

Paradigmata programování 1

Instrukce:

- Řešení v podobě zdrojového kódu ve Scheme zasílejte v jediném souboru na e-mail

`eduard.bartl@upol.cz`.

- Jako předmět e-mailové zprávy uveďte

`PAPR1 -- reseni ukolu c. 1.`

- Do zdrojového kódu nepište ani příklady užití uvedené v zadání ani Vaše testy.
- Soubor se zdrojovým kódem pojmenujte jako

`UzivatskeJmeno.ss` (nebo `UzivatskeJmeno.rkt`),

kde `UzivatskeJmeno` je uživatelské jméno ve Vaší univerzitní e-mailové adrese.

Příklad: Student Jiří Novák s univerzitní e-mailovou adresou `jiri.novak02@upol.cz` si ví rady pouze s prvními třemi příklady v níže uvedeném zadání. Vytvoří tedy soubor `jiri.novak02.ss` (nebo `jiri.novak02.rkt`) s následujícím obsahem:

```
(+ (log e) 1)
```

```
(define x 1)
```

```
(define y 2)
```

```
(define z 3)
```

```
(defien w 4)
```

```
(define mynegative?
```

```
  (lambda (a b)
```

```
    (+ a b 1)))
```

Student nedostane žádný bod, protože první dva příklady nejsou správně. Třetí příklad (bez ohledu na to, zdali procedura funguje) nebude otestován, protože procedura není navázána na symbol, který je uveden v zadání (viz níže).

Upozornění: Nedodržení instrukcí může znamenat neuznání celého úkolu.

Úkol č. 1

datum zadání: 22. října 2012

termín odevzdání: 29. října 2012

1. **(1 bod)** Přepište následující výraz do prefixové notace.

$$\frac{\ln(e) + 1}{\sqrt{4} + \frac{10}{-1+6}}$$

Pozn.: Procedura počítající přirozený logaritmus je navázaná na symbol `log`.

2. **(2 body)** Vytvořte vazby na symboly `x`, `y`, `z`, `w` tak, aby následující symbolické výrazy vracely dané hodnoty:

```
> (x y z)
3
> (w y (x y y))
-1
```

3. **(2 body)** Vytvořte proceduru `pyramid` počítající povrch pravidelného čtyřbokého jehlanu. Tato procedura očekává dva vstupní parametry, délku uhlopříčky podstavy d a výšku v jehlanu. Pro výpočet povrchu P využijte tento vztah:

$$P = a \cdot (a + \sqrt{4v^2 + a^2}),$$

kde

$$a = \frac{\sqrt{2}}{2} \cdot d.$$

Příklad použití:

```
> (pyramid 1 1)
2.0
```

4. **(3 body)** Vytvořte predikát `my-negative?`, který zjistí, zdali je argument záporné číslo. Příklady použití:

```
> (my-negative? -2)
#t
> (my-negative? 4)
#f
```

5. (**3 body**) Vytvořte proceduru `my-proc` realizující funkci f definovanou následovně:

$$f(x) = \begin{cases} x + 2 & \text{pro } x > 0, \\ 0 & \text{pro } x = 0, \\ x - 2 & \text{pro } x < 0. \end{cases}$$

Příklady použití:

```
> (my-proc 1)
3
> (my-proc 0)
0
> (my-proc -1)
-3
```

6. (**4 body**) Vytvořte proceduru `implies`, která realizuje pravdivostní funkci logické spojky implikace. Příklady použití:

```
> (implies #f #f)
#t
> (implies #f #t)
#t
> (implies #t (> 2 3))
#f
> (implies + (< 2 3))
#t
```

7. (**5 body**) Vytvořte proceduru `my-even`, která očekává jeden číselný argument a vrací jeho hodnotu, jestliže je tento argument sudý. V opačném případě vrací `#f`. Není dovoleno použít procedury `modulo`, `remainder`, `odd?` ani `even?`. Dále není dovoleno použít speciální formu `if`. Příklady použití:

```
> (my-even 2)
2
> (my-even -4)
-4
> (my-even 1)
#f
> (my-even -5)
#f
```