



Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)



Кафедра «Информатики и защиты информации»

Курсовая работа на тему:

Разработка компилятора подмножества процедурного языка в ассемблер

*Специальность: 10.05.04 – Информационно-аналитические системы
безопасности*

**КОРОЛЕВА Александра Александровна,
ст. гр. ИСБ-118**



Введение

Цель работы:

разработать компилятор для подмножества
процедурного языка в ассемблер.



Описание компилятора и стадии разработки

Компилятор реализован на Java с использованием библиотек Antlr4 и ASM. Основная функция компилятора – проверка принадлежности исходной цепочки входному языку и генерация выходной цепочки символов в виде байт-кода для JVM.

- **Построение лексического анализатора**
- **Построение синтаксического анализатора**
- **Построение генератора объектного кода**



КС-грамматика

Грамматика строится на основе вложенных конструкций.

Инструкции объединяются в блоки, блоки помещаются внутри конструкций, которые могут содержать в себе эти инструкции.

Из таких блоков выстраивается шаблон, при совпадении с которым текст входного языка заполняет собой дерево, генерируемое библиотекой Antlr4 на основе этой грамматики.

```
grammar cring;
```

```
}program
```

```
    : function? mainProg EOF
```

```
}
```

```
}function
```

```
    : type FUNC IDENTIFIER OPAR par
```

```
}
```

```
}mainProg
```

```
    : MAIN block
```

```
}
```

```
}block
```

```
    : OBPACE (statement)* CBPACE
```

```
}
```

```
}statement
```

```
    : varDeclaration SCOL
```

```
    | callMethod SCOL
```

```
    | expression SCOL
```

```
    | print SCOL
```

```
    | ifStatement
```

```
    | assign
```

```
    | whileStatement
```

```
    | forStatement
```

```
    | breakStat SCOL
```

```
    | contStat SCOL
```

```
    | retStat SCOL
```

```
}
```



Таблица символов

Таблица символов реализована в классе, где переменная является ключом, а значением — ее тип.



Генерация кода и трансляция в объектный код

Генерация промежуточного кода и его трансляция в целевой код реализована с помощью библиотеки ASM для каждого элемента дерева. При помощи функций библиотеки можно записывать нужные инструкции в выходной поток при обходе дерева.



Пример работы компилятора

```
main() {
    int stroka = 100;
    int fros = 500;

    while (stroka < fros) {
        fros = fros - 2;
        break;
    }
    print(fros);
}
```

```
public class cring {
    public cring() {
    }

    public static void main(String[] var0) {
        byte var1 = 100;

        int var2;
        for(var2 = 500; var1 <= var2; var2 -= 2) {
        }

        System.out.print(var2);
    }
}
```

E:\IntelliJ IDEA Community Edition 2019.2.4\Kurs>java cring

98

Спасибо за внимание!

***КОРОЛЕВА Александра Александровна,
ст. гр. ИСБ-118***