RESUMEN

La definición formal de requisitos de software usando especificaciones algebraicas tiene todas las ventajas de las especificaciones formales y su sólida base teórica. Este tipo de especificaciones es generalmente textual. La mayor parte de los lenguajes de especificación modernos tienen una representación gráfica para mejorar su usabilidad. Esto también es el caso de las especificaciones algebraicas. En este artículo presentamos una recopilación de las formas en que los tipos abstractos de datos pueden ser representados gráficamente usando especificaciones algebraicas, proponiendo una notación que incluye el conjunto de todas las facetas encontradas en la literatura. También mostramos un ejemplo de aplicación y algunos resultados experimentales de usar esta notación gráfica en la práctica.

Palabras Claves: Ingeniería de software, métodos formales, especificación algebraica, tipos abstractos de datos.

ABSTRACT

Formally specifying software requirements using algebraic specifications has all the advantages of formal specifications. This type of specifications is usually textual. Most modern specification languages have a graphical representation in an attempt to improve usability. This is also the case for algebraic specifications. Here we present a survey on how abstract data types are represented graphically. We propose a structure containing a superset of all elements surveyed. We also show an application example, and we report some experimental results when using this graphical representation.

Keywords: Software engineering, formal methods, algebraic specification, abstract data types, graphic language.
REFERENCES


Recibido el 5 de diciembre de 2003, aceptado el 7 de mayo de 2004
In computer science, an abstract data type (ADT) is a mathematical model for data types, where a data type is defined by its behavior (semantics) from the point of view of a user of the data, specifically in terms of possible values, possible operations on data of this type, and the behavior of these operations. This contrasts with data structures, which are concrete representations of data, and are the point of view of an implementer, not a user. Abstract Data Type - msit. agenda. abstraction real time example definition of abstract data type difference between abstract data type and data structure different adts dictionary adt operations in dictionary adt. abstraction. hiding. Introduction to Abstract Data Type & C++ Revision - Introduction to abstract data type & C++ revision. data structure & algorithm part ii. objectives. at the end of the class students are expected to: understand abstract. Data Abstraction -. chapter 9. outline. concept of abstraction abstract data types with example encapsulation with example. the co Research in the area of abstract data types started about 20 years ago. Since then there has been continuous activity with strong influence both on the applications and the theoretical foundations of methodologies for software design. The Ninth Workshop on Specification of Abstract Data Types was held jointly with the Fourth COMPASS Workshop in Spain in 1992. The main topics covered were: object-oriented specifications, rewriting methods, specification languages and associated tools, type systems, and algebraic specification of concurrency. This volumes contains four invited papers presented a