Design a programming language for the problem domain of Scientific Computations

SPL (Scientific Programming Language)

This essay is written in order to create a language tailored to the scientific computations problem. Scientific computing is a methodology incorporating numerical analysis, physical knowledge, the development of algorithms, and formal programming. On the world's largest machines, many yottacycles are used to model problems as diverse as weather forecasts, the properties of material composites, the behaviour of biomolecules in solution, and the quantity of chemical compounds (Knepley, 2012). These languages include java, COBOL, FORTRAN, Pascal, Python , PHP, C / C++ etc. Language has its own features and characteristics, methods of syntaxing, compiling and translating, as well as managing and maintenance exceptions. So, I will create a new language here that will be used for the scientific computations issue domain. The language is SPL (Scientific Programming Language), the design purpose, functionality, attributes, compilation and description, articulation, type checking, orthogonality, functionality, types of data and abstraction support, readability, writing and usability, etc. Therefore, in this I will write an essay creating a new language for the specific problem of scientific computations.

Introduction

This essay is written in order to develop a language suitable for scientific computations problem area. Scientific computing is a set of computational techniques and techniques used with a computer system to build and solve mathematical models of a number of scientific problems. Numerical linear algebra is one of the most common methods of quantitative mathematical computation(Michailidis & Margaritis, 2016). Matrix computations (for example, the addition of vectors and matrices, point product, external product, transpose matrix, matrix vector function and matrix function) and linear systems solutions (for example the direct Gaussian elimination technique and the iterative Jacobi method) that are used as the basis of a number of computing problems , for example computational statistics (Michailidis & Margaritis, 2016). One of the most important type of scientific computation is the computational linear algebra. In numerical linear algebra the fundamental computations are matrix computations and solution of linear systems. In certain computing problems these computations are used as kernels(Michailidis & Margaritis, 2016).

The computer programmes are designed with different programming languages. The programming language is essentially an artificial language programmed or implemented for the use of a computer, or programming language is, in other words, a set of rules, conditions and symbols used to build a machine. Programming languages could be used to build computer programmes to control the machine 's behaviour. These programming languages are the high-level languages. The languages of 1st generation (1GL), 2nd generation (2GL), 3rd generation (3GL), 4th generation (4GL), 5th generation (5GL), low programming and high programming languages are included. Most languages including Java , C++, C, JavaScript, APL, FORTRAN and Pascal are available. Could language has its own programming instructions syntax. Each language has its own functionality, output, coding, method of implementing, compilation and interpretation.

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/

Any of those languages now allow the abstraction of data types, some of which use static and dynamic scopes, and are highly typed and loosely or weakly typed. Thus, the nature and features of and programming language in its field are special, as the languages are used to build programmes in computation, financial, educational, computer and artificial intelligence fields. We will now use a syntax in this essay that is used in the Scientific computation problem domain. This language is an Scientific Programming Language SPL . It has its own interface and paradigms of programming. The programming languages are distinct from the current ones.

By using imperative languages, such as C and Fortran, the programmer specifically tells the machine what to do with each move. The calculation is based on the input values of other variables which carry values, and which measure the variable value. For example, arithmetical functions, such as division and linear algebraic functions, such as matrix multiplication, are important functions in scientific computing. Imperative languages have the key benefit over simplistic frameworks like Excel because they are able to integrate these basic elements flexibly.

Purpose

The SPL language is a programming language in the Web. This language is a general term. In particular, there are vast quantities of processing data and problems in all scientific computing fields, such as scientific computing and educational institutions. In scientific computing, therefore, there are several questions. This language can be structured to fit into the scientific computing problem field. This language is simple, and it uses Procedural, OOP and parallel processing and the data types as well. This language can be used conveniently for computer science applications. This language also offers data structuring and abstraction support. This language is therefore built only for the problems of science computing and mathematical implementation.

Features

The features of the language SPL (Scientific Programming Language) are as follows:

- It uses syntax elements such as words, which include user-defined identifiers and labels.
- It uses as much as appropriate English syntax.
- It uses approximately 200 reserved words.
- The language is strong.
- Scientific Programming Language as the name suggests is a language that is computation-oriented so it can be used for commercial applications such as research and educational institutions.
- This language is also a basic language for personal computers and other systems.
- This language has the potential to apply quickly.
- The visual programming environment is provided in this language. The atmosphere in which programmes are developed and tested is the programming environment. So, the object and the visual programming environment are used in this language.
- The trace functionality of this language is execution.
- This is a language that is clearly defined.
- There are several computation data with formula predefined in this language.

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/

Thus, above all, this language (SPL) is characteristic. This language is also known as an imperative language and those languages are dominated by the programming paradigm of procedural and procedural. Thus, these are all characteristics of this language and all these characteristics make it ideal in the scientific field of calculations.

