تحليل المفردات

Lexical Analysis

تحليل المفردات (التمشيط)

- هي عملية قراءة من اليسار إلى اليمين لسيل الرموز المكونة لبرنامج المصدر وتجميعها في سلسلة بطاقات Tokens.
- البطاقات عبارة عن تسلسل لرموز تحمل معنى مشترك.
- المعنى المشترك يعبر عن مضمون التعليمات البرمجية
- وهذا الطور لا يهتم بماهو معنى التعليمات ولكن يهتم بتجزئتها إلى مفردات

عمليات محلل المفردات

- في بعض الأحيان يقسم محلل المفردات إلى عمليتين متواليتين:

1. المسح/التمشيط Scanning:

- وهي عملية بسيطة لاعلاقة لها بإنتاج البطاقات ولكن تهتم بتحديد التعليقات ومعالجة/تقليص الفراغات المتكررة أو التي تكون في بداية أوامر البرنامج.

2. تحليل النص Syntax Analysis:

- وهي المهمة الأكثر تعقيداً وتختص بإنتاج البطاقات.

ما هو دور محلل المفردات؟

تتلخص مهمة محلل المفردات في:

1. قراءة الرموز (حروف, أرقام, وعلامات خاصة) الداخلة من البرنامج المصدري ثم وضعها في مجموعات من المفردات Lexemes ثم إنتاج سلسة من البطاقات Tokens.
2. التفاعل مع جدول الرموز حالة اكتشاف مفردة Lexeme تشكل تعريف لمتغير لكي يتم إدخالها بالجدول.

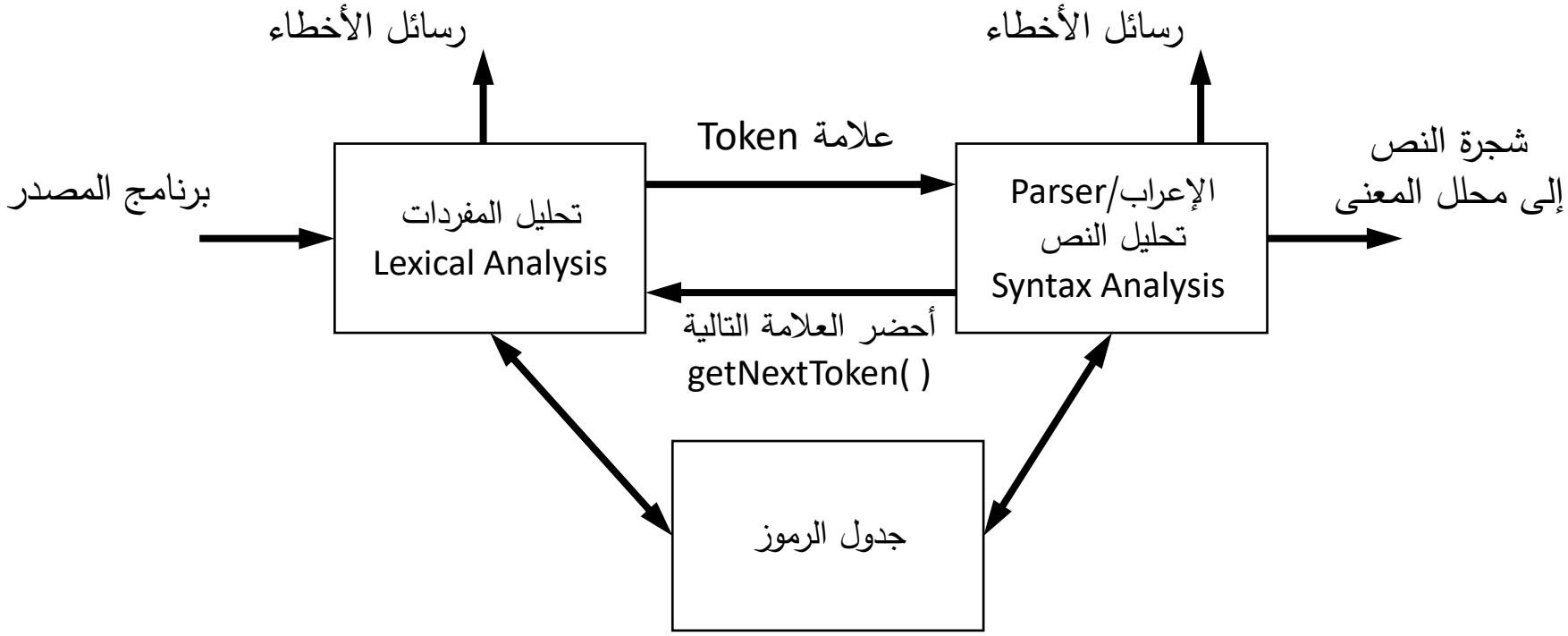
- في بعض الحالات يحتاج محلل المفردات إلى قراءة بعض المعلومات من جدول الرموز ليتمكن من تحديد نوع البطاقة Token لهذه المفردة ليمرره إلى محلل النص Parser.
  - مثلاً: التأكد أن المعرف identifier تم حفظه سلفاً في جدول الرموز أم لا.

