ВОРОНКОВ С.О. / КАФЕДРА ВТ

# DSL “БАЗОВЫЙ КУРС”

# ЧАСТЬ 1 “ВВЕДЕНИЕ”

- Обзор
- Процессы / потоки / каналы / регулярные выражения
- Лексический анализ: лексема, токен, терминал
- Формальные грамматики
- Синтаксический анализ

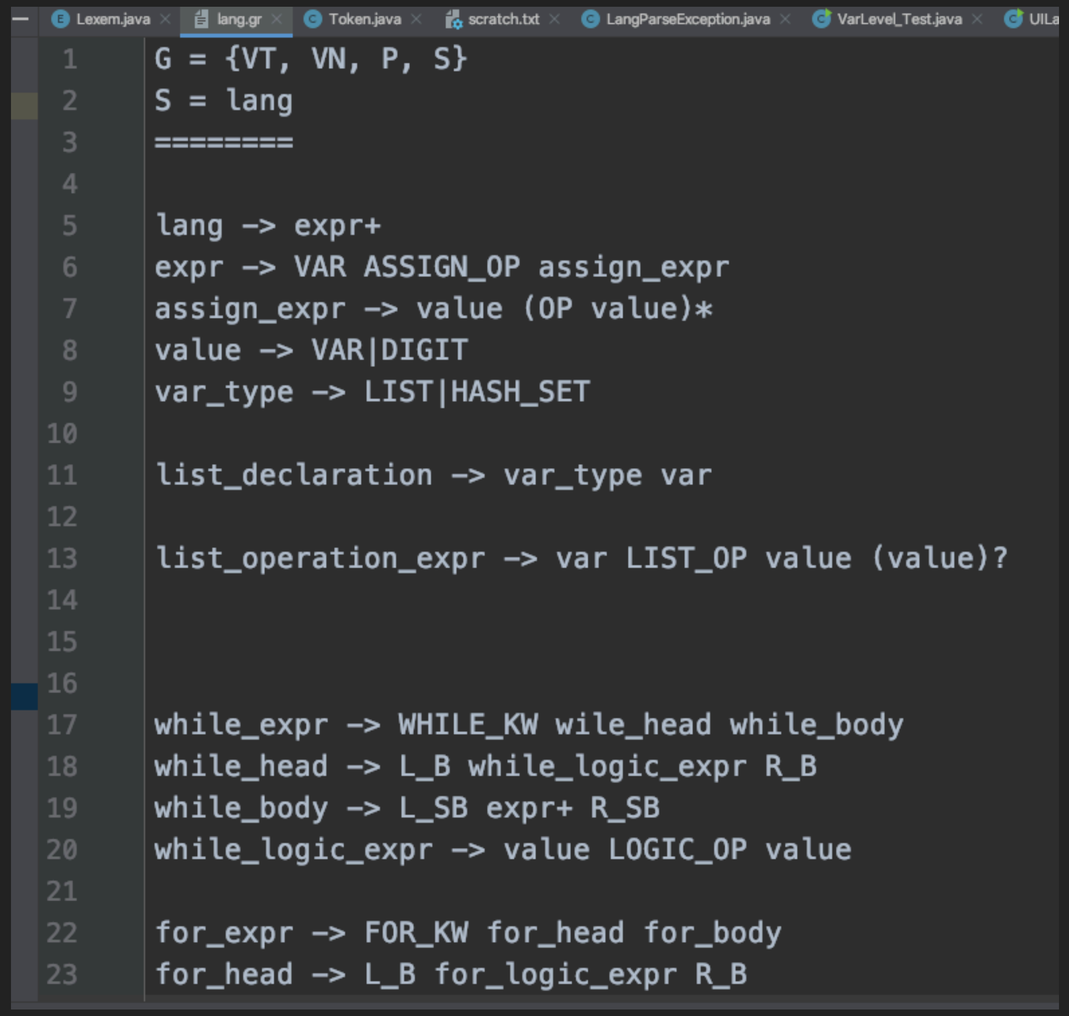
# ЧАСТЬ 2 “ТЕОРИЯ ТРАНСЛЯЦИИ”

- Лексический анализатор
- Синтаксический анализатор
- Внутреннее представление: триады, тетрады, ast (abstract syntax tree)
- RPN (revers polish notation)
- Алгоритм Дейкстры

# ЧАСТЬ 3 “ТЕОРИЯ ТРАНСЛЯЦИИ: РЕАЛИЗАЦИЯ”

- Интерпретация
- Стек машина
- Проектирование и разработка языка
- Реализация структур для данных: list, hashset
- Вопросы реализации многопоточности и оптимизации

# ЧАСТЬ 3 “ТЕОРИЯ ТРАНСЛЯЦИИ: РЕАЛИЗАЦИЯ”



# ЧАСТЬ 3 “ТЕОРИЯ ТРАНСЛЯЦИИ: РЕАЛИЗАЦИЯ”

ЛР 1 “ЛЕКСЕР”

```
new Token(Lexem.VAR, "a")
new Token(Lexem.ASSIGN_OP, "=")
new Token(Lexem.DIGIT, "1")
```

