_matemáticas_discretas_(Kwong)/05:_Teoría_básica_de_números/5.07:_Aritmética_Modular)
5. Recursividad en programación. (s.f.). <https://www.netmentor.es/entrada/Recursividad-programacion>
<https://www.netmentor.es/entrada/Recursividad-programacion>
6. Constructores (C++). (s.f.). Microsoft Learn: Build skills that open doors in your career.
<https://learn.microsoft.com/es-es/cpp/cpp/constructors-cpp?view=msvc-170>
7. Estructura de Datos : Vectores. (s.f.). Sitio Oficial de la F C E I A.
<https://www.fceia.unr.edu.ar/estruc/2006/vector.htm>
8. Castro, S. (2023, 10 de enero). ¿Qué es un string en programación? Fundamentos de la Programación.
<https://www.fundamentosprogramacion.site/2023/01/que-es-un-string-en-programacion.html>
9. Punteros (C++). (s.f.). Microsoft Learn: Build skills that open doors in your career.
<https://learn.microsoft.com/es-es/cpp/cpp/pointers-cpp?view=msvc-170>
10. Referencias (C++). (s.f.). Microsoft Learn: Build skills that open doors in your career.
<https://learn.microsoft.com/es-es/cpp/cpp/references-cpp?view=msvc-170>
11. Plantillas (C++). (s.f.). Microsoft Learn: Build skills that open doors in your career.
<https://learn.microsoft.com/es-es/cpp/cpp/templates-cpp?view=msvc-170>
12. González, J. D. M. (2018, 20 de marzo). Matrices en C++. Declaración, uso y sintaxis en C++. Aprende a programar desde cero en múltiples lenguajes. <https://www.programarya.com/Cursos/C++/Estructuras-de-Datos/Matrices>