

Universidad del Valle de Guatemala
Facultad de ingeniería
Algoritmos y Estructuras de Datos
Grupo 27



Proyecto #1

Nelson Eduardo García Bravatti 22434
Oscar Andres Fuentes Dubón 22763
Joaquín Puente Grajeda 22296

Investigación Java Collections

Java Collections Framework es un conjunto de clases e interfaces que proporcionan métodos estandarizados para trabajar con colecciones de objetos. Incluye una amplia variedad de estructuras de datos como listas, conjuntos y mapas, así como varios algoritmos para manipular estas estructuras de datos. Algunos de los temas importantes relacionados con el sistema de recopilación de Java son:

Interfaces: el marco de la colección define varias interfaces para representar diferentes tipos de colecciones, como listas, conjuntos y mapas. Estas interfaces definen las operaciones básicas que se pueden realizar en las colecciones, como agregar y eliminar elementos, e implementan las diversas clases proporcionadas por el marco. **Clases:** el marco proporciona varias clases que implementan varias interfaces, como ArrayList, HashSet y HashMap. Estas clases son útiles para crear y manipular colecciones, y se pueden adaptar de muchas maneras para satisfacer diferentes necesidades. **Algoritmos:** el marco de la colección incluye una variedad de algoritmos que se pueden usar para realizar operaciones comunes en las colecciones, como ordenar y buscar. Estos algoritmos se implementan como métodos estáticos en la clase Collections y se pueden usar para realizar operaciones en cualquier colección que implemente la interfaz correspondiente. **Datos genéricos:** el marco de colecciones utiliza datos genéricos para definir colecciones de forma segura. Esto significa que se puede definir una colección para que contenga elementos de un determinado tipo, y el compilador verificará que solo se agreguen elementos de ese tipo a la colección. Para usar Java Collections Framework, debe importar las clases e interfaces requeridas. Luego puede crear instancias de clases que implementen la interfaz y usarlas para manipular colecciones de datos.

Investigación Lisp

Lisp es una familia de lenguajes de programación con una larga historia y una amplia gama de funciones y aplicaciones. Fue desarrollado por John McCarthy en la década de 1950 y desde entonces se ha utilizado en una variedad de campos. Una de las características más notables de Lisp es el uso de listas como estructura básica de datos. Proporciona una gran flexibilidad de representación de datos y fácil manipulación de datos. Lisp también es un lenguaje de programación funcional, lo que significa que se enfoca en manejar datos y funciones como entidades separadas. Esto contrasta con la programación orientada a objetos, que se centra en objetos con estado y comportamiento. En términos de aplicaciones prácticas, Lisp se ha utilizado ampliamente en inteligencia artificial, robótica, programación de sistemas integrados, creación de sistemas automáticos de procesamiento de lenguaje natural y optimización de sistemas económicos. En particular, Lisp se usa ampliamente en la investigación académica sobre inteligencia artificial debido a su capacidad para manejar fácilmente estructuras de datos complejas y su enfoque en la programación funcional. La programación funcional se enfoca en manejar datos y funciones como una entidad separada, en comparación con la programación orientada a objetos, que se enfoca en manipular objetos con estado y comportamiento. Si bien ambos métodos tienen sus usos, la programación funcional generalmente se considera más fácil de depurar y escalar, y tiene mayores capacidades de paralelización, lo que la hace adecuada para sistemas altamente concurrentes. En resumen, Lisp es un lenguaje de programación único y versátil con una larga historia y una amplia gama de funciones y aplicaciones. Su enfoque en la programación funcional lo diferencia de otros lenguajes de programación y lo hace adecuado para una variedad de propósitos, especialmente en áreas relacionadas con la inteligencia artificial.