Characteristics

1. The main characteristic of this language is that it is very easy to code and understand.

2. It holds all the data of the computation such as data of accounts, customer's related data, deposits, withdrawals and data of transfers. So, this language can holds the large processing data of the bank.

3. The language is well-suited for the problem domain of scientific computations.

4. This language is a simple language.

5. This language takes less time for the execution.

6. This language providing the maintainability of the code.

7. This language providing the readability.

8. This language providing the automatic memory management.

9. This language is a computation-oriented language.

These all are the characteristics of this language IBPL (Internet Business Programming Language).

INTERPRETATION/COMPILATION METHOD:

Compilation is the process to convert the programs that are written in high-level languages into the equivalent programs that are written in the machine language. It means the high-level language convert into the machine code or the machine language.

This language is a high-level language so this language can either interpret or compiled. This Internet Business Programming Language must be converted into the machine language or machine code through the compiler. Then the source program code is run through the compiler. The compiler performs the type checking. Also the compiler checks the syntax error and other errors in the program source code and then it is converted into machine language that a computer can understand. Because computers can only understand the machine language and it is a language of binary stream of 0s and 1s. Now after converting the source program code into the machine code the compiler generates the output file for the output of the program. This output file named as load module and this output file stores the executable code in the form of the machine language i.e. 0s and 1s. When the compiler translates the source code to machine language then the machine code is save in executable files. So, the source code instructions are translate into machine code when the program runs. The repetition of the source code does not require the again translation of the source code into the machine code. By this, the language have the faster speed to execute or for the compilation. If the compiler finds the errors in your program, then it is important for us to correct or edit

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/

these errors in the editors and again re-compile and then again checks the errors. So, it is English-like language and its compilation is very easy and its execution is very fast. Hence it provides the greater performance, readability and writability of the source code.

MEMORY MANAGEMENT:

Memory management means to manage the computer memory at the system level. This Internet Business Programming language provides the automatic memory manage in the system. In the banking this language can automatically manage the whole data of the accounts, customers, deposits, withdrawals and transfers. This language can manages this data according to the memory capacity. This language is free to reuse. The memory management is necessary for every language and in every system. This memory management provide the dynamically allocation of memory to the programs that are run in the banking systems. This language has garbage collection and for the memory management it has the memory pool system. This language also supports the abstraction through the garbage collection.

SPECIFICATIONS:

SIMPLICITY:

The Internet Business Programming Language provides simplicity. It means it provides clear, simple and unified sets of concepts to the programming. This language uses simple readable concepts in the banking problem. This language has the feature of simplicity because it is very readable and also easy to understand. This language can easy to spot and also finding programming errors. This language is created with fewer features as comparative to the existing programming languages. There is a limited use of primitive data types in this Internet Business Programming Language. This language also has the lack of user-defined data types that make this language to more simple. So, all these features provide the simplicity to this language a comparative to the other languages.

ORTHOGONALITY:

In programming languages Orthogonality means if a language supports more than one feature than it must have an attributes to combine all possible combination with every combination being meaningful and when the features of the language are orthogonal then the language becomes easier to learn and programs are easier to write.

The orthogonality is directly related to the simplicity and the readability of the programming language. So, this Internet Business Programming Language (IBPL) can be considered as the orthogonal language. By orthogonality the language becomes easier to learn and write because there are fewer exceptions and special cases to remember.

DATA TYPES:

A class of data objects with the set of operations for creating and manipulating them is called a data type. The set of primitive data types are built into the language that every language has. This language uses the alphanumeric, alphabetic data types, primitive data types and uses the elementary data types such as integer, character and Boolean etc. So, the data types indicate the classes of objects, the meaning, values, operations and functions associated with it. The

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/

alphanumeric characters represent the characters and numeric data types represent the numbers. So, these are the data types that are used within this language.

SYNTAX DESIGN:

The syntax design of a programming language means how a program looks like. It includes rules of a language to define different construct of that languages. The rules include such as how statements, declarations and other language constructs are written and perform. Here the semantic is also used to design a new language. The semantics of a programming language is the meaning or the logical units. The syntax of this language is very easy. This syntax gives the important information to understand the program that is used for the banking problem and it provides easily translation or compilation of the source code program into the machine code or the machine language.

TRANS_DEPOSIT (“Customer Account Number”)

{

Check account number and name of the customer

Deposit Money in the account

}

TRANS_ WITHDRAWAL (“CUSTOMER Account Number”);

{

Check balance

Check account number

If valid then withdrawal from account

}

SUPPORT FOR ABSTRACTION:

Abstraction means hiding the details of the abstract data types from the users. In other words, abstraction hiding the non-required details or information while using the important details. Abstraction can be control and data depending process. The programming language provides direct support for the abstraction but the concept is not present in the language. For example in banking if we use the subprogram of withdrawals to use some difficult task such as managing the accounts. Then if subprogram is not used then it will not allow the reader to read or write the code easily. As a result this Internet Business Programming Language supports for the abstraction through the garbage collection and by using subprograms.

