



Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

ТЕМА ЗА ПРОЕКТ

към курс “Функционално програмиране”
за специалности Информатика, Компютърни науки (2 поток)
и избираема дисциплина
зимен семестър 2023/24 г.

Интерпретатор на Scheme

Synopsis: 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, не трябва да поддържа пълната функционалност на езика. Достатъчно е да бъдат поддържани най-често използваните математически операции, както и функциите `list`, `cons`, `car` и `cdr`. Измежду специалните форми в езика се изисква коректната интерпретация на специалните форми `define` и `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 последователност от комбинации в символен вид. Резултатът от работата на програмата трябва да бъде извеждането на оценките на всяка една от комбинациите на стандартния изход. Примерна употреба на програмата на езика Scheme би изглеждала по следния начин:

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

Да се поддържа режим на работа *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.

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 недефинирано. Всякакви предварителни проверки за коректност не са задължителни, но биха били поощрени с бонус към проекта. Използването на вградената функция `eval` в езика Scheme не се допуска. Реализирането на допълнителни функционалности и поддържането на по-голямо множество от езика Scheme ще бъдат възнаграждавани с бонус.