

**دانشگاه اصفهان**

**دانشکده مهندسی کامپیوتر**

**گزارش فاز اول پروژه‌ی درس طراحی کامپایلر**

**تحلیلگر لغوی – Lexical Analyzer**

**مهرآذین مرزوق – ۴۰۰۳۶۱۳۰۵۵**

# کلمات کلیدی

کلمات کلیدی زبان با حروف کوچک نوشته می شوند که در زیر ذکر شده اند.

bool break char continue else false for if int print return true

برای اینکه کلمات کلیدی به عنوان شناسه، شناخته نشوند، پیش از اعلام یک عبارت به عنوان شناسه،‌ آن عبارت را با لیستی از این کلمات کلیدی مقایسه می‌کنیم و در صورتی که این مقایسه به تشابه انجامید، عبارت را به عنوان کلمه کلیدی عنوان می‌کنیم.

KW = {"bool", "break", "char", "continue", "else", "false", "for", "if", "int", "print", "return", "true"}

# شناسه‌ها

یک شناسه نامی برای یک موجودیت در یک زبان برنامه نویسی است. دو موجودیت در این زبان برنامه نویسی عبارتند از متغیر و تابع. شناسه‌ها با یک حرف یا علامت زیر‌خط (ـ) آغاز می‌شوند و می‌توانند حاوی ارقام ، حروف و علامت های زیرخط باشند. شناسه ها نمی توانند برابر با هیچ یک از کلمات کلیدی باشند

عبارت منظم:

letter\_ [A-Za-z\_]

digit [0-9]

id letter\_ (letter\_)\*

دیاگرام گذار:

A diagram of a diagram

Description automatically generated

برای این‌که هیچ کلمه‌ی کلیدی‌ای به عنوان شناسه معرفی نشود باید در نهایت همه شناسه‌ها را با لیست کلمات کلیدی مقایسه کرد و در صورت شباهت،‌ عبارت را به عنوان کلمه کلیدی معرفی کنیم.

# علامت‌های نشانه‌گذاری

− علامت های آ کولاد باز } و آ کولاد بسته { در تعریف بلوک ها استفاده می شوند.

− علامت های پرانتز باز ) و پرانتز بسته ( در تعریف توابع، فراخوانی توابع، و عبارت های محاسباتی استفاده می شوند.

− علامت های کروشه باز ] و کروشه بسته [ برای تعریف آرایه های استفاده می شوند.

− علامت ویرگول , برای جدا کردن ورودی های تابع از یکدیگر در تعریف و فراخوانی تابع استفاده می شود.

− علامت نقطه ویرگول ; در پایان تعریف متغیرها، دستورات محاسبه ای و فراخوانی توابع، و همچنین در تعریف حلقه ها استفاده می شوند.

Notations -> { | } | ( | ) | [ | ] | , | ;

A black text on a white background

Description automatically generated

# توضیحات

توضیحات با دو علامت اسلش یا خط اریب // آغاز می شوند و با کاراکتر پایان خط معادل کد اسکی ۱۰ یا \n پایان می یابند. تحلیل گر لغوی توضیحات را به تحلیل گر نحوی ارسال نمی کند، اما پس از تحلیل لغات لیست همۀ توکن ها را چاپ می کند.

comments -> // (anything but \n)\* (\010)

A diagram of a diagram

Description automatically generated

# مقادیر عددی

مقادیر عددی می توانند در مبنای ده (دهدهی یا دسیمال) یا در مبنای شانزده (شانزده شانزدهی یا هگزادسیمال) باشند. یک عدد دهدهی می تواند مثبت یا منفی باشد که در صورت منفی بودن با علامت - آغاز می شوند. اعداد هگزادسیمال با دو کاراکتر 0x آغاز می شوند.

Numbers -> numbers10 | numbers16

numbers10 -> (-?) (digit)+

numbers16 -> 0x (hexdigit)+

digit -> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

hexdigit -> digit | [A - F]

A diagram of a diagram

Description automatically generated

# کاراکترها و رشته‌های ثابت

# عملگرها

عملگرهای حسابی 9 مانند + ، - ، \* ، / برای جمع، تفریق، ضرب، و تقسیم، استفاده کرد. برای به دست آوردن باقیمانده از عملگر % استفاده می شود.

arthop -> + | - | \* | / | %

عملگرهای یگانی + و - برای تعیین مثبت و منفی بودن اعداد به کار می روند. عملگرهای یگانی + و - بالاترین اولویت را دارند و پس از آنها \* ، / ، % هم اولویت بوده و در درجۀ دوم اولویت قرار دارند و در نهایت + و − اولویت سوم قرار می گیرند.

unitop -> + | -

عملگر‌های رابطه‌ای > ، >= ، < ، <= ، == ، != برای مقایسه دو مقدار به کار می روند.

relop -> >(=?) | <(=?) | == | !=

عملگرهای منطقی عطف && و فصل || و نقیض ! نیز در عبارات منطقی به کار می روند.

logop -> && | || | !

A diagram of a network

Description automatically generated

# فاصله‌های خالی

توکن ها توسط یک فاصلۀ خالی و یا ترکیبی از فاصله های خالی از یکدیگر جدا می شوند. یک فاصله خالی شامل کاراکتر فاصله با کد اسکی ۳۲ ، کاراکتر خط جدید با کد اسکی ۱۰ ، و کاراکتر ستون جدید با کد اسکی ۹ می شود.

ws -> \032 | \010 | \009