ما هو دور محلل المفردات؟

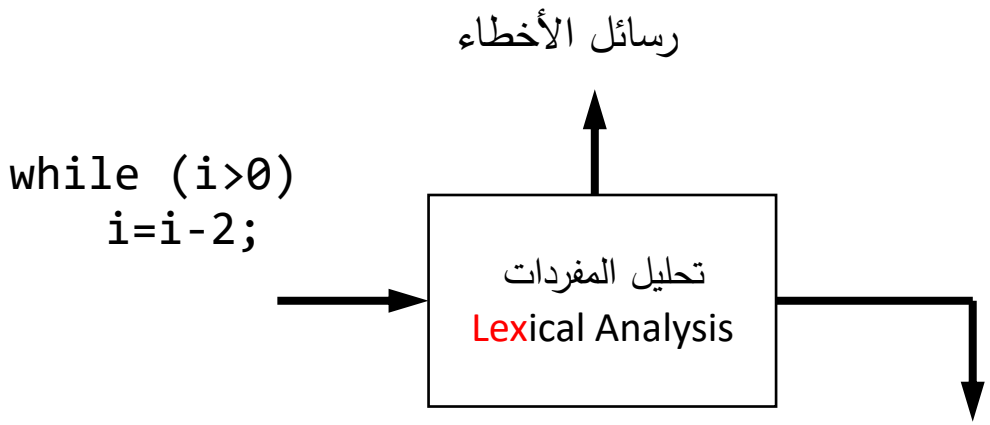
3. التواصل مع محلل النص Parser لإرسال البطاقات Tokens التي تُنتج من خلال تحليل مفردات برنامج المصدر.

- غالباً يكون التفاعل بعد استقبال نداء من محلل النص طالباً الدليل التالي. يتجسد هذا النداء في الأمر *getNextToken()* الذي يجعل محلل المفردات يقرأ الرموز من برنامج المصدر إلى أن يتعرف على المفردة التالية Next Lexeme ويستخرج منها الدليل التالي Next Token والتي بدورها يتم تسليمها إلى محلل النص.

