Лекция 05
БГТУ, ФИТ, 1 семестр

Основы программной инженерии (ПИ)

Современные системы программирования

План лекции:
- система программирования, язык программирования;
- система программирования Windows;
- система программирования Linux (UNIX-подобная);
- стандарт POSIX;
- стандарты языков программирования;
- парадигмы программирования.

1. На прошлых лекциях:

Система программирования:

комплекс программных средств, предназначенных для автоматизации процесса разработки, отладки ПО и подготовки программного кода к выполнению

Структура языка программирования:

Программа – алгоритм, записанный на языке программирования.

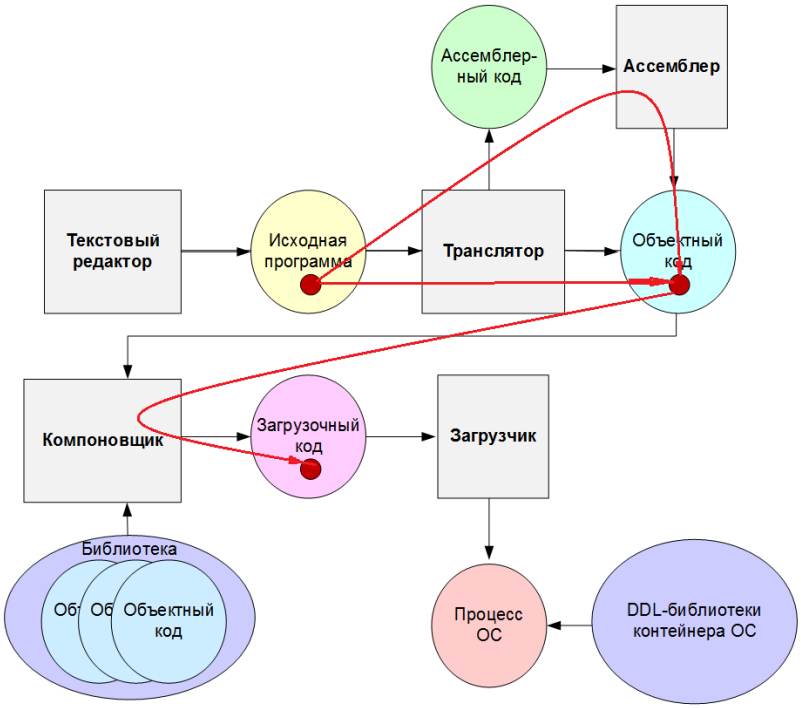
Текст программы (исходный код) – полное законченное и детальное описание алгоритма на языке программирования.

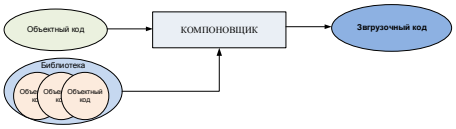
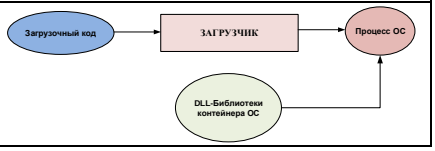
Объектный код: – результат работы транслятора. Один файл объектного кода – объектный модуль.

Объектный модуль – двоичный файл, который может быть объединён с другими объектными файлами при помощи редактора связей (компоновщика) для получения готового исполняемого модуля, либо библиотеки.

Загрузочный код – результат работы компоновщика.
Один файл загрузочного кода – загрузочный модуль.

От исходного кода к исполняемому модулю:



Компилятор (транслятор) – программа, преобразующая исходный код на одном языке программирования в исходный код на другом языке; результат – объектный модуль	**Компоновщик** (linker, редактор связей) – программа, принимающая один или несколько объектных модулей и формирующая на их основе загрузочный модуль	**Загрузчик** (loader) – программа, предназначенная для запуска процесса операционной системы на основе загрузочного модуля
		

2. Система программирования

Система программирования – инструментальное ПО, предназначенное для разработки программного продукта на этапах программирования и отладки. Каждая система программирования должна иметь некоторый встроенный в нее язык программирования. Ядром системы программирования является язык программирования.	

