**Лабораторная работа №1**

**Синтаксис и семантика языка программирования**

***Цель работы*** – закрепление теоретических знаний, приобретение умений и практических навыков формального описания синтаксиса языка программирования и формирования перечня семантических соглашений.

**Основные сведения**

Разработка языка программирования начинается с определения его синтаксиса и семантики. Язык – это множество строк (предложений, цепочек), представляющих собой последовательность символов, каждый из которых принадлежит некоторому конечному алфавиту (словарю языка). Каждая строка языка формируется из словаря в соответствии с заданными правилами. Совокупность таких правил формирования называется грамматикой языка и определяет его структуру. Язык, используемый для описания грамматики какого-либо языка, называется метаязыком.

В описании языка различают синтаксис и семантику. Синтаксис – это множество формальных правил порождения правильно построенных строк языка. Семантика – это смысловое, содержательное значение каждой строки.

Одним из распространенных метаязыков для формального описания синтаксиса формального языка является форма Бэкуса-Наура (БНФ) и его различные модификации. БНФ используется для описания [контекстно-свободных](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D0%B5%D0%BA%D1%81%D1%82%D0%BD%D0%BE-%D1%81%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0) [формальных грамматик](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0). В БНФ одни синтаксические категории последовательно определяются через другие категории в виде правил (продукций). В левой части правила в угловых скобках записывается определяемое понятие, представляющее собой металингвистическую переменную. В правой части правила записываются все возможные варианты определения конструкции, задаваемой этим правилом. Варианты разделяются метасимволом «**|**», имеющим смысл «или». Левая и правая части правила разделяются метасимволом «::=», означающим «определяется как». Определяемые понятия (металингвистические переменные) называют нетерминальными символами (нетерминалами), символы словаря языка – терминальными символами (терминалами). Таким образом, левая часть правила всегда нетерминал, а вариант определения конструкции из правой части представляет собой цепочку нетерминалов и терминалов. Набор правил, определяющих все синтаксические конструкции языка, описывает ее грамматику.

Возможным значением нетерминала (допустимой строкой языка) может являться только та последовательность (строка) терминалов, которая может быть выведена из этого нетерминала. Процесс вывода начинается со строки, содержащей нетерминалы. Последовательно производится замена одного из нетерминалов на соответствующую правую часть правила до тех пор, пока не получится строка, состоящая только из терминалов.

В качестве примера рассмотрим БНФ, определяющую синтаксис константы целого типа:

<константа> ::= <цифра> **|** <константа> <цифра>

<цифра> ::= 0 **|** 1 **|** 2 **|** 3 **|** 4 **|** 5 **|** 6 **|** 7 **|** 8 **|** 9

Здесь нетерминалами являются <константа>, <цифра>, а терминалами – арабские цифры. Строка 218 выводится следующим образом (⇒ обозначает замену):

<константа> ⇒ <константа> <цифра> ⇒ <константа> 8 ⇒

<константа> <цифра> 8 <константа> 1 8 ⇒

<цифра> 1 8 ⇒ 2 1 8

Следует заметить, что особенностью некоторых правил может являться наличие в них рекурсии (как в приведенном выше примере), когда в определении синтаксической конструкции содержится сама определяемая конструкция (так называемая явная рекурсия). Возможна также неявная рекурсия, когда конструкция используется на некотором промежуточном шаге определения этой конструкции.

БНФ часто отождествляют с формальными грамматиками, используемыми в теории формальных языков.

Для удобства восприятия и более естественного представления альтернативных, необязательных и повторяющихся частей синтаксических конструкций часто используют расширения БНФ (РБНФ). Главное преимущество РБНФ перед БНФ – возможность описывать простые повторяющиеся конструкции неопределённой длины (списки, строки, последовательности) без рекурсивных правил. Из множества существующих вариантов РБНФ можно рекомендовать следующий.

Металингвистическая переменная (нетерминал) обозначается произвольной символьной строкой (без использования угловых скобок как в БНФ). Если нетерминал состоит из нескольких смысловых слов, то они записываются слитно или разделяются символом подчеркивания. При этом для удобства восприятия целесообразно каждое ее слово начинать с прописной буквы.

