

1. Въведение. Обща организация на учебния компилатор

Цел на упражнението:

Упражнението дефинира задачата за построяване на учебен компилатор, като представя общата му структура, организацията на процеса на разработване и тестване.

Ръководството представя методика за създаване на учебен компилатор с входен процедурно ориентиран език, подмножество на езиците C и Java, транслиран в обектен език на виртуална стекова машина.

1.1. Входен, обектен и инструментален език на учебния компилатор

Входният език е процедурно ориентиран и е подмножество на езиците C и Java. Входната програма за компилатора представлява последователност от дефиниции на променливи и функции. Езикът на учебния компилатор е силно типизиран. Променливите, константите и изразите са свързани със специфичен тип. Входно-изходните операции изискват определен тип аргументи, аритметично-логическите операции са дефинирани за конкретни типове аргументи и тип на връщания резултат.

За опростяване на реализацията на учебния компилатор входният език има следните ограничения:

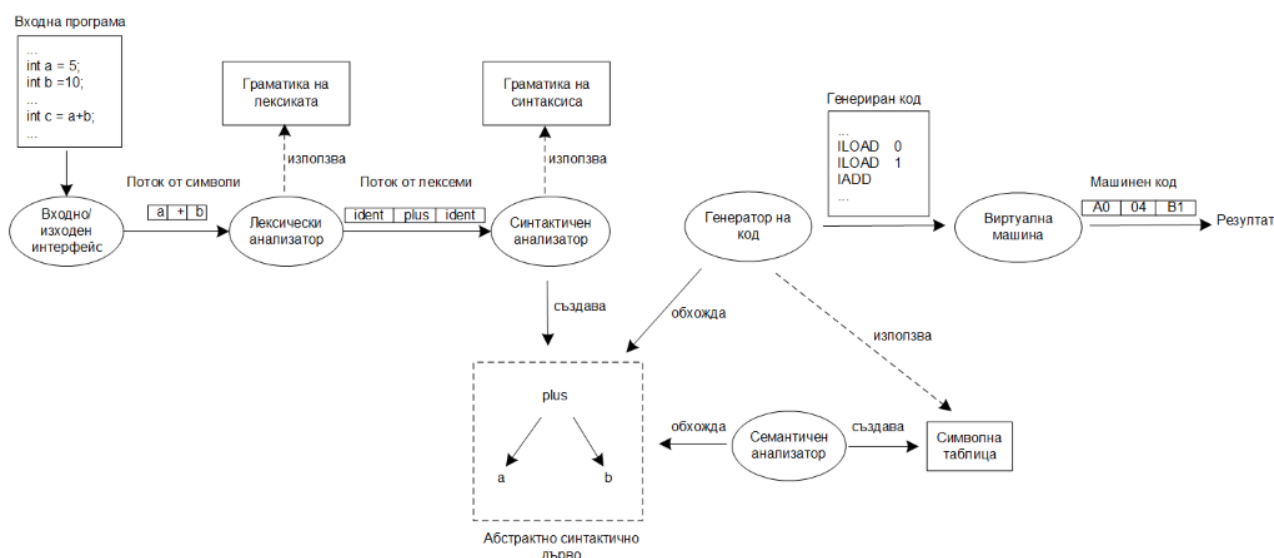
- В езика не са включени декларация на константи, етикети, типове данни.
- Поддържат се следните типове данни:
 - Скаларни типове данни - целочислен, символен и булев тип данни
 - Структурни типове данни - едномерен масив

Учебният компилатор генерира код за виртуална стекова машина. Генерираният код се изпълнява от подходящ интерпретатор. Виртуалната машина е абстрактна изчислителна машина, която има множество от инструкции и е независима от входния език на компилираната програма. Програмният модел на виртуалната машина включва три регистъра: програмен брояч, указател на върха на стека и базов указател. Стекът се използва за изчислителни цели и за осъществяване на обръщания към функции. Входният език има блокова структура, която изисква динамично разпределение на паметта. При влизане във всеки блок се създава фрагмент в стека, съдържащ указатели към извиквания и обхващащия блок и паметта за локалните променливи. При излизане от блока стековият фрагмент се унищожава и управлението се предава на извикващия блок. Реализиран е heap за съхранение на динамично създадени масиви.

Инструменталният език на учебния компилатор е Java. В лабораторните зали е инсталирана среда IntelliJ, но това не ограничава студентите да работят с други среди. Ръководството илюстрира работата на компилатора с подходящи примери и фрагменти.

1.2. Организация на учебния компилатор

Учебният компилатор има модулна структура (фиг. 1.1).



Фигура 1.1: Обща организация на учебния компилатор

- Входно изходен интерфейс – прочита текста на входната програма и го трансформира в поток от символи
- Лексически анализатор – прочита потока от символи, премахва незначещите символи и го трансформира в поток лексеми (ключови думи, идентификатори, оператори/разделители, константи).
- Синтактичен анализатор – прочита потока от лексеми, разпознава валидно изречение на входния език, формира абстрактно синтактично дърво.
- Семантичен анализатор – обхожда абстрактното синтактично дърво, проверява област на видимост (scope), съвместимост на типове данни и формира символна таблица
- Генератор на код – на базата на символната таблица синтактичното дърво генерира код за виртуалната машина
- Виртуална машина – прочита генерираните инструкции, изпълнява ги и произвежда резултат

Отделните модули са самостоятелни, като всеки модул е реализиран в отделен клас. Това позволява прилагане на ефективни методи за проектиране, реализация, тестване и настройки.

1.3. Тестване на компилатора

Модулната организация позволява удобно тестване и настройване на всеки относително самостоятелен модул. Всеки модул осъществява преобразуване на анализирания тестов пример (входна програма), като използва методите на съответния клас и генерира определен изход (изходен поток). Например тестовата програма за лексическия анализатор чете предварително преобразуваната от входно/изходния интерфейс входна програма и генерира последователност от разпознатите лексеми във входния поток.

Контролни въпроси:

1. Посочете входния, обектния и инструменталния език на учебния компилатор.
2. Кой модул от компилатора е отговорен за разпознаване на ключовите думи във входната програма?