# **Глава 1. Спецификация языка программирования**

## **Характеристика языка программирования**

Язык программирования SDE-2019 предназначен для выполнения простейших арифметических действий и операций над строками.

Язык программирования SDE-2019 является процедурным, строго типизированным, не объектно-ориентированным, компилируемым.

Процедурный язык программирования — язык высокого уровня, в котором используется метод разбиения программ на отдельные связанные между собой модули — подпрограммы (процедуры и функции).

Строго типизированный язык программирования — язык, в котором переменные привязаны к конкретным типам данных. Язык не позволяет смешивать в выражениях различные типы и не выполняет автоматические неявные преобразования.

Объектно-ориентированный язык программирования — язык, построенный на принципах объектно-ориентированного программирования. В основе концепции объектно-ориентированного программирования лежит понятие объекта — некой сущности, которая объединяет в себе поля (данные) и методы (выполняемые объектом действия).

Компилируемый язык программирования — язык программирования, исходный код которого преобразуется компилятором в исходный код на другом языке программирования.

* 1. **Алфавит языка**

Алфавит языка SDE-2019 основан на кодировке Windows-1251. Таблица кодировки Windows-1251 представлена на рисунке 1.1.

Исходный код может содержать символы латинского алфавита, цифры десятичной системы счисления от 0 до 9, русские символы разрешены только в строковых литералах.



Рисунок 1.1 – Алфавит входных символов

* 1. **Применяемые сепараторы**

Применяемые сепараторы в языке SDE-2019, приведены в таблице 1.1. Таблица 1.1 — Применяемые сепараторы

|  |  |
| --- | --- |
| Сепаратор | Область применения |
| « »(пробел) | Допускается везде, кроме идентификаторов и ключевых слов |
| ; | Разделение конструкций |
| { } | Заключение программного блока |
| = | Присвоение значения |
| ( ) | Приоритет операций (в выражениях), параметры функции |
| , | Разделение параметров |
| +,-,\*,/ ,% | Арифметические операции |

* 1. **Применяемые кодировки**

Для написания исходного кода на языке программирования SDE-2019 используется кодировка Windows-1251, которая представлена на рисунке 1.1.

tablr

## **Типы данных**

В языке SDE-2019 есть 2 типа данных: целочисленный и строковый. Описание типов данных, предусмотренных в данным языке представлено в таблице 1.2.

Таблица 1.2 – Типы данных языка SDE-2019

|  |  |
| --- | --- |
| Тип данных | Описание типа данных |
| little | Фундаментальный тип данных. Предусмотрен для объявления целочисленных положительных данных (2 байта).  Автоматически инициализируется нулевым значением.  Возможные операции:  *арифметические*  + – бинарный, суммирование;  - – бинарный, вычитание;  / – бинарный, деление;  \* – бинарный, умножение;  % – бинарный, остаток от деления;  = – присваивание значения; |
| text | Фундаментальный тип данных. Предусмотрен для объявления строк. (1 символ – 1 байт). Автоматическая инициализация строкой нулевой длины. Максимальное количество символов в строке – 255. … |

## **Преобразование типов данных**

В языке программирования SDE-2019 преобразование типов данных не поддерживается, т.е. язык является строго типизированным.

## **Идентификаторы**

Общее количество идентификаторов ограничено максимальным размером таблицы идентификаторов. Идентификаторы должны начинаться только с символов латинского алфавита, могут содержать цифры. Максимальная длина идентификатора равна восьми символам. Идентификаторы, объявленные внутри функционального блока, будут иметь область видимости (массив чисел), обозначающий блок в которой они объявлены. Максимальная вложенность по блокам равна 5.Данные правила действуют для всех типов идентификаторов.

Key word

## **Литералы**

В языке существует 2 типа литералов: целого и строкового типов. Краткое описание литералов представлено в таблице 1.3.