Терминальные символы изображаются словами, написанными буквами латинского алфавита (ключевые слова) или цепочками символов, заключенными в одиночные (′) или двойные (″) кавычки. Для удобства восприятия ключевые слова дополнительно можно выделить жирным шрифтом.

Левая и правая части правила разделяются метасимволом "=" (вместо "::=" в БНФ), альтернативные варианты разделяются метасимволом "**|**". Каждое правило заканчивается точкой.

Квадратные скобки "[" и "]" означают, что заключенная в них синтаксическая конструкция может отсутствовать.

Фигурные скобки "{" и "}" означают нуль или более повторений заключенной в них синтаксической конструкции.

Недостатком РБНФ (как и БНФ) является то, что они описывают грамматическую структуру формального языка без учёта контекстных зависимостей, а значит, при наличии таких зависимостей РБНФ-описание оказывается неполным, и некоторые правила синтаксиса и семантические особенности описываемого языка приходится излагать в обычной текстовой форме. Например, в РБНФ невозможно естественным образом отобразить тот факт, что некоторая операция требует операндов одного и того же типа. Поэтому, поскольку РБНФ не позволяет задавать контекстные условия, раскрывающие особенности семантики языка, их можно записать в словесной форме в виде перечня неформальных семантических соглашений.

Ниже приведен пример описания синтаксиса учебного языка с использованием РБНФ (назовем его языком «У»). Данный язык будет применяться для иллюстрации методических указаний и рекомендаций для остальных лабораторных работ.

1. Модуль = "**module**" Идент "**;**" Блок "**.**".

2. Идент = Буква {Буква **|** Цифpа }.

3. Блок = { Описание } "**begin**" ПоследОператоров "**end**".

4. Описание = "**type**" { ОписаниеТипа "**;**" } **|**

"**var**" { ОписаниеПерем "**;**" }.

5. ОписаниеТипа = Идент "**=**" Тип.

6. ОписаниеПерем = СписокИдент "**:**" Тип.

7. СписокИдент = Идент { "**,**" Идент }.

8. Тип = ПростойТип **|** ТипМассив.

9. ПростойТип = Идент **|** Диапазон.

10. Диапазон = Константа "**..**" Константа.

11. ТипМассив = "**array**" "**[**" ПростойТип { "**,**" ПростойТип } "**]**"

"**of**" Тип.

12. ПоследОператоров = Оператор { "**;**" Оператор }.

13. Оператор = Присваивание **|** Цикл.

14. Присваивание = Переменная "**:=**" Выражение.

15. Переменная = Идент **|** ИндексПеpеменная.

16. ИндексПеpеменная = Идент"**[**"ПростоеВыраж

{ "**,**" ПростоеВыраж } "**]**".

17. Цикл = "**while**" Выражение "**do**" ПоследОператоров "**end**".

18. Выражение = ПростоеВыраж [ Отношение ПростоеВыраж ].

19. Отношение = "**<**" **|** "**<=**" **|** "**>**" **|** "**>=**" **|** "**=**" **|** "**#**".

20. ПростоеВыраж = ["**+**" **|** "**–**" ] Терм { АддитОперация Терм }.

21. АддитОперация = "**+**" **|** "**–**" **|** "**or**".

22. Терм = Фактор { МультОперация Фактор }.

23. МультОперация = "**\***" **|** "**/**" **|** "**and**" **|** "**div**".

24. Фактор = Константа **|** Пеpеменная **|** "**(**" Выражение "**)**" **|**

"**not**" Фактор.

25. Константа = Целое [ "**.**" Целое ].

26. Целое = Цифpа { Цифpа }.

27. Цифра = "0"**|**"1"**|**"2"**|**"3"**|**"4"**|**"5"**|**"6"**|**"7"**|**"8"**|**"9".

Определение нетерминала «Буква» здесь не приведено ввиду его очевидности – определяется выбранным алфавитом (обычно строчные и прописные буквы латинского алфавита).

