

ЛЕКЦИЯ 01

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

АЛГОРИТМИЗАЦИЯ И
ПРОГРАММИРОВАНИЕ



ИСТОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

- 1843 год – Ada Lovelace изобретает первый машинный алгоритм
- Ранние 1940-е годы – Conrad Zuse изобретает первый высокоуровневый язык программирования Plankalkul (запланированные вычисления)
- 1949 год – появляется низкоуровневый язык Assembler, использованный для программирования Electronic Delay Storage Automatic Calculator. В это же время John McCauley представляет миру Shortcode
- 1952 год - Alick Glennie изобретает Autocode

ИСТОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

- 1957 год - John Backus изобретает FORMula TRANslation (FORTRAN), который активно используется для решения задач в науке, математике и в статистике.
- 1958 год – появляется Algorithmic language (ALGOL). Позже этот язык ляжет в основу таких языков программирования, как C, C++, Java, Pascal. В это же время John McCarthy создает List Processor (LISP), который используется для написания систем искусственного интеллекта и актуален сегодня на уровне Ruby и Python
- 1959 год - Dr. Grace Murray Hopper изобретает Common Business Oriented Language (COBOL). Язык используется в программировании банковских систем, ATM, сотовых телефонов.
- 1964 год – студенты Dartmouth College создают Beginner's All-Purpose Symbolic Instruction Code (BASIC) для упрощения решения задач по программированию и математике. Позже Paul Allen и Bill Gates сделают этот язык компилируемым.

ИСТОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

- 1970 год - Niklaus Wirth создает PASCAL. Язык назван в честь известного математика Blaise Pascal.
- 1972 год – Alan Kay, Dan Ingalls, и Adele Goldberg изобретают Smalltalk для помощи программистам в написании программ. Это событие станет причиной популярности таких языков, как Java, Ruby и Python. Dennis Ritchie в это же время создает язык C и пишет на нем операционную систему Unix. Donald Chamberlain и Raymond Boyce параллельно создают Structured Query Language (SQL).
- 1980-е год - Jean Ichbiah создает язык программирования Ada, названный в честь Ada Lovelace. Этот язык используется для создания и поддержки систем управления авиаперелетов в Бельгии, Германии и Австрии.
- 1983 год – Bjarne Stroustrup дорабатывает язык C и называет его C++, добавив в него такие важные концепции, как классы, шаблоны и виртуализацию. Язык становится очень популярным и крупные компании начинают его использовать для написания своих продуктов, например Adobe использовали C++ для создания известного Adobe Photoshop. Вместе с тем, Tom Love и Brad Cox создают Objective-C, который будет использован для создания компанией Apple таких операционных систем, как iOS и macOS

ИСТОРИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

- 1987 год - Larry Wall изобретает Perl
- 1990 год – появляется Haskell, названный в честь известного математика Haskell Brooks Curry. Активно используется для решения бизнес-задач и в видеоигровой индустрии
- 1991 год – Microsoft выпускает свой собственный язык программирования Visual Basic, который значительно упрощает разработку программ под операционные системы Microsoft и создание пользовательского интерфейса в них. В это же время Guido Van Rossum выпускает Python, что сильно меняет индустрию программирования.
- 1993 год – Yukihiro Matsumoto изобретает Ruby. Brendan Eich изобретает JavaScript. Появляется Hypertext Preprocessor (PHP)
- 2000 год – Microsoft выпускает C#
- 2003 год - Martin Odersky создает Scala. Bob McWhirter и James Strachan создает Groovy
- 2009 год - Google запускает Go для упрощения разработки и поддержки крупных программных проектов
- 2014 год - Apple изобретает Swift, чтобы заменить Objective-C, C++, и C

СТРУКТУРА ЯЗЫКА ПРОГРАММИРОВАНИЯ



Алфавит

Синтаксис

Семантика

СТРУКТУРА ЯЗЫКА ПРОГРАММИРОВАНИЯ

Алфавит - это набор символов, которые можно применять в инструкциях языка программирования. Другие символы допустимы только в особых случаях, например в строковых константах

СТРУКТУРА ЯЗЫКА ПРОГРАММИРОВАНИЯ

Алфавит - это набор символов, которые можно применять в инструкциях языка программирования. Другие символы допустимы только в особых случаях, например в строковых константах

Синтаксис языка определяет правила построения операторов.

СТРУКТУРА ЯЗЫКА ПРОГРАММИРОВАНИЯ

Алфавит - это набор символов, которые можно применять в инструкциях языка программирования. Другие символы допустимы только в особых случаях, например в строковых константах

Синтаксис языка определяет правила построения операторов.

Семантика – это смысловое содержание операторов языка программирования. Семантические правила определяют действия, описываемые различными операторами, и, в итоге сущность всего алгоритма.