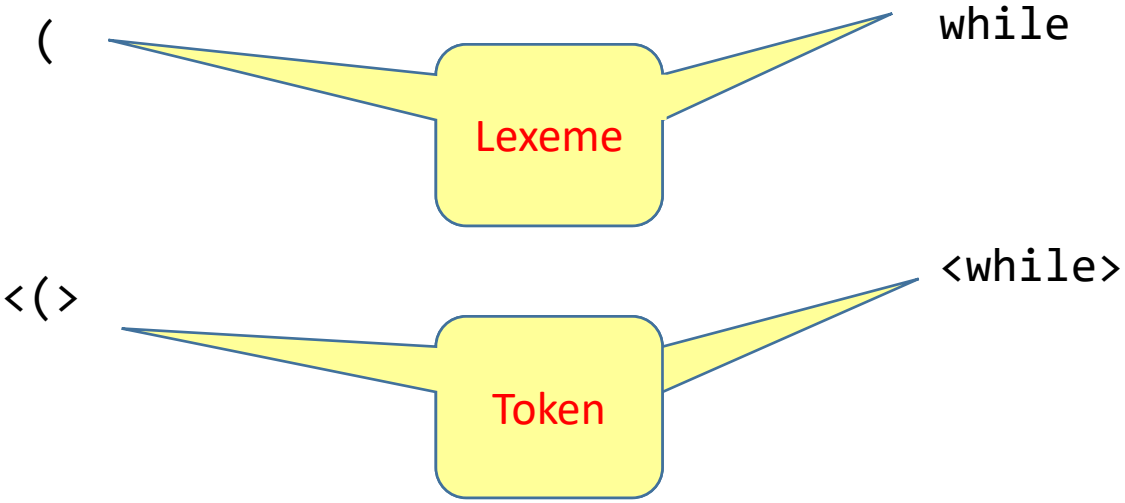
التفاعل بين محلل المفردات ومحلل النص



تحليل المفردات (التمشيط)



<while> <(> <id,1> <>> <0> <)> <id,1> <=> <id,1> <-> <2> <;>



ما هو دور محلل المفردات؟

4. إزالة التعليقات والفراغات التي لا تعتبر جزء من مفردات لغة البرمجة ولن تكون من ظمن العلامات Tokens.

- التعليقات هي التي يكتبها المبرمج لتفسير أو توضيح تعليمات البرنامج المصدري ولها علامات تختلف باختلاف لغة البرمجة مثل // و */ في لغة C و C++.
- الفراغات: وتشمل المسافات الفارغة, علامات السطر الجديد↵, مسافات متعددة Tab ↤↦,

ما هو دور محلل المفردات؟

5. اكتشاف الأخطاء في صياغة المفردات وربطها بمواقعها في البرنامج المصدري.

- يقوم محلل المفردات بمتابعة عدد رموز السطر الجديد ↵ لكي يستفاذ منها في تحديد رقم السطر الذي وقع فيه الخطأ ليتم إظهاره مع رسالة الخطأ.
- بعض المترجمات تحتفظ بنسخة أخرى لبرنامج المصدر مبيناً فيها الأخطاء حسب مواقعها في أسطر البرنامج.

Classes of Tokens أصناف العلامات

1. علامة كلمة أساسية/محجوزة وفي هذه الحالة غالباً يكون اسم العلامة يحمل نفس تشكيل حروف الكلمة الأساسية.
   - كلمة if تعطى علامة <if>و كلمة else تعطى العلامة <else>.
   - كلمة while تعطى علامة <while>
   - كلمة cout تعطى علامة <cout>
2. علامة مشغلات: أحياناً يتكون المشغل من رمز واحد كالعمليات الرياضية وبعض المقارنات, وأحياناً يتكون من أكثر من رمز كبعض عمليات المقارنة أو العمليات المنطقية:

<+>, <!=>, <&&>, <=>, <==>, <*>, <<=>

Classes of Tokens أصناف العلامات

3. علامة المعرفات Identifiers: وهي غالباً تحمل نفس الاسم id مع رقم تسلسلي يميز المعرفات عن بعضها.

3. الاسم الكامل للمعرفات مع نوعها يوجد بجدول الرموز.

- `Number = X + 10;`
- `<id,1> <=> <id,2> <+> <10> <;>`

4. علامة الثوابت constants: وهي تشمل الارقام بأنواعها (صحيح, حقيقي) والرموز الفردية char والنصوص literal strings المبينة بين علامتي تنصيص.

- `Name = “Ali Saleh”;`
- `<id,3> <=> <“Ali Saleh”> <;>`

Classes of Tokens أصناف العلامات

5. علامة إشارات النص Punctuation: لكل إشارة يمنح علامة, مثال لهذه الإشارات: الفاصلة , والفاصلة المنقوطة ; والقوس الأيسر) والقوس الأيمن (والقوس الملتوي الأيسر والأيمن { }.

- `printf(“Total = %d\n”, score);`
- تشكل العلامات كالتالي:

➢ `<printf> <(> <“Total = %d\n”> <,> <id,12> <)> <;>`

- لاحظ إشارات النص.

صفات العلامات

- علمنا أن الشكل العام للبطاقات يكون كالتالي:

<قيمة الصفة, اسم العلامة>

<token-name, attribute-value>

- اسم العلامة: رمز يبين نوع العلامة ويستفاد منه في طور تحليل النص.
- قيمة الصفة: تشير إلى قيمة بجدول الرموز تخص هذه العلامة Token.
- معلومات القيم بجدول الرموز تستخدم في أطوار التحليل والتجميع.

صفات العلامات

- أكثر من علامة Token قد تحتوي على نفس شكل المفردة Lexeme مثل المفردة id التي تعبر عن المعرفات. لذلك على محلل المفردات توفير معلومات كافية لبقية الأطوار ليتم التمييز بين المفردات.
- **لهذا**, في حالات كثيرة يوفر محلل المفردات, بالإضافة إلى اسم العلامة, قيمة لصفة توضح العلامة المنتجة.
- اسم العلامة يؤثر في عملية الإعراب (محلل النص), بينما قيمة الصفة تؤثر في عملية الترجمة أي في المراحل التي تلي الإعراب.

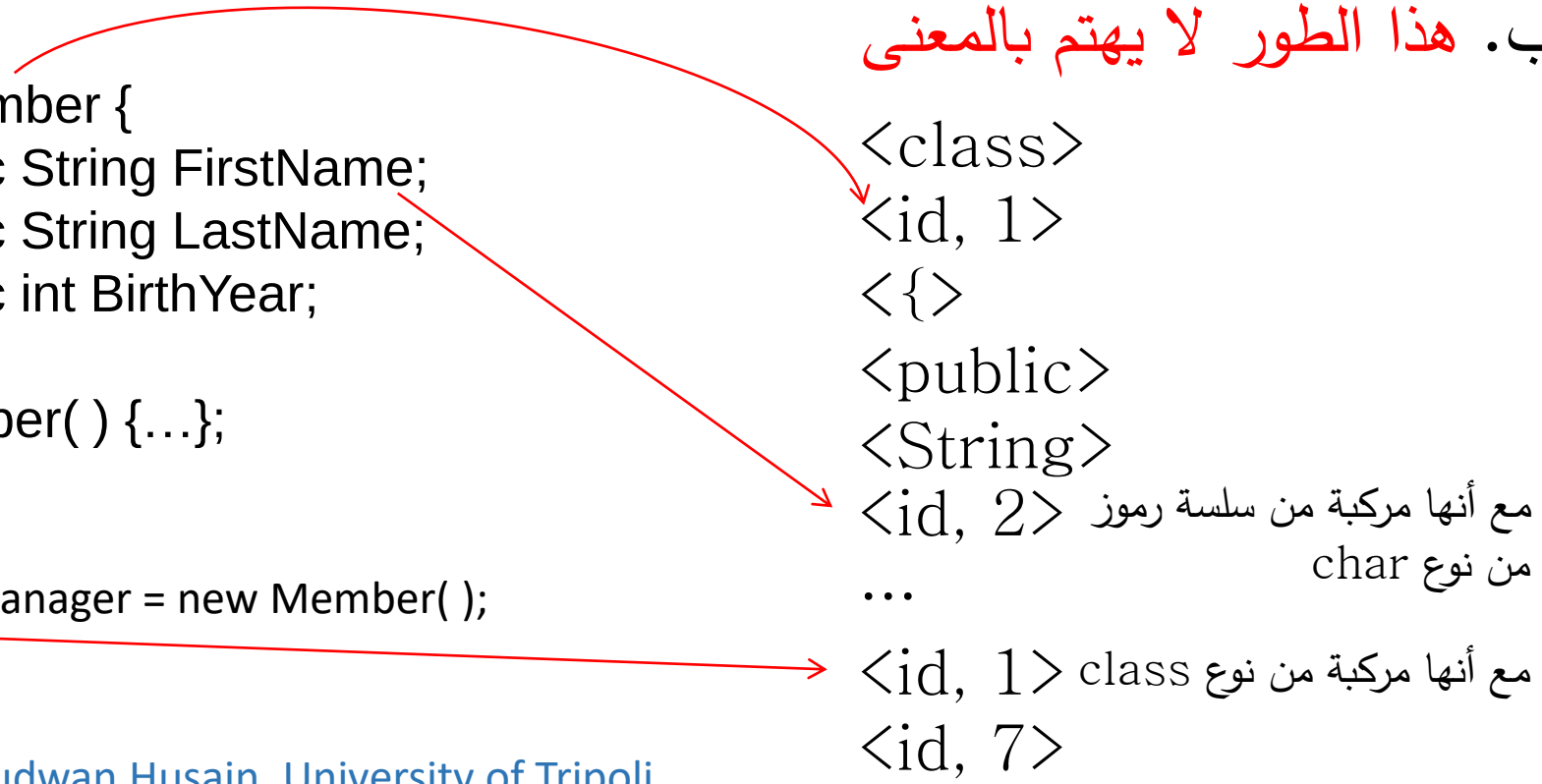
صفات العلامات

- العلامات لا تحتوي إلا على قيمة صفة واحدة على الأكثر.
- قد تكون هذه القيمة من نوع مركب Structure/class الذي بوسعه أن يشتمل على أكثر من قيمة بأنواع مختلفة, ولكن كلها تحت مسمى واحد من نوع هذا المركب. هذا الطور لا يهتم بالمعنى

```
class Member {
      public String FirstName;
      public String LastName;
      public int BirthYear;

      Member( ) {…};
};

Member  manager = new Member( );
```



صفات العلامات

- من أهم الأمثلة العلامة id التي تحتاج إلى توضيح نوع المتغير وعنوان أول ظهور له ببرنامج المصدر.
  - يستفاذ من هذا العنوان للإشارة إلى المتغير في حالة وجود رسالة خطأ في تعريفه.
  - وضح؟
  - المترجم يخاطب المبرمج حول المتغير باسم المتغير وليس باسم بطاقته أو عنوانه بجدول الموز
- أفضل قيمة تعطى للصفة العلامة هي عنوان لذلك المتغير بجدول الرموز.

مثال لعملية تحليل المفردات

مدخلات تحليل المفردات:

```
position = initial + rate * 60 ;
```

مخرجات تحليل المفردات:

<id, 1> <=> <id, 2> <+ > <id, 3> <*> <60> <;>

