

## Paradigmata programování 3

# Knihovna *micro-graphics*

verze z 12. září 2022

Knihovna *micro-graphics* je jednoduchá procedurální grafická knihovna pro prostředí LispWorks určená k praktickému procvičování objektového programování.

Knihovna se do prostředí LispWorks načítá tak, že se načte (funkcí `load` nebo z menu) soubor `load.lisp` (musí být ve stejném adresáři jako soubory `micro-graphics.lisp` a `package.lisp`). K dispozici dává následující funkce:

## 1 Základní funkce

```
(mg:display-window) => window
```

*window*: odkaz na okno

Vytvoří a zobrazí nové grafické okno. Jako výsledek vrátí odkaz na toto okno, který je třeba používat jako argument v ostatních funkcích, jež s oknem pracují. Nové okno má několik kreslicích parametrů, které lze zjišťovat pomocí funkce `mg:get-param` a nastavovat pomocí funkce `mg:set-param`. Pomocí funkce `mg:get-callback` lze také zjišťovat a pomocí funkce `mg:set-callback` nastavovat zpětná volání. Souřadnice v okně se udávají v pixelech, jejich počátek je v levém horním rohu okna, hodnoty druhé souřadnice se zvětšují směrem dolů. Rozměry okna jsou 297 na 210 pixelů.

```
(mg:get-param window param) => value
```

*window*: hodnota vrácená funkcí `mg:display-window`

*param*: symbol

Funkce `mg:get-param` vrátí hodnotu kreslicího parametru *param* okna *window*. Pro nás jsou důležité tyto parametry:

**:thickness** Tloušťka čáry v pixelech. Ovlivňuje funkce na kreslení obrazců (např. `mg:draw-circle`), pokud není nastaven parametr `:filledp`. Počáteční hodnota: 1.

<b>:foreground</b>	Barva inkoustu. Ovlivňuje funkce na kreslení obrazců (např. <code>mg:draw-circle</code> ). Počáteční hodnota: <code>:black</code> .
<b>:background</b>	Barva pozadí. Ovlivňuje funkci <code>mg:clear</code> . Počáteční hodnota: <code>:white</code> .
<b>:filledp</b>	Zda kreslit obrazce vyplněné. Ovlivňuje funkce na kreslení obrazců (např. <code>mg:draw-circle</code> ). Počáteční hodnota: <code>nil</code> .
<b>:closedp</b>	Zda spojit poslední a první vrchol polygonu. Ovlivňuje funkci <code>mg:draw-polygon</code> , pokud není nastaven parametr <code>filledp</code> . Počáteční hodnota: <code>nil</code> .

Přípustnými hodnotami parametrů `:foreground` a `:background` jsou všechny symboly, které v grafickém systému LispWorks pojmenovávají barvu. Jejich seznam lze zjistit funkcí `color:get-all-color-names`, nebo, pokud uvedeme část názvu barvy, kterou chceme použít, funkcí `color:apropos-color-names`. Vzorkovník barev je také ve zvláštním souboru.

Barvy lze také v LispWorks vytvářet z komponent pomocí zabudovaných funkcí `color:make-rgb`, `color:make-hsv`, `color:make-gray`. Zájemci se mohou na tyto funkce podívat do dokumentace.

Kreslicí parametry lze nastavovat funkcí `mg:set-param`.

```
(mg:set-param window param value) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`  
*param*: symbol  
*value*: hodnota

Funkce `mg:set-param` nastavuje kreslicí parametr *param* okna *window* na hodnotu *value*. Význam kreslicích parametrů je uveden u funkce `mg:get-param`. Nové kreslicí parametry ovlivňují způsob kreslení do okna od momentu, kdy byly nastaveny.

```
(mg:drawing window . body) => nil (makro)
```

*window*: hodnota vrácená funkcí `mg:display-window`  
*body*: výrazy

Vytvoří prostředí, ve kterém je možné kreslit do okna *window*. V knihovně `micro-graphics` je zakázáno používat jakékoli funkce na kreslení do okna mimo jeho zpětné volání typu `:display`. Na Mac OS vede pokus o takové kreslení k chybám, jinde sice ne, ale nevíme, zda se něco špatného stejně neděje. Pokud okno zpětné volání typu `:display` nemá, je možné také použít makro `mg:drawing`. Makro dočasně vytvoří zpětné volání, během kterého se vyhodnotí výrazy jeho těla.

```
(mg:clear window) => nil
```

*window*: hodnota vrácená funkcí mg:display-window

Funkce mg:clear vymaže celé okno *window* barvou aktuálně uloženou v kreslicím parametru :background.

Všechny kreslicí funkce je povoleno používat jen uvnitř zpětného volání :display nebo makra mg:drawing.

```
(mg:draw-circle window x y r) => nil
```

*window*: hodnota vrácená funkcí mg:display-window

*x, y, r*: čísla

Funkce mg:draw-circle nakreslí do okna *window* kruh se středem o souřadnicích *x, y* a poloměrem *r*. Kruh se kreslí barvou uloženou v kreslicím parametru :foreground okna *window*. Kreslicí parametr :filledp okna *window* udává, zda se bude kruh kreslit vyplněný. Pokud není nastaven, bude se kreslit pouze obvodová kružnice čarou, jejíž tloušťka je uložena v kreslicím parametru :thickness okna *window*.

Všechny kreslicí funkce je povoleno používat jen uvnitř zpětného volání :display nebo makra mg:drawing.

```
(mg:draw-ellipse window x y rx ry phi) => nil
```

*window*: hodnota vrácená funkcí mg:display-window

*x, y, rx, ry, phi*: čísla

Funkce mg:draw-ellipse nakreslí do okna *window* elipsu se středem o souřadnicích *x, y* s poloosami *rx* a *ry*. Elipsa bude natočená (kolem středu) o úhel *phi*. Kreslí se barvou uloženou v kreslicím parametru :foreground okna *window*. Kreslicí parametr :filledp okna *window* udává, zda se bude elipsa kreslit vyplněná. Pokud není nastaven, bude se kreslit pouze obvod čarou, jejíž tloušťka je uložena v kreslicím parametru :thickness okna *window*.

Všechny kreslicí funkce je povoleno používat jen uvnitř zpětného volání :display nebo makra mg:drawing.

```
(mg:draw-polygon window points) => nil
```

*window*: hodnota vrácená funkcí mg:display-window

*points*: seznam čísel

Funkce mp:draw-polygon nakreslí do okna *window* polygon s vrcholy danými argumentem *points*. Tento argument musí obsahovat seznam sudé délