
ИНТЕРПРЕТАТОР НА SCHEME

Да се напише прост интерпретатор на Scheme, който поддържа най-често използваните функции за работа с числа и наредени двойки, както и специалните форми define, if, cond и lambda.

Целта на този проект е да се напише интерпретатор за езика Scheme, който позволява интерпретиране и изпълняване на код, подаден на програмата в символен вид. Този интерпретатор, естествено, може да не поддържа пълната функционалност на езика. Достатъчно е да бъдат поддържани най-често използваните математически оператори – за събиране, изваждане, умножение и деление (целочислено и прецизно) и функциите list, cons, car и cdr. Измежду специалните форми в езика се изисква коректната интерпретация на функциите define, lambda, както и условните оператори if и cond. Ще приемаме, че define и lambda могат да дефинират само функции, приемащи фиксиран брой аргументи. Програмата ще приема на входа си парче код в символен вид, което може да се представлява само списък от 0 или повече дефиниции и най-много едно извикване на някоя функция накрая. Резултатът от работата на програмата трябва да бъде само резултатът от дефиницията и коректното извикване на израза за оценяване (ако има такъв). Примерно извикване на програмата на езика Scheme би изглеждало по следния начин:

```
(define code1 '((define (f x) (+ x 2))
               (define x 5)
               (if (null? '()) (f x) (g x))))
(interpret code1)
→ 7
(define code2 '((define h (lambda (x) (quotient x 10))))
(interpret code2)
→ <няма значение какво връща в този случай>
```

За целите на този проект можем да считаме, че подаденият код е коректен. При приемане на некоректен код се допуска поведението на програмата да бъде „undefined behavior“. Всякакви предварителни проверки за коректност не са задължителни, но биха били поощрени с бонус към проекта. Друга възможност за бонус точки е функционалност тип read-eval-print, както в повечето интерпретатори на езика: след интерпретиране на кода потребителят да въвежда от стандартния вход изрази за оценяване, които да бъдат интерпретирани на базата на вече „компилирания“ код и резултатът да бъде върнат на стандартния изход. Използването на вградената функция eval в езика Scheme се допуска, но препоръчително да бъде избягвана освен в тривиални случаи (напр. за оценка на математически изрази). Отново, всякаква функционалност, която допълва изискванията по-горе, ще бъде възнаградена с бонус. Този проект също може да бъде реализиран на Scheme или Haskell.