Совокупность правил, определяющих допустимые конструкции (слова, предложения) языка, т.е. его форму, образуют **синтаксис** языка, а сово- купность правил, определяющих смысл синтаксически корректных конструк- ций языка, т.е. его содержание, – **семантику** языка.

**Императивные языки** описывают процесс вычислений в виде команд, изменяющих состояние программы.

В **языках функционального программирования** вычисления в основном производятся путём применения функций к заданному набору данных.

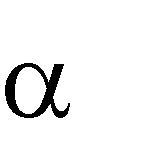
Термин **декларативный** используется по отношению к языкам, которые указывают, *какие* вычисления должны быть выполнены.

**Объектно-ориентированные языки** – это языки программирования, поддерживающие концепцию объектно-ориентированного программирования. Вместо проблемы разбиения задачи на функции, в объектно-ориентированном программировании задача представляется в виде совокупности объектов, обла- дающих сходными свойствами и набором действий, которые можно с ними производить.

**Цепочкой символов** называют произвольную упорядоченную конечную последовательность символов, записанных один за другим.

Количество символов в цепочке определяет её **длину**.

**Алфавит** – это счётное множество допустимых символов языка.



**Языком L** над алфавитом **V** (**L (V**)) называется некоторое счётное под- множество цепочек конечной длины из множества всех цепочек над алфавитом

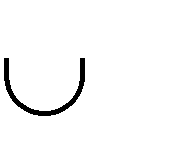
**V**.

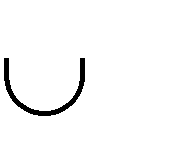
**Грамматика** *–* это описание способа построения предложений некоторо- го языка. Она относится ко второму способу определения языков – порожде- нию цепочек символов.

**Правило** (или продукция) – это упорядоченная пара цепочек символов (α, β). В правилах важен порядок цепочек, поэтому их чаще записывают в виде α

→ β (или α ::= β). Такая запись читается как «α порождает β» или «α по опреде- лению есть β».

Язык, заданный грамматикой **G**, обозначается как **L (G**). Две грамматики, **G** и **G'**, называются **эквивалентными**, если они определяют один и тот же язык: **L (G**) = **L (G'**). Две грамматики, **G** и **G'**, называются **почти эквивалент- ными**, если заданные ими языки различаются не более чем на пустую цепочку символов: **L (G**) {ε} = **L (G'**) {ε}.



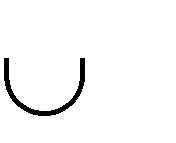


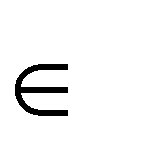
**Множество терминальных символов VT** содержит символы, которые входят в алфавит языка, порождаемого грамматикой.

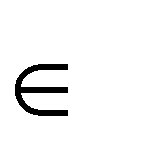
**Множество нетерминальных символов VN** содержит символы, которые определяют слова, понятия, конструкции языка.

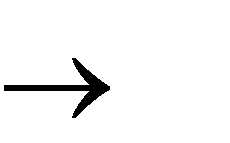
**Выводом** называется процесс порождения предложения языка на основе правил определяющей язык грамматики.

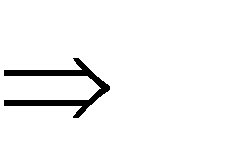
Цепочка β = δ1γδ2 называется **непосредственно выводимой** из цепочки α = δ1ωδ2 в грамматике **G** (**VT**, **VN**, **P**, *S*), **V** = **VN VT**, δ1, γ, δ2 **V**\*, ω **V**+, ес- ли в грамматике существует правило ω γ. Иными словами, цепочка β выво- дима из цепочки α в том случае, если можно взять несколько символов в цепоч- ке α, поменять их на другие символы, согласно некоторому правилу граммати- ки, и получить цепочку β. Непосредственная выводимость цепочки β из цепоч-





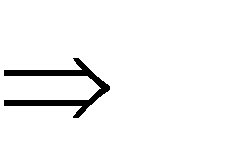


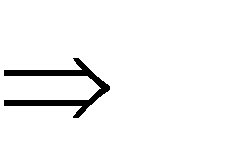


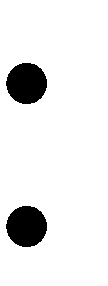


ки α обозначается как: α β.