ЛР 2 “ПАРСЕР”

```
private void assignExpr() throws LangParseException {
    value();
    while (true) {
        op();
        value();
    }
}
```

ЛР 3 “ИНТЕРПРЕТАТОР”

```
a=1
b=1
b=a+1
while(a<10){
 a=a+2;
}
```

ЛР 4 “ТИПЫ ДАННЫХ”

```
Собственные типы данных

1) Связанный список

LIST a
a add 1
a add 10
```

# ТИПЫ ФАЙЛОВ / КАНАЛЫ / ПРОЦЕССЫ / ПОТОКИ / РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

# ТИПЫ ФАЙЛОВ

- Текстовые файлы : символьные и двоичные данные
- Каталоги (d)
- Символьные ссылки (l)
- Блочные устройства (b)
- Символьные устройства (c)
- Каналы (p)
- Сокеты (s)

# ТИПЫ ФАЙЛОВ: БЛОЧНЫЕ УСТРОЙСТВА

- Предназначены для обращения к аппаратному обеспечению
- Обращение к файлу устройства означает прозрачный вызов драйвера устройства
- К блочным устройствам обращаться можно произвольно
- Storage devices capable of buffering input/output

# ТИПЫ ФАЙЛОВ: СИМВОЛЬНЫЕ УСТРОЙСТВА

- Предназначены для обращения к аппаратному обеспечению
- Обращение к файлу устройства означает прозрачный вызов драйвера устройства
- К символьным устройствам обращаться можно только последовательно (посимвольно)
- Audio/graphics cards, keyboard/mouse

# ТИПЫ ФАЙЛОВ: КАНАЛЫ И СОКЕТЫ

- Channel is a connection between a process and a file that appears to the process as an unformatted stream of bytes. The kernel presents and accepts data from the channel as a process reads and writes that channel.  To a process then, all input and output operations are synchronous and unbuffered.
- Сокеты: nc -lU a.sock / echo 123 | nc -U a.sock

# ТИПЫ ФАЙЛОВ

```
lrwxrwxrwx 1 root     root        23 Apr 20  2014 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root     root      4096 Feb  7 12:05 vulkan
drwxr-xr-x 2 root     root      4096 Feb  7 12:08 w3m
-rw-r--r-- 1 root     root      4942 May  8  2018 wgetrc
drwxr-xr-x 7 root     root      4096 Feb  7 12:08 X11
-rw-r--r-- 1 root     root       642 Sep 24  2019 xattr.conf
drwxr-xr-x 4 root     root      4096 Feb  7 12:08 xdg
drwxr-xr-x 3 root     root      4096 Feb  7 12:09 xml
-rw-r--r-- 1 root     root       477 Jul 19  2015 zsh_command_not_found
```

```
brw-rw---- 1 root disk      1,   3 Feb  7 12:33 ram3
brw-rw---- 1 root disk      1,   4 Feb  7 12:33 ram4
brw-rw---- 1 root disk      1,   5 Feb  7 12:33 ram5
brw-rw---- 1 root disk      1,   6 Feb  7 12:33 ram6
brw-rw---- 1 root disk      1,   7 Feb  7 12:33 ram7
brw-rw---- 1 root disk      1,   8 Feb  7 12:33 ram8
brw-rw---- 1 root disk      1,   9 Feb  7 12:33 ram9
crw-rw-rw- 1 root root      1,   8 Feb  7 12:33 random
lrwxrwxrwx 1 root root           4 Feb  7 12:33 rtc -> rtc0
crw------- 1 root root    249,   0 Feb  7 12:33 rtc0
brw-rw---- 1 root disk      8,   0 Feb  7 12:33 sda
brw-rw---- 1 root disk      8,  16 Feb  7 12:33 sdb
crw-rw---- 1 root disk     21,   0 Feb  7 12:33 sg0
crw-rw---- 1 root disk     21,   1 Feb  7 12:33 sg1
drwxrwxrwt 2 root root         100 Feb  7 12:33 shm
crw------- 1 root root     10, 231 Feb  7 12:33 snapshot
lrwxrwxrwx 1 root root          15 Feb  7 12:33 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root          15 Feb  7 12:33 stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root          15 Feb  7 12:33 stdout -> /proc/self/fd/1
```

```
root@li705-29:/dev# file nbd7
nbd7: block special (43/224)
```

```
root@li705-29:/dev# file tty16
tty16: character special (4/16
```

# ПРОЦЕСС / ПОТОК

- Процесс: программа + контекст + данные
- Поток: нить выполнения внутри программы
- Самое простое отличие: у процессов область памяти разная, у потоков одна и та же

# РЕГУЛЯРНЫЕ ВЫРАЖЕНИЯ

- Done