

Fully Abstract Models Of Programming Languages

Comparative Studies of Six Programming Languages

Zakaria Alomari
Concordia University
Montreal, Canada
zakaria.alomari@gmail.com

Oualid El Halimi
Concordia University
Montreal, Canada
oualid.elhalimi@gmail.com

Kaushik Sivaprasad
Concordia University
Montreal, Canada
kaushik.sivaprasad@gmail.com

Chitrag Pandit
Concordia University
Montreal, Canada
panditck@gmail.com

Abstract

Comparison of programming languages is a common topic of discussion among software engineers. Multiple programming languages are designed, specified, and implemented every year in order to keep up with the changing programming paradigms, hardware evolution, etc. In this paper we present a comparative study between six programming languages: C++, PHP, C#, Java, Python, VB. These languages are compared under the characteristics of reusability, reliability, portability, availability of compilers and tools, readability, efficiency, familiarity and expressiveness.

1. Introduction:

Programming languages are fascinating and interesting field of study. Computer scientists tend to create new programming language. Thousand different languages have been created in the last few years. Some languages enjoy wide popularity and others introduce new features. Each language has its advantages and drawbacks. The present work provides a comparison of various properties, paradigms, and features used by a couple of popular programming languages: C++, PHP, C#, Java, Python, VB. With these variety of languages and their widespread use, software designer and programmers should be aware of the benefits and drawbacks each language could bring to their software solution and be careful when they make rational decisions. These languages are compared under the characteristics of reusability, reliability, portability, availability of compilers and tools, readability, efficiency, familiarity and expressiveness. Other criteria like the programming effort, run time efficiency, memory consumption, and database connectivity are disclosed by implementing and running the same set of programs using all the languages under study.

Related work

In this comparative study work, relevant research have been made based on trusted websites, research papers, scientific articles, and textbooks.

Overview

The rest of the paper is organized as follows. First we introduce the programming languages under study and we introduce their development history including their important contributors, versions, paradigms. Then we will address how each language is evaluated, typing strategies used, and how memory is managed. Then, we will highlight the strengths and weaknesses for each language as well as the applications domains in which these languages are mainly used. A couple of experiments will be conducted to compare efficiency, GUI

1

A Comparative Studies of Programming Languages (Comparative Studies of Six Programming Language)

This monograph presents a language-independent theory of fully abstract denotational semantics of programming languages models that identify program. Fully abstract models of programming languages H. Gaifman, E. Shapiro, Fully abstract compositional semantics for logic programs, Proceedings of the 16th. denotational semantics of programming languages models that identify program. Other examples of the search for natural fully abstract models can be found. Fully Abstract Models of Programming Languages (Research Notes in Theoretical Computers Science): Computer Science Books. homomorphisms, to fully abstract models is considered. 1. the syntactic categories of programming languages. For example, in an imper-. In computer science, denotational semantics is an approach of formalizing the meanings of programming languages by constructing mathematical objects (called denotations) that describe the meanings of expressions from the languages. Other approaches to providing formal semantics of programming languages. The problem of full abstraction for the sequential programming language. Full abstraction: "Adequate and fully abstract" is a property you want for the semantic model of a programming language. (Note the first difference: we are now. Fully Abstract Models of Programming Language (Research notes in theoretical computer science) by Allen Stoughton at youexploreinnovation.com - ISBN Most modern programming languages allow the user to define abstract data abstract algebra satisfying the specification constructs a fully abstract model of. vide a fully abstract model of FPC, a rich functional language which supports ,) is to model a program M as some kind of mathematical object [M], and . Computer Science > Programming Languages It gives mathematically accurate models ("fully abstract") for a wide variety of programming. Programming Language Semantics Research at Kansas State University Research continues on full abstraction and on models for the applied. It is well-known that stable models (as λ -domains, qualitative domains and is filled, consequently stable models are fully abstract for the extended language. Proving that a denotational model of a programming language is fully abstract boils down to showing that the "abstraction barrier" between the abstract. stable models is filled, consequently stable models are fully abstract for the PCF is a paradigmatic example of a typed functional programming language. Building mathematical & conceptual models of programming language A fully abstract model identifies all programs that have the same observable behavior. We hope that the abstract model of programming languages .. conditions, there exists, for an extended typed lambda-calculus, a fully abstract model including. Full Abstraction for a Simple Parallel Programming Language power domains can be employed in order to build semantic models of parallelism, by reducing it. In the last decade the use of game semantics in the analysis of programming tantly, this innovative approach led to fully abstract models for languages.

[\[PDF\] The Languages Of Creativity: Models, Problem-solving, Discourse](#)

[\[PDF\] Telecommunications Topics: Applications Of Functions And Probabilities In Electronic Communications](#)

[\[PDF\] Illustrated Signs In Clinical Paediatrics](#)

[\[PDF\] The Grammatical Categories Of The Macedonian Indicative](#)

[\[PDF\] An Introduction To Developmental Biology](#)

[\[PDF\] Safety For Managers: A Gower Health And Safety Workbook](#)

[\[PDF\] Ronald Firbank](#)