Цепочка β называется **выводимой** из цепочки α (α \* β) в случае, если выполняется одно из двух условий:

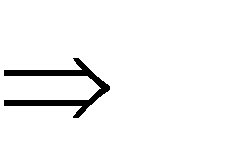


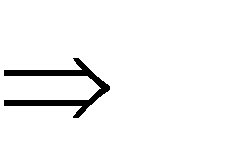




β непосредственно выводима из α (α β);

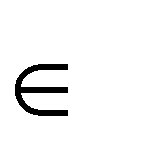
существует γ такая, что γ выводима из α, и β непосредственно выводима из γ (α γ, γ β).

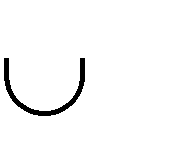


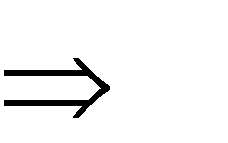


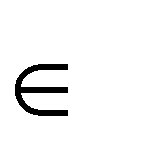
Вывод называется **законченным** (или **конечным***),* если на основе цепоч- ки β, полученной в результате этого вывода, нельзя больше сделать ни одного шага вывода. Иначе говоря, вывод называется законченным, если цепочка β, полученная в результате этого вывода, пустая или содержит только терминаль- ные символы грамматики. Цепочка β, полученная в результате законченного вывода, называется **конечной** цепочкой вывода.

Цепочка символов α **V**\* называется **сентенциальной формой** грамма- тики **G** (**VT**, **VN**, **P**, *S*), **V** = **VT VN**, если она выводима из целевого символа грамматики *S*: *S \** α. Если цепочка α **VT**\* получена в результате закончен- ного вывода, то она называется **конечной сентенциальной формой***.*





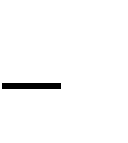


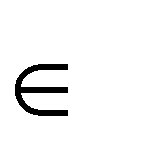


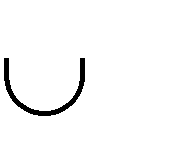
Вывод называется **левосторонним***,* если в нём на каждом шаге вывода правило грамматики применяется всегда к крайнему левому нетерминальному символу в цепочке.

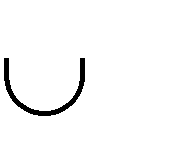
Аналогично, вывод называется **правосторонним**, если в нём на каждом шаге вывода правило грамматики применяется всегда к крайнему правому не- терминальному символу в цепочке.

**Деревом вывода** грамматики **G** (**VT**, **VN**, **P**, *S*) называется дерево, кото- рое соответствует некоторой цепочке вывода и удовлетворяет следующим условиям:

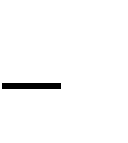
 каждая вершина дерева обозначается символом грамматики

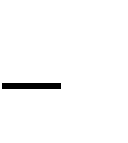


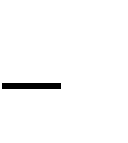


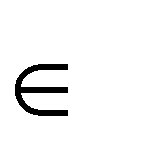


*А* (**VT VN** {ε});

 корнем дерева является вершина, обозначенная целевым символом грамматики – *S*;

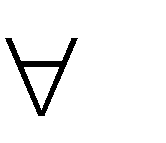
 листьями дерева (концевыми вершинами) являются вершины, обозна- ченные терминальными символами грамматики или символом пустой цепочки ε;

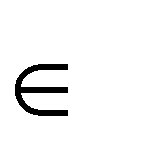
 если некоторый узел дерева обозначен нетерминальным символом

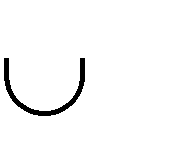


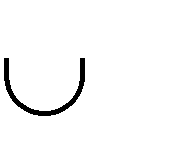
*А* **VN**, а связанные с ним узлы – символами *b*1, *b*2, …, *bn*; *n* > 0,

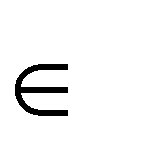
*i*, 0 ≤ *i≤ n*: *bi* (**VT VN** {ε}), то в грамматике **G** (**VT**, **VN**, **P**, *S*)











существует правило *A* → *b*1 | *b*2 | … | *bn* **Р**.

Если для каждой цепочки символов языка, заданного грамматикой, мож- но построить единственный левосторонний (и единственный правосторонний) вывод или, что то же самое, построить единственное дерево вывода, то такая грамматика называется **однозначной**. Иначе грамматика называется **неодно-**

**значной.**