Таблица 1.3 – Описание литералов

|  |  |
| --- | --- |
| Тип литерала | Описание |
| Литералы целого типа | Целочисленные неотрицательные литералы, инициализируются 0. Литералы только rvalue. Представление!!!!!!!!!!!!!!!! |
| Строковые литералы | Символы, заключённые в “” (двойные кавычки), инициализируются пустой строкой, строковые переменные. Только rvalue. |

## **Область видимости идентификаторов**

Область видимости «сверху вниз» (по принципу С++). В SDE-2019 требуется обязательное объявление переменной перед её использованием. Все переменные должны находиться внутри программного блока языка. Имеется возможность объявления одинаковых переменных в разных блоках. Каждая переменная получает массив чисел ,обозначающий блок в котором она находится.

## **Объявление данных и область видимости**

В языке SDE-2019 объявление данных начинается с ключевого слова new, указывается тип данных и имя идентификатора. Требуется обязательное объявление переменной перед её использованием.

Примеры: new little a, new text b;

Все переменные в языке SDE-2019 имеют область видимости, а именно массив чисел отображающий блок в которой они находятся, что разрешает использование в различных функциях переменных с одинаковым именем. Параметры функции видны только внутри неё. Переменные, объявленные в одной функции, недоступны в другой.

## **Инициализация данных**

В момент объявления переменных в языке SDE-2019 происходит автоматическая инициализация в зависимости от типа данных. Инициализация другими значениями в момент объявления не допускается. Присвоение значения переменным должны происходить после непосредственного объявления. Виды инициализации представлены в таблице 1.4.

Таблица 1.4 — Способы инициализации переменных

|  |  |
| --- | --- |
| Вид инициализации | Примечание |
| new <тип данных> <идентификатор>; | Автоматическая инициализация: переменные типа little инициализируются нулём, переменные типа text – пустой строкой. |
| <идентификатор> = <значение>; | Присваивание переменной значения. |
| поумолч |  |

## **Инструкции языка**

Инструкции языка SDE-2019 представлена в таблице 1.5.

Таблица 1.5 – Инструкции языка

|  |  |
| --- | --- |
| Инструкция | Форма записи |
| Объявление переменной | new <тип данных> <идентификатор>; |
| Объявление функции | <тип данных> function <идентификатор> (<тип данных> <идентификатор>, …)  {  / программный блок /  return <идентификатор|литерал>;  } |
| Вызов функции | <идентификатор> (<идентификатор>, ...) |
| Присвоение значения | <идентификатор> = <значение>; |
| Печать данных | output <литерал|идентификатор>; |
| Функции стандартной библиотеки  (применяются при инициализации и в выражениях) | copytext(text a , text b) — копирование строки.  textlength(text a) — вычисляет длину строки и возращает ее. |
| Возвращаемое значение | return <литерал|идентификатор>; |

## **Операции языка**

Арифметические операции, которые можно использовать в языке SDE-2019, представлены в таблице 1.6.

Таблица 1.6 — Операции языка

|  |  |
| --- | --- |
| Инструкция | Форма записи |
| Арифметические операции языка | + — бинарное сложение  − — бинарный минус  \* — бинарное произведение  / — бинарное деление  % — бинарное остаток от деления  ( ) — приоритет операций |

Приоритет подчиняется правилам:

* если в вы­ра­же­ние без ско­бок вхо­дят толь­ко сло­же­ние и вы­чи­та­ние или толь­ко умно­же­ние и де­ле­ние, то дей­ствия вы­пол­ня­ют в том по­ряд­ке, в каком они на­пи­са­ны;
* если в вы­ра­же­ние без ско­бок вхо­дят не толь­ко дей­ствия сло­же­ния и вы­чи­та­ния, но и умно­же­ния и де­ле­ния, или оба этих дей­ствия, то сна­ча­ла вы­пол­ня­ют по по­ряд­ку (слева на­пра­во) умно­же­ние и де­ле­ние, а затем сло­же­ние и вы­чи­та­ние;
* если в вы­ра­же­нии име­ют­ся скоб­ки, то сна­ча­ла вы­чис­ля­ют зна­че­ние вы­ра­же­ний в скоб­ках.