عنوان بجدول الرموز

جدول الرموز:

ID	name	Type	Address/ Line	Live
1	position	f	10	YES
2	initial	f	10	NO
3	rate	f	10	YES

Live يكتشف أن المتغير قيمته تتبدل مع متابعة تعليمات البرنامج, وإلا سيعتبر Not Live

عنوان بالبرنامج المصدري

تُهمل الفراغات التي تفصل بين الكلمات والمتغيرات.

التعامل مع أخطاء المفردات

- من الصعب على طور تحليل المفردات تحديد الخطأ في البرنامج المصدري دون مساعدة أدوات أخرى.
- على سبيل المثال: إذا تعرض المحلل إلى المفردة `fi` بدلاً من الكلمة المحجوزة `if` في حالة أول ظهور لها في برنامج بلغة `C` فإن المحلل سيعتبرها اسم لمتغير أو اسم دالة Function غير معرفة. لذلك سيمنحها علامة `id` ويترك التصحيح للأطوار التي تليه.
- في بعض الحالات يواجه محلل المفردات هذا الخطأ ولا يستطيع الاستمرار في التحليل نظراً لعدم تجانس المفردة الطارئة مع حروف أو جزء من حروف المفردات التالية في برنامج المصدر. إذا لم تتكرر `fi` سيعاملها كمتغير ميت

علاج أخطاء المفردات

- يستخدم في محلل المفردات عدد من الحلول لتفادي تلك الأخطاء:
- يمكن إضافة حقل في جدول الرموز ليبين أن المعرف جاري استخدامه أم لا

1. إلغاء حروف/رموز متتالية من بقية المفردات المدخلة القادمة من برنامج المصدر حتى أن يتم الحصول على مفردة متجانسة مع مسببة الخطأ.
2. إلغاء حرف/رمز واحد من بقية المفردات المدخلة.
3. إضافة حرف/رمز واحد إلى بقية المفردات المدخلة.
4. تغييرحرف/رمز واحد بآخر ببقية المفردات المدخلة.
5. تبادل مواقع حرفين/رمزين متتالين ببقية المفردات المدخلة.

موضوعنا التالي:

التعبيرات النظامية/الاعتيادية

REGULAR EXPRESSIONS