Краткая характеристика языка и семантические соглашения:

* Язык удовлетворяет семантическим соглашениям, характерным для многих языков программирования (единственность именования различных объектов программы, необходимость описания идентификатора до его использования и т.п.).
* В ключевых словах и идентификаторах прописные и строчные буквы не различаются.
* Ключевые слова языка зарезервированы, их нельзя использовать в качестве идентификаторов.
* Отсутствуют именованные константы.
* Предопределенные типы: integer, float, Boolean, char.
* В конструкции «Диапазон» (правило 10) константы должны быть целого типа, кроме того, значение первой константы должно быть меньше значения второй константы.
* Если в конструкции «ТипМассив» (правило 11) в качестве «ПростойТип» используется идентификатор («Идент»), он должен представлять тип «Диапазон».
* В операторе цикла конструкция «Выражение» после ключевого слова **while** должно быть типа Boolean (правило 17).
* Комментарий представляет собой любую последовательность символов, заключенную в фигурные скобки "{" и "}".

**Индивидуальные задания**

В качестве индивидуального задания на выполнение лабораторной работы предлагается некоторый язык-прототип, на основе которого необходимо разработать собственный учебный язык программирования. В качестве прототипа может быть предложен синтаксис какого-либо из существующих языков программирования. Вариант учебного языка с собственным синтаксисом может быть предложен и самими студентами. Синтаксис учебного языка разрабатывается студентом самостоятельно с поэтапным контролем и согласованием отдельных вопросов в процессе индивидуальных консультаций с преподавателем.

Общие требования к учебному языку:

* язык должен быть со строгой явной статической типизацией, т.е. типы всех объектов должны быть объявлены в специальном разделе описаний и не допускается неявное преобразование типов;
* должны быть ключевые слова, обозначающие начало и конец программы;
* программа должна состоять из раздела описаний и раздела (последовательности) операторов;
* должно быть, как минимум, три простых предопределенных базовых типа (целый, вещественный, логический);
* для арифметических выражений должны быть определены, как минимум, операции сложения, вычитания, умножения, деления, унарный плюс и унарный минус;
* для логических выражений должны быть определены, как минимум, операции дизъюнкции (ИЛИ), конъюнкции (И), отрицания (НЕ) и шесть операций отношения;
* обязательным оператором является оператор присваивания;
* текст программы должен допускать использование комментариев.

Приоритетность арифметических и логических операций определяется языком-прототипом.

Если в языке-прототипе отсутствуют какие-либо пункты перечисленных выше общих требований, следует разработать их самостоятельно.

К общим требованиям добавляются производные типы и другие операторы в зависимости от особенностей разрабатываемого языка.

Синтаксис учебного языка должен быть описан с использованием БНФ или РБНФ. Поскольку РБНФ (как и БНФ) не позволяет задавать контекстные условия, раскрывающие особенности семантики языка, необходимо сформировать соответствующий перечень семантических соглашений, раскрывающих эти семантические особенности. Следует дать также краткую характеристику языка, отражающую его особенности.

**Подготовка и выполнение работы**

1. Изучить описание работы и другие теоретические сведения, касающиеся тематики выполняемой работы.

2. Описать синтаксис учебного языка, используя нотацию БНФ или РБНФ. Рекомендуется следующий порядок построения правил:

2.1. Определить синтаксическую конструкцию, соответствующую главной программе, построить правила, определяющие программу в целом и место в ней описаний и операторов.

2.2. Построить правила, определяющие синтаксис отдельных описаний и операторов.

2.3. Провести анализ предусмотренных в языке операций (арифметических, логических, отношений и т.п.), определить их приоритетность (составить таблицу приоритетов операций, которую следует включить в характеристики языка) и построить правила, определяющие синтаксис выражений.

2.4. Построить правила, определяющие синтаксис идентификаторов и констант, которые могут быть использованы в описываемом языке.

3. Сформировать перечень семантических соглашений, раскрывающих семантические особенности синтаксических конструкций учебного языка.

4. Дать неформальное описание учебного языка на естественном языке, отражающее его особенности. В описании отразить используемый механизм типизации, указать предопределенные базовые типы, привести таблицу приоритетов операций, зарезервированы ключевые слова или нет и т.п.

**Контрольные вопросы**

1. Назначение метаязыков. Какие метаязыки Вы знаете?
2. Что такое синтаксис языка программирования?
3. Что такое семантика языка программирования?
4. Что такое словарь языка?
5. Что такое грамматика языка?
6. Назначение нетерминальных символов в БНФ.
7. Назначение терминальных символов в БНФ.
8. Как выводится строка по правилам БНФ?
9. Что такое РБНФ?
10. Основные металингвистические символы РБНФ.