## **Выражения и их вычисление**

Вычисление выражений – одна из важнейших задач языков программирования. Всякое выражение составляется согласно следующим правилам:

* Допускается использовать скобки для смены приоритета операций;
* Выражение записывается в строку без переносов;
* Использование двух подряд идущих операторов не допускается;
* Допускается использовать в выражении вызов функции, вычисляющей и возвращающей целочисленное значение.

Перед генерацией кода каждое выражение приводится к записи в польской записи для удобства дальнейшего вычисления выражения на языке ассемблера

## **Программные конструкции языка**

Основные программные конструкции языка SDE-2019 представлены в таблице 1.7.

Таблица 1.7 — Основные конструкции языка

|  |  |
| --- | --- |
| Конструкция | Реализация |
| Главная функция  (точка входа) | start  {  / программный блок /  } |
| Функции | <тип данных> function <идентификатор>  (<тип данных> <идентификатор>, …)  {  / программный блок /  return <идентификатор/литерал>.  } |

## **Семантические проверки**

В языке SDE-2019 выполняются следующие семантические проверки:

* проверка на присутствие единственной главной функции;
* проверка на правильную инициализацию;
* проверка на предварительное объявление идентификатора с ключевым словом type;
* проверка деления на нуль;
* проверка на дублирование идентификаторов;
* проверка на несоответствие типов в выражении;
* проверка на совпадение количества параметров функции;
* проверка на соответствие типа возвращаемого значения функции.

## **Стандартная библиотека и её состав**

В языке SDE-2019 предусмотрена стандартная библиотека. Функции, входящие в состав библиотеки, описаны в табл. 1.7. Стандартная библиотека подключается автоматически на этапе генерации кода.

Таблица 1.7 - Функции стандартной библиотеки языка MPG-2018

|  |  |
| --- | --- |
| Функция | Описание |
| copytext(text a , text b) | Данная функция void типа принимает два параметра (две строки). Функция копирует содержание второй строки(b) в первую строку(а). |
| textlenght (text a) | Данная функция целочисленного типа принимает один строковый параметр. Функция возводит количество значимых символов в строке а. |

## **Ввод и вывод данных**

Ввод данных языком программирования SDE-2019 не поддерживается.

Для вывода данных используется функция output (<имя идентификатора>).

Пример: output(a);

## **Точка** **входа**

Функция точки входа в языке программирования SDE-2019 представлена в таблице 1.10.

Таблица 1.10 — Точка входа

|  |  |
| --- | --- |
| Конструкция | Реализация |
| Главная функция  (точка входа) | start |

## **Препроцессор**

Препроцессор, принимающий и выдающий некоторые данные на вход транслятору, в языке SDE-2019 отсутствует.

## **Соглашения о вызовах**

В языке вызов функций происходит по соглашению о вызовах stdcall. Особенности stdcall:

– все параметры функции передаются через стек;

– память высвобождает вызываемый код;

– занесение в стек параметров идёт справа налево

## **Объектный код**

SDE-2019 транслируется в язык ассемблера.

## **Классификация сообщений транслятора**

Генерируемые транслятором сообщения определяют степень его информативности, то есть сообщения транслятора должны давать максимально полную информацию о допущенной пользователем ошибке при написании программы. Сообщения транслятора приведены в таблице 1.10.

Таблица 1.10 Классификация ошибок

|  |  |
| --- | --- |
| Номера ошибок | Характеристика |
| 0 – 200 | Системные ошибки |
| 200 – 299 | Ошибки лексического анализа |
| 300 – 399 | Ошибки семантического анализа |
| 600 – 699 | Ошибки синтаксического анализа |
| 400-499, 700-999 | Зарезервированные коды ошибок |