Эзотерический четырехзнаковый язык программирования

**HELLSMI**

Справочник по программированию

**1**

**Описание**

Эзотерический язык программирования “HELLSMI” является намеренно ограниченным в символах языком, в котором применяются лишь основные символы, используемые в чат-общении для выражения двух базовых эмоций: радость ( '**)**’ и '**0**' ) и печаль ( '**(**' и '**9**' ). Название языка образованно от слов “Hellish” (*англ. Адский)* и“Smile” (*англ. Улыбка)*, поскольку программирование на нем требует определенной усидчивости и исходный код немного напоминает выражение эмоций в чате или иных средствах обмена сообщениями.

**Синтаксис**

Любой исходный код программы начинается с метки «**()90**». Далее, без какого-либо разделителя, следует количество необходимых числовых блоков памяти для работы программы в особой записи. Как пример:

**((9(9)9)9)0)**

В данном случае под работу выделяется 21000 блоков памяти (т.е. от 0 до 20999) для хранения и обмена числовой информации в них. Количество скобок равно цифре, ноль обозначается обратной скобкой, символом ‘**9**’ осуществляется переход на младший разряд и метка «**0)**» сигнализирует о завершении строки. Если нам нужно отрицательное число, то к началу добавляется «**)0**» - запись знака минус. В данном виде оформляются все числовые строки в языке, поэтому для удобства в комплекте предоставляется программа–кодер (**hncode**), осуществляющая перевод из обычного числа в запись HELLSMI и наоборот.

После объявления количества памяти осуществляется переход на другую строку, с этого момента начинаются операторы, каждый из которых также заканчивается переводом на другую строку (кроме последнего). У всех команд есть стандартизованный синтаксис:

**XXXX{Первый параметр}…{Х параметр}**

Где XXXX – номер команды согласно списку доступных команд (на странице 3). Далее следуют числовые строки-параметры оформленные изложенной выше записью. Например, команда записи в блок памяти:

**0009)0)(9(((0)**

Команда («**0009**») производит запись в ячейку 0 («**)0)**») числа 13 («**(9(((0)**»).

**Общие сведения**

Класс языка: Эзотерический

Тип языка: Интерпретируемый (в будущем возможно и компилируемый)

Количество доступных символов: 4 ( ‘**9**’, ‘**0**’, ‘**(**‘, ‘**)**’ )

Количество доступных операторов: 24

Тип блоков памяти: Целочисленный (Integer)

Работа с символами: Через преобразование десятичных ASCII-кодов

**2**

**Список доступных команд**

* **0009 – Запись числового значения в указанный блок**

0009{Номер блока}{Числовое значение}

* **9000 – Чтение числового значения из блока и вывод на экран**

9000{Номер блока}

* **9((9 – Вывести на экран числовую последовательность (Разделитель – пробел)**

9((9{От блока}{До блока}

* **9999 – Увеличить значение блока на некоторое число**

9999{Номер блока}{Прибавляемое число}

* **0000 – Уменьшить значение блока на некоторое число**

0000{Номер блока}{Вычитаемое число}

* **0099 – Сложить два блока с записью результата в отдельный блок**

0099{Блок результата}{Первый слагаемый блок}{Второй слагаемый блок}

* **0900 – Вычесть один блок из другого с записью в отдельный блок**

0900{Блок результата}{Уменьшаемый блок}{Вычитаемый блок}

* **0909 – Умножить два блока с записью в отдельный блок**

0909{Блок результата}{Первый блок-множитель}{Второй блок-множитель}

* **0990 – Разделить нацело один блок на другой с записью в отдельный блок**

0990{Блок результата}{Блок делимого}{Блок делителя}

* **0999 – Разделить один блок на другой с записью остатка в отдельный блок**

0999{Блок результата}{Блок делимого}{Блок делителя}

* **9009 – Вывести на экран ASCII символ по десятичному коду из блока**

9009{Номер блока}

* **9))9 – Вывести на экран последовательность ASCII символов по десятичным кодам из блоков**

9))9{От блока}{До блока}

* **9090 – Скопировать значение одного блока в другой**

9090{Конечный блок}{Исходный блок}

* **9099 – Сброс программы (возврат на вторую строку)**

9099

* **9900 – Цикл WHILE**

9900{Первый блок условия}{Второй блок условия}(){Тип сравнения (Стр. 5)}

**3**

* **9909 – Условие IF**

9909{Первый блок условия}{Второй блок условия}(){Тип сравнения (Стр. 5)}

* **9990 – Переход на указанную строку программы**

9990{Номер строки программы}

* **0()0 – Ввод числового значения в блок с клавиатуры („?>”)**

0()0{Номер блока}

* **0)(0 – Ввод ASCII символа (в виде кода) в блок с клавиатуры („A?>”)**

0)(0{Номер блока}

* **9()9 – Ввод числовой последовательности в блоки с клавиатуры („S?>”) (Разделитель – пробел)**

9()9{От блока}{До блока}

* **9)(9 – Ввод последовательности ASCII символов в блоки с клавиатуры („SA?>”) (По блоку на символ, разделитель – пробел)**

9)(9{От блока}{До блока}

* **(09) – Окончание тела цикла WHILE**

(09)

* **(90) – Окончание тела условия IF**

(90)

* **0090 – Выход из программы**

0090

**4**

**Операторы сравнения**

* **()9 – Если равно**

В случае равенства значений блоков тело выполнится

* **()0 – Если не равно**

В случае неравенства значений блоков тело выполнится

* **()( – Если первое меньше второго**

В случае превосходства значения второго блока над первым тело выполнится

* **()) – Если первое больше второго**

В случае превосходства значения первого блока над вторым тело выполнится

* **()(( – Если первое меньше или равно второму**

В случае превосходства или равенства значения второго блока над первым тело выполнится

* **())) – Если первое больше или равно второму**

В случае превосходства или равенства значения первого блока над вторым тело выполнится

**5**