EXPRESSIVITY:

Expressivity means that how much a programming language has the power to express something in the language more clearly.

A programming language has the following capability to express the solution of the problem:

• In original problem formulation.

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/

• In a clear and simple way.

• In the way of other solved problems.

The Internet Banking Programming Language is a procedural language and it has different expressive power than the other existing programming languages. To increase the expressivity of any programming language is that adding the more features to it. By doing this the expressivity of the programming language can be increased. The expressivity of imperative and declarative programming languages is different. Boolean types support languages have more expressive power because they are express in the form of true or false.

TYPE CHECKING:

Type checking is the process by which an operation is checked for its argument list and argument data types. It means it checks that each operation receives the proper number of arguments of the proper data type. Type checking checks the errors in the program either by a compile-time or at a run-time. The static type checking is done at the compile-time or the dynamic type checking is basically done at the run-time. The programming languages use the static type checking because it checks the errors during the compile-time. This Internet Business Programming Language also used the static type checking because dynamic type checking requires extra storage.

EXCEPTION HANDLING:

The exceptions are defined as the event or conditions that occur during the execution of the program or at the run-time. In other words, to find the run-time errors is called exception. So the event and the condition are called exception and which perform the special processing or take corrective measures is called exception handler. There are some exceptions which are predefined in the language and some exceptions are programmer defined. This Internet Business Programming Language uses the programmer-defined exceptions i.e. Underflow: exception or Overflow: exception. The language uses the try catch throw method for the exception handling.

RESTRICTED ALIASING:

Aliasing is defined as data object which is visible through more than one name in a single referencing environment. It means the two methods to reference single memory location. The data object has multiple names but having a unique name in each referencing environment does not arise any problem but if the same data object using the different names in the same referencing environment then the problems are arise. It is a dangerous feature for all programming languages. This language (IBPL) does not provide aliasing. The aliasing should be restricted in every programming language.

READABILITY:

Readability means the written text can be understood by the user or by the programmer easily. Readability is provided the ease to the users and the readers. This language is easy to code and understand. This Internet business Programming Language (IBPL) uses the

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/

primitive data types and it also has the lack of user-defined data types. This language is also a simple language and English like language so it can be easy to readable. The keywords that we are using in this language they all show their functionality in the language. So this is the main feature of this language to make it readable.

WRITABILITY:

By the use of concise and regular syntactic structures writability is enhanced. Writability is defined as the ease with which the language writes the program or code. The writability can be enhanced by using the small numbers of statements and constructs. When the control structures are used in this language to write code then it is the difficult time for the writability of the language. If the redundancy is used in the language then it is harder to write the code of this language. The writability of this language depends on the factors of simplicity and the orthogonality, expressivity and supports for the abstraction.

RELIABILITY:

If a language performs according to its specification using all the conditions in every situation then the language is said to reliable. This language has the following factors of reliability:

• Type checking

• Exception handling

• Aliasing

At the compile time, this language performs the type checking and size checking. This language uses static type checking at the compile time. The type checking and exception handling enhance the reliability of this IBPL language.

CONCLUSION:

This essay represents a new language for the problem domain of banking. The new design language is IBPL i.e. Internet Business Programming Language. The banking is concern with the business. By deposit the money of the customers the bank provides the secure services to the customers. There are many programming languages that are used for the business applications of computers. So, business includes the areas such as banking and finance institutions etc. In this essay I have designed a new language for the banking problem. This Internet Business Programming Language has its own features, syntax and characteristics etc. which are different from the existing language. So, this language enhances the readability because it is a simple language and easy to read and understandable by the reader. It also provides the writability because of simplicity and orthogonality, expressivity and supports of abstraction factors and this language also enhances the reliability by using type checking and exception handling factors. As a result, this shows that it is a very good programming language for the business applications mainly for the banking problem.

References

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/

Knepley, M. G. (2012). Programming Languages for Scientific Computing. doi:10.1007/978-3-540-70529-1

Michailidis, P. D., & Margaritis, K. G. (2016). Scientific computations on multi-core systems using different programming frameworks. *Applied Numerical Mathematics*, *104*, 62-80. doi:https://doi.org/10.1016/j.apnum.2014.12.008

Introduction of Programming Paradigms - GeeksforGeeks. (2018). Retrieved 11 September 2020, from https://www.geeksforgeeks.org/introduction-of-programming-paradigms/

This study source was downloaded by 100000831375417 from CourseHero.com on 08-14-2022 05:47:41 GMT -05:00

https://www.coursehero.com/file/92408256/Language-Design-Essaydocx/