|  |  |
| --- | --- |
| 3. Machine language and programming languages. | |
| Machines understand 0s and 1s. The task of software is to express computation in a higher-level language and then translate it into a sequence of 0s and 1s that machines can understand. When you express the computation in a higher-level language this is referred to as raising the level of abstraction in programming context. A typical software application, such as a word processor or an operating system, may include millions of lines of software code. However, the hardware can only execute low-level instructions presented to it in the form of the machine language consisting of 0s and 1s. Several layers of software are needed to convert the high-level application code into the machine language. As shown in the figure below, a system software layer such as the operating system controls the hardware; the user controls application programs, which run on top of the operating system layer.  During the 1950s and 1960s, several high-level programming languages were introduced, such as FORTRAN, COBOL, Lisp, ALGOL, and PL/I. The 1970s saw the introduction of languages such as Pascal, C, and Prolog. Most of these languages such as Pascal, C, and FORTRAN are procedural programming languages. Procedural programming is also referred to as imperative programming. The basic idea of this programming style is to specify the steps that the program must take to reach the desired state.  An exception to this style of programming comes from languages called Lisp and Prolog. Lisp is a functional programming language, and Prolog is a logic programming language. In functional programming style, evaluation of mathematical style functions take place and the state variables are not used. In logic programming, a program is expressed as a sequence of logical assertions and these assertions are automatically evaluated to come up with the result. These languages include a logic evaluation engine along with the compiler.  In the late 1970s, a new design approach called object-oriented programming (OOP) was developed. This programming technique has several advantages, and almost all languages today follow this approach. The 1980s saw the introduction of object-oriented programming language C++, which was later followed by another objective-oriented programming language for the development of mobile and web-based application, and Java will be the focus of this course. The 1980s and the 1990s also saw the introduction of scripting programming languages such as Perl, Python, and Ruby. | Машины понимают 0 и 1. Задача программного обеспечения состоит в том, чтобы выразить вычисления на языке более высокого уровня, а затем преобразовать их в последовательность нулей и единиц, которую могут понять машины. Когда вы выражаете вычисления на языке более высокого уровня, это называется повышением уровня абстракции в контексте программирования. Типичное программное приложение, такое как текстовый процессор или операционная система, может включать миллионы строк программного кода. Однако аппаратное обеспечение может выполнять только низкоуровневые инструкции, представленные ему в виде машинного языка, состоящего из 0 и 1. Для преобразования высокоуровневого кода приложения в машинный язык требуется несколько слоев программного обеспечения. Как показано на рисунке ниже, уровень системного программного обеспечения, такой как операционная система, управляет оборудованием; пользователь управляет прикладными программами, которые работают поверх уровня операционной системы.  В 1950-х и 1960-х годах было представлено несколько языков программирования высокого уровня, таких как FORTRAN, COBOL, Lisp, ALGOL и PL/I. В 1970-е годы появились такие языки, как Pascal, C и Prolog. Большинство из этих языков, таких как Pascal, C и FORTRAN, являются языками процедурного программирования. Процедурное программирование также называют императивным программированием. Основная идея этого стиля программирования состоит в том, чтобы указать шаги, которые программа должна предпринять, чтобы достичь желаемого состояния.  Исключением из этого стиля программирования являются языки Lisp и Prolog. Lisp — это функциональный язык программирования, а Prolog — язык логического программирования. В стиле функционального программирования имеет место вычисление функций математического стиля, а переменные состояния не используются. В логическом программировании программа выражается в виде последовательности логических утверждений, и эти утверждения автоматически оцениваются для получения результата. Эти языки включают механизм логической оценки вместе с компилятором.  В конце 1970-х годов был разработан новый подход к проектированию, называемый объектно-ориентированным программированием (ООП). Этот метод программирования имеет несколько преимуществ, и сегодня почти все языки следуют этому подходу. В 1980-х годах появился объектно-ориентированный язык программирования C++, за которым позже последовал другой объектно-ориентированный язык программирования для разработки мобильных и веб-приложений, и Java будет в центре внимания этого курса. В 1980-х и 1990-х годах также появились языки программирования сценариев, такие как Perl, Python и Ruby. |