Интегрированная среда разработки:

набор инструментов для разработки и отладки программ, имеющий общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы.

Система программирования является основным инструментом программиста.

3. Классическая система программирования

С развитием сервисных средств ОС появились командные процессоры, что позволило объединять последовательность вызовов системных программ в единые командные файлы. Это упростило работу по запуску компонентов систем.

Специализированные командные процессоры (***координатор make*** или командный ***интерпретатор shell***) представляли собой интерпретаторы, на вход которым подавались файлы, записанные на особом командном языке.

Ядром системы программирования является язык программирования.

4. Интегрированная среда разработки

> **Интегрированная среда разработки (integrated development environment – IDE):** – набор инструментов для разработки и отладки программ, имеющий общую интерактивную графическую оболочку, поддерживающую выполнение всех основных функций жизненного цикла разработки программы.

Примеры IDE: `Visual Studio, NetBeans, Eclipse, Embarcadero Delphi` и пр.

JDeveloper — бесплатная интегрированная среда разработки ПО. Для разработки на языках программирования Java, JavaScript, BPEL, PHP, SQL, PL/SQL; на языках разметки HTML, XML.	
	1998 г.
разработчик	Oracle
написана на	Java
ОС	кроссплатформенная
платформа	Java Virtual Machine
последняя версия	12c (12.2.1.2.0) (19.10.2016
сайт	http://www.oracle.com/technetwork/developer-tools/jdev/

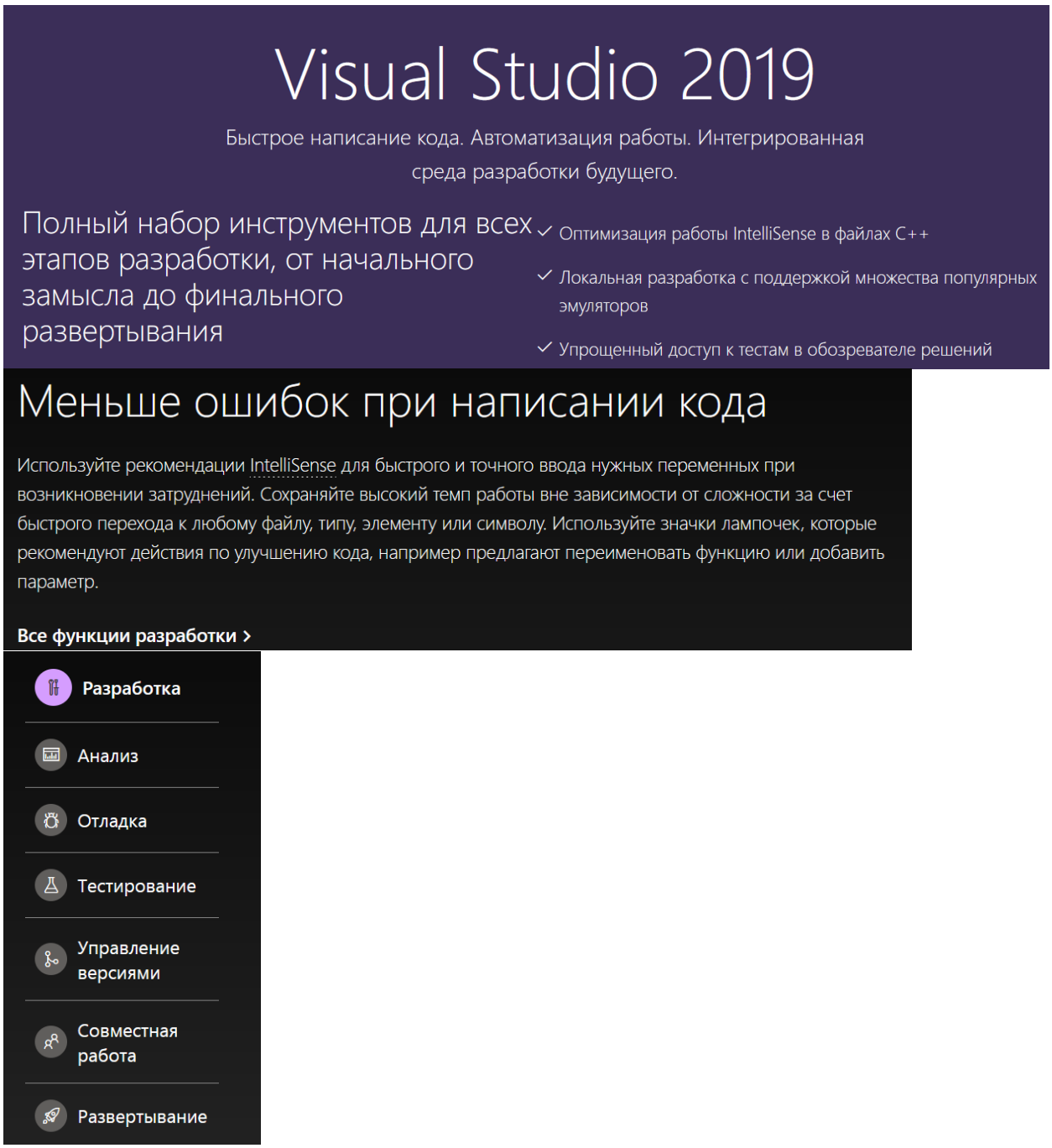
NetBeans — свободная интегрированная среда разработки ПО на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и др.	
	1997 г.
разработчик	Apache Software Foundation, Oracle и Sun Microsystems
написана на	Java
ОС	Microsoft Windows, Linux, macOS и Solaris
платформа	Java Virtual Machine
последняя версия	12.5 (13 сентября 2021)
сайт	netbeans.apache.org