ОБЛАСТИ ПРИМЕНЕНИЯ

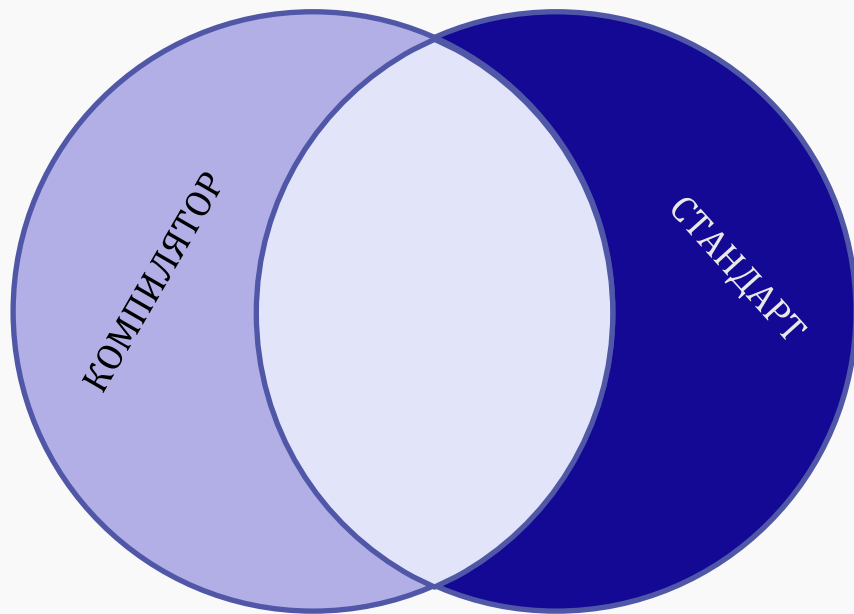
- научные вычисления (языки **C++**, **FORTRAN**, **Java**);
- системное программирование (языки **C++**, **Java**);
- обработка информации (языки **C++**, **COBOL**, **Java**);
- искусственный интеллект (**LISP**, **Prolog**, **Julia**);
- издательская деятельность (**Postscript**, **TeX**);
- удаленная обработка информации (**Perl**, **PHP**, **Java**, **C++**);
- описание документов (**HTML**, **XML**).

СТАНДАРТЫ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Для того чтобы *компиляция (интерпретация)* одной и той же программы различными *компиляторами (интерпретаторами)* всегда давала одинаковый результат, разрабатываются **стандарты языков программирования**.

Однако, не все компиляторы соблюдают стандарт по некоторому ряду причин.

СТАНДАРТЫ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ



ВИДЫ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

ВИД	ПРИМЕРЫ	ОПИСАНИЕ
Императивные	<i>C, C++, Java, Python</i>	Описывают цепочку операций, которые ПК должен сделать для решения.
Декларативные	<i>SQL, Prolog, Haskell</i>	Фокусируются на описании желаемого результата.
Скриптовые	<i>JavaScript, Ruby, Perl</i>	Предназначены для написания скриптов, которые выполняются интерпретатором или виртуальной машиной
Функциональные	<i>Lisp, Haskell, Erlang</i>	Функции как созидательный материал и квалифицируются неизменяемостью данных
Объектно-ориентированные	<i>Java, C++, C#, Python</i>	Способствуют формированию модульного, повторяющегося и расширяемого кода

КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

По уровню абстракции:

- Низкоуровневые (*Assembler*)
- Среднеуровневые (*C, C++*)
- Высокоуровневые (*Python*)

КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

По способу выполнения:

- компилируемые (*Assembler, C, C++*)
- интерпретируемые (*Python*)

КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

По типизации данных:

- Статическая типизация (*Assembler, C, C++*)
- Динамическая типизация (*Python*)

КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

По системе типов:

- слабая типизация (*Assembler, C, C++*)
- сильная типизация (*Python, Ada*)

КОМПИЛЯТОР VS ИНТЕРПРЕТАТОР

Компилятор — программа или техническое средство, выполняющее компиляцию.

Компиляция — трансляция программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду (абсолютный код, объектный модуль, иногда на язык ассемблера). Входной информацией для компилятора (исходный код) является описание алгоритма или программа на предметно-ориентированном языке, а на выходе компилятора — эквивалентное описание алгоритма на машинно-ориентированном языке (объектный код)

КОМПИЛЯТОР VS ИНТЕРПРЕТАТОР

Интерпретатор — программа (разновидность транслятора), выполняющая интерпретацию.

Интерпретация — пооператорный (покомандный, построчный) анализ, обработка и тут же выполнение исходной программы или запроса (в отличие от компиляции, при которой программа транслируется без её выполнения).

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Дорохова Т.Ю., Основы алгоритмизации и программирования : учебное пособие для СПО / Т.Ю. Дорохова, И.Е. Ильина. – Саратов, Москва : Профобразование, Ай, Пи Ар Медиа, 2022. – 139 с.
2. Кудинов Ю.И., Основы алгоритмизации и программирования : учебное пособие для СПО / Ю.И. Кудинов, А.Ю. Келина. – 2-е изд. – Липецк, Саратов: Липецкий государственный технический университет, Профообразование, 2020. – 71 с.