

# Спецификация прототипа

v 1.0

## Основные свойства

- 1) С-подобный синтаксис
- 2) Программа в одной файле
- 3) Стандартная библиотека - read, write
- 4) Система типов - примитивные
  - а) int - 4-байтовый знаковый целый тип данных
- 5) Автоматический класс памяти
- 6) Программа состоит из списка функций без параметров и без возвращаемого значения

## Лексемы

### Разделители

пробел, перевод строки, табуляция

### Ключевые слова

void, int, if, else, while, return

### Литералы

Целочисленные: 0, 42, 256

### Операторы

Арифметика: +, - (бинарный), - (унарный) \*, /, %

Присваивание: =

Логические: ==, !=, <, >, <=, >=, &&, ||, !

Скобки: {}, ()

Служебные: ;

### Идентификаторы

a..zA..Z\_[a..zA..Z\_0..9]\*

Примеры: i, j, abc, x42, \_myVariable, foo, bar\_baz\_37

# СИНТАКСИС

*Функция* - **void name()** { **[op]\*** }, где:

- 1) name - идентификатор
- 2) op - оператор

*Оператор*:

- 1) **if (cond) { [op1]\* } [ else { [op2]\* } ]**, где:
  - a) cond - выражение
  - b) op1, op2 - операторы
- 2) **while (cond) { [op]\* }**, где:
  - a) cond - выражение
  - b) op - оператор
- 3) **int name;**, где
  - a) name - идентификатор
- 4) **x = value;**, где
  - a) x - имя переменной
  - b) value - выражение
- 5) **value;**, где
  - a) value - выражение

*Выражение* - набор операндов и операторов, поддающийся вычислению по RPN.

*Семантика*

- 1) Точка входа в программу - функция void main().
- 2) Встроенные функции:
  - a) write(x) - выводит значение x в зависимости от его типа
  - b) read(x) - считывает в x значение, в зависимости от его типа
- 3) Семантика арифметических операций и синтаксических структур C-подобная

# Байткод

Текстовый формат (для простоты отладки).

Список функций, каждая функция описывает количество необходимых регистров и список команд.

Точка входа в программу - функция с номером 0.

*Байткод*:

```
N // количество функций
F1 // описание функции
...
FN // описание функции
```

#### Описание функции:

name // имя функции  
X // количество регистров  
Z // количество команд  
C1 // команда  
...  
CZ // команда

#### Команда:

OP A1...An R (значение n и наличие R специфично команде)  
OP - идентификатор команды  
A1...An - номера регистров-операндов  
R - номер регистра-результата

## Список команд

{ IADD, ISUB, IMUL, IDIV, IMOD } A0 A1 R - целочисленные +, -, \*, /, %  
{ LAND, LOR, LNOT } A0 A1 R - логические &&, ||, !  
IMOV A0 R - присваивание значения из регистра A0 в R  
ILOAD X R - загрузка константы X в регистр  
{ ICMPEQ, ICMPLS } A0 A1 R - целочисленные сравнения  
GOTO X - безусловный переход на команду №X  
IF A0 X - переход на команду №X, если в A0 - не 0  
RET - выход из функции  
IWRITE A0 - вывод значения из регистра A0  
IREAD R - ввод значения в регистр R  
CALL N - вызов функции номер N

## Пример

```
void main() {  
    int number;  
    read(number);  
  
    int x;  
    int y;  
    int i;  
  
    i = 1;  
    x = 1;  
    y = 1;  
  
    while (i < number) {
```

```

        y = x + y;
        x = y - x;
        i = i + 1;
    }

    write(y);
}

```

```

1          // одна функция
main      // начинаем описывать функцию с именем main
5          // используем 5 регистров (0 - number, 1 - i, 2 - x, 3 - y, 4 - tmp)
14         // 14 команд
IREAD 0    // 00: read(number);
ILOAD 1 1  // 01: i = 1;
ILOAD 1 2  // 02: x = 1;
ILOAD 1 3  // 03: y = 1;
ICMPLS 1 0 4 // 04: tmp = i < number
LNOT 4 4   // 05: tmp = !tmp
IF 4 12    // 06: if tmp, goto 12
IADD 2 3 3  // 07: y = x + y
ISUB 3 2 2  // 08: x = y - x
ILOAD 1 4   // 09: tmp = 1
IADD 1 4 1  // 10: i = i + tmp
GOTO 4      // 11: goto 04
IWRITE 3    // 12: write(y)
RET         // 13: return

```