Eclipse — свободная интегрированная среда разработки модульных кроссплатформенных приложений.	
	7 ноября 2001
разработчик	Eclipse Foundation
написана на	Java
ОС	GNU/Linux, macOS, Microsoft Windows, Solaris
платформа	Java Virtual Machine
последняя версия	4.21.0 (15 сентября 2021)
сайт	eclipse.org

Oracle Sun Studio — интегрированная среда разработки для языков программирования Си, C++ и Фортран	
	
разработчик	Oracle Corporation
написана на	SPARC, x86, x86-64
ОС	Solaris, OpenSolaris, Linux
последняя версия	Oracle Developer Studio 12.6 (5 июля 2017 года)
сайт	http://www.oracle.com/technetwork/server-storage/solarisstudio/overview

Embarcadero Delphi — интегрированная среда разработки ПО для Microsoft Windows, Mac OS, iOS и Android на языке Delphi (раньше Object Pascal). Embarcadero Delphi является частью пакета Embarcadero RAD Studio	
	1995 г.
разработчик	Embarcadero Technologies
написана на	Delphi и Object Pascal
ОС	Microsoft Windows
последняя версия	10.4.2 Sydney (24.02.2021)
сайт	embarcadero.com/ru/product

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки ПО и другие инструменты. Для разработки консольных приложений, игр, приложений с графическим интерфейсом, веб-сайтов, веб-приложений, веб-служб как в нативном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.	
	1997 г.
разработчик	Microsoft
написана на	C++ и C#
ОС	Microsoft Windows, macOS
последняя версия	16.11.3 (14 сентября 2021)
сайт	visualstudio.microsoft.com



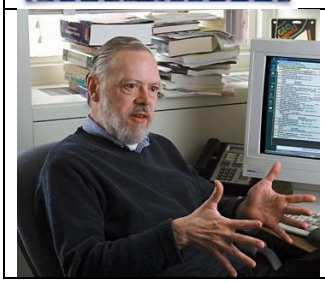
5. ОС и поддерживаемые платформы

Windows	ARM, IA-64, Itanium, MIPS, DEC Alpha, PowerPC и x86	Microsoft
Linux	DEC Alpha, x86, x86-64, ARM, PowerPC, RISC-V и MIPS	Свободное программное обеспечение
macOS	Motorola 68k, PowerPC, IA-32, x86-64, ARM	Apple Inc.
OS/2	x86	IBM, Microsoft
Unix	Intel x86	
Solaris	SPARC, x86, x86-64, PowerPC	Sun Microsystems

6. Система программирования UNIX

Unix («UNIX» – зарегистрированной торговой маркой The Open Group) – семейство переносимых, многозадачных и многопользовательских операционных систем, которые основаны на идеях оригинального проекта AT&T Unix, разработанного в 1970-х годах в исследовательском центре Bell Labs Кеном Томпсоном, Деннисом Ритчи и другими.

Дизайнеры языков программирования:

	Кеннет Лейн Томпсон – пионер компьютерной науки, известен своим вкладом в создание языка программирования C и операционной системы UNIX. Написал язык программирования B, предшественник языка C, участвует в создании языка программирования Go
	Деннис Ритчи – создатель языка программирования Си. Вместе с Кеном Томпсоном разработал Си для создания операционной системы UNIX. «У Ньютона есть фраза о стоящих на плечах гигантов. Мы все стоим на плечах Денниса», – Брайан Керниган.

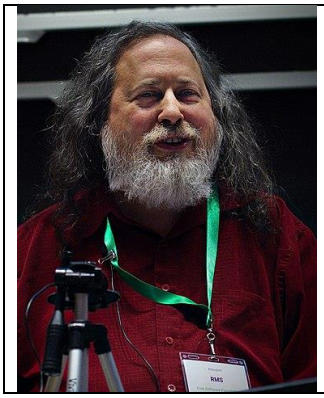
Операционные системы семейства Unix характеризуются модульным дизайном, в котором каждая задача выполняется отдельной утилитой, взаимодействие осуществляется через единую файловую систему, а для работы с утилитами используется командная оболочка.

1. Unix-подобные системы в отличие от других операционных систем изначально разрабатывались как многопользовательские многозадачные системы. В Unix может одновременно работать сразу много людей, каждый за своим терминалом, при этом каждый из них может выполнять множество различных вычислительных процессов, которые будут использовать ресурсы именно этого компьютера.

2. Unix является мультиплатформенной системой. Ядро системы разработано таким образом, что его легко можно приспособить практически под любой микропроцессор.

Linux (GNU/Linux) – семейство Unix-подобных операционных систем на базе ядра Linux, включающих тот или иной набор утилит и программ **проекта GNU** (проект по разработке свободного программного обеспечения, запущенный известным программистом и сторонником СПО Ричардом Столлманом 27

сентября 1983 года в Массачусетском технологическом институте), и, возможно, другие компоненты.

	Ричард Мэттью Столлман – основатель движения свободного программного обеспечения, проекта **GNU**, Фонда свободного программного обеспечения и Лиги за свободу программирования. Автор концепции «копилефта» (в противоположность традиционному подходу к авторскому праву, при котором ограничивается свобода копирования произведений, копилефт стремится использовать законы об авторском праве для расширения прав и свобод людей).

Все Unix-подобные операционные системы используют монолитное ядро, которое управляет процессами, сетевыми функциями, периферией и доступом к файловой системе. Драйверы устройств либо интегрированы непосредственно в ядро, либо добавлены в виде модулей, загружаемых во время работы системы.

Linux доминирует на рынке смартфонов (операционная система Android имеет в основе Linux ядро) и интернет-серверов. Linux полностью бесплатная система, в основном построенная на открытом программном обеспечении,

7. Система программирования Windows

Windows – семейство закрытых (или проприетарных) операционных систем, разрабатываемых компанией Microsoft.

Принципы взаимодействия программ в Windows.

Интерфейс вызовов функций в Windows: доступ к системным ресурсам осуществляется через целый ряд системных функций. Совокупность таких функций называется прикладным программным интерфейсом или API (Application Programming Interfase). Для взаимодействия с Windows приложение запрашивает функции API, с помощью которых реализуются все необходимые системные действия, такие как выделение памяти, вывод на экран, создание окон и т.п.

Функции API содержатся в библиотеках динамической загрузки (Dynamic Link Libraries, или DLL), которые загружаются в память только в тот момент, когда к ним происходит обращение, т.е. при выполнении программы. Работа через Windows API – это наиболее близкий к операционной системе способ взаимодействия с ней из прикладных программ.

Многозадачность в Windows

В Windows существует два типа многозадачности: основанный на процессах и основанный на потоках.

Процесс – это программа, которая выполняется. При многозадачности такого типа две или более программы могут выполняться параллельно.

Поток – это отдельная часть исполняемого кода (название произошло от понятия «направление протекания процесса»). В многозадачности данного типа отдельные потоки внутри одного процесса также могут выполняться одновременно.

Программа ожидает получения сообщения от Windows. Когда это происходит, то выполняется некоторое действие. После завершения выполнения этого действия программа ожидает следующего сообщения. Windows может посылать программе сообщения различных типов. Например, каждый раз при щелчке мышью в окне активной программы посылается соответствующее сообщение. Другой тип сообщений посылается, когда необходимо обновить содержимое активного окна. Сообщения посылаются также при нажатии клавиши, если программа ожидает ввода с клавиатуры. По отношению к программе сообщения появляются случайным образом.

8. Стандарт POSIX

> **Стандарт POSIX** – переносимый интерфейс операционных систем) — набор стандартов, описывающих интерфейсы между операционной системой и прикладной программой (системный API), библиотеку языка C и набор приложений и их интерфейсов.

Традиционно говорят о двух мирах, двух системах мировоззрения, присущих пользователям операционных систем Windows и UNIX.

POSIX – это стандарт, описывающий интерфейс между операционной системой и прикладной программой.

Общие принципы POSIX систем:

1) Это многопользовательские системы с разграничением прав по имени пользователя и группы; наличие пользователя root с неограниченными правами.
2) Древовидная файловая система, с единым корнем от /; большинство сущностей системы отображается как имя в дереве файловой системы; функциональное назначение каталогов файловой системы сохраняется примерно постоянным от одной системы к другой.

3) Применение символьных форматов: конфигурация системы и всех программных пакетов представляются в текстовых файлах (например, в файлах XML), это позволяет изменять все конфигурации простым текстовым редактированием.
4) Единообразный набор консольных утилит-команд (стандарт POSIX 2).
5) Единый API программирования языка C.

Преимущества, которые даёт совместимость операционной системы со стандартами POSIX:

- простота переноса (портирование) программных проектов их одной операционной системы в другую: часто это достигается путём выполнения ряда чисто формальных действий;
- простота для программиста-разработчика перехода от одной операционной системы к другой.

К POSIX системам принадлежат операционные системы: Linux, все ветви BSD (FreeBSD, NetBSD, OpenBSD, ...), Sun/Oracle Solaris, Mac OS, QNX, MINIX3 и другие. Системы, которые не являются POSIX совместимыми: Windows, Plan 9, Inferno, Blue Botle и некоторые другие.

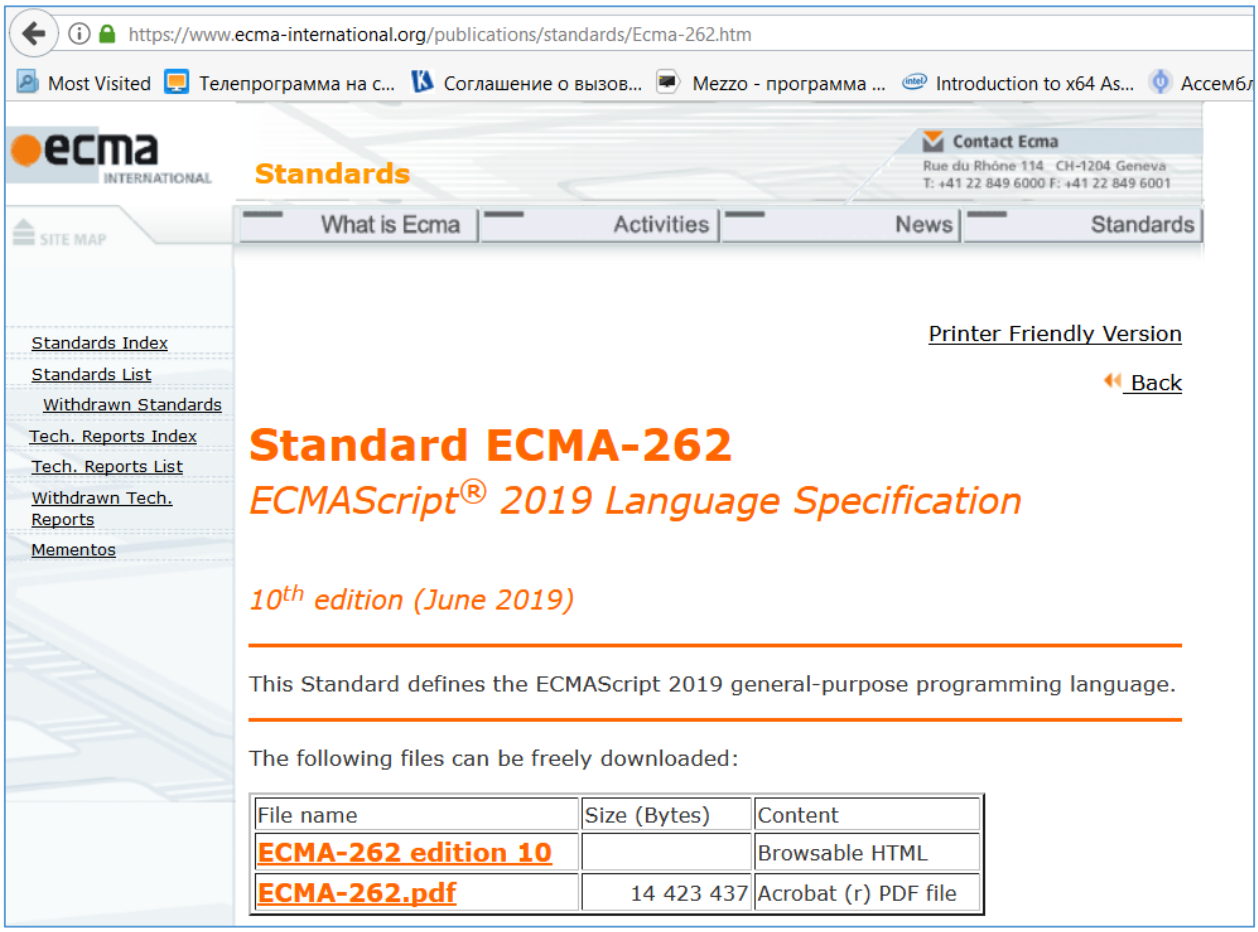
Однако никто не запрещает системам, не являющимся клонами (потомками) UNIX, поддерживать стандарт POSIX.

9. Стандарты языков программирования

Стандарт языка программирования: Visual C++ 2017 версия 15.3 – это реализация стандарта C++17 или ISO/IEC 14882:2017.

Международная организация по стандартизации (ИСО) одобрила и опубликовала C++ 20, последнюю версию объектно-ориентированного языка программирования.

Новая версия C++ выходит примерно каждые три года, ей присваивают номер года. Стандарт языка C++ 20 является преемником стандарта C++ 17.



ECMAScript — это встраиваемый расширяемый язык программирования, основа для построения других скриптовых языков. Стандартизирован международной организацией Ecma в спецификации ECMA-262. Расширениями языка являются JavaScript (Netscape), JScript (Microsoft) и ActionScript.

ECMAScript — это официальный стандарт языка JavaScript (Слово JavaScript не могло быть использовано, потому что слово Java является торговой маркой компании Sun)

Это восьмое издание спецификации языка ECMAScript.

Java Language and Virtual Machine Specifications
https://docs.oracle.com/javase/specs/

10. Парадигмы (стили) программирования

Язык программирования строится в соответствии с базовой моделью вычислений и парадигмой программирования (пример парадигмы: объектно-ориентированное программирование).

> **Парадигма программирования** – это совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию).

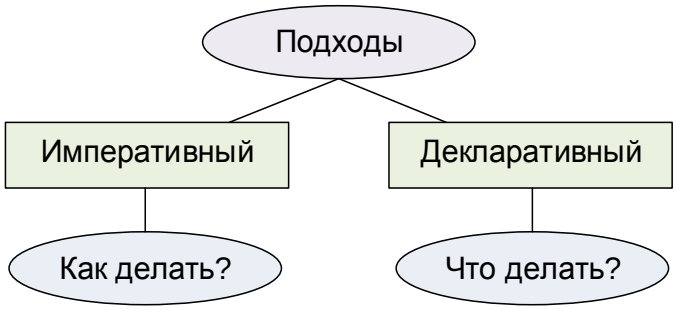
Методология программирования включает в себя модель вычислений для данного стиля.

Методология разработки программного обеспечения – совокупность методов, применяемых на различных стадиях жизненного цикла программного обеспечения.

Неструктурное программирование характерно для наиболее ранних языков программирования. Сложилось в середине 40-х с появлением первых языков программирования.
Основные признаки:

- строки как правило нумеруются;
- из любого места программы возможен переход к любой строке;

Основные подходы к программированию:



Императивное программирование (от греч. `imper` – действие) предполагает, что программа явно описывает алгоритм решения конкретной задачи (действия исполнителя), т.е. описывает как решать поставленную задачу.

Декларативное программирование (лат. `declaratio` – объявление) – это предварительная реализация «**решателя**» для целого класса задач.
Тогда для решения ***конкретной задачи*** этого класса достаточно декларировать в терминах данного языка только её условие:
`(исходные данные + необходимый вид результата)`
«**Решатель**» сам выполняет процесс получения результата, реализуя известный ему алгоритм решения.

Императивное программирование:
`Программа = последовательность действий`, дающая точные указания компьютеру о том, **как** получить решение.

Используемые конструкции языка:
- последовательность действий;
- условная конструкция;
- цикл.

Пример:	***Псевдокод:***

Функция с именем add, принимает на вход массив и возвращает сумму всех его элементов.	`function add(array)`
	`{`
	`  let result = 0`
	`  for (let i = 0; i < array.length; i++)`
	`  result += array[i];`
	`  return result;`
	`}`

Декларативное программирование:

`Программа = описание действий`, которые необходимо выполнить компьютеру для получения результата.

Отвечает на вопрос **что** надо выполнить.

Пример:	***Псевдокод:***

Функция с именем add, принимает на вход массив и возвращает сумму всех его элементов.	`function add(array)`
	`{`
	`  return array.reduce((prev, current) => prev + current, 0)`
	`}`

Функциональное программирование:

`Программа = система определений и функций`, описывающих что нужно вычислить, а как это сделать – решает транслятор; последовательность действий не прослеживается.

Раздел дискретной математики. Основой функционального программирования является лямбда-исчисление

Объектно-ориентированное программирование:

`Программа = несколько взаимодействующих объектов + функциональность` (действия и данные распределяются между этими объектами).

Распределённое (параллельное) программирование:

Программа = совокупность описаний процессов, которые могут выполняться как параллельно (при наличии нескольких процессоров), так и в псевдопараллельном режиме (при наличии одного процессора).

Логическое программирование:

`Программа = система определений вида «условие => новый факт».`

Программа представляет собой описание фактов и правил вывода в некотором логическом исчислении. Результат, (который часто записывается как вопрос), получается системой путем логического вывода. Раздел математической логики.

Распределённое (параллельное) программирование:

Программа = совокупность описаний процессов, которые могут выполняться как параллельно (при наличии нескольких процессоров), так и в псевдопараллельном режиме (при наличии одного процессора).

Визуальное программирование:

`Программа = способ создания программы для ЭВМ путём манипулирования графическими объектами` (вместо написания её текста).

Визуальное программирование позволяет программировать на уровне алгоритмов, а не программного кода.

Пакет визуального программирования генерирует, написанный на языках программирования (1GL, 2Gl, 3GL), на основании составленной программистом «блок-схемы» в автоматическом режиме.

Аспектно-ориентированное программирование:
Программа = к уже существующему коду добавляется дополнительного поведение, так называемой сквозной функциональности.

Пример функции возведения в квадрат в некоторых языках программирования.

Императивный **C**:	Функциональный **Scheme**:

int square(int x)
{
  return x * x;
}
	(define square
	(lambda (x)
	(* x x)))

Конкатенативный **Joy**:	Конкатенативный **Factor**:
DEFINE square == dup * .	: square (x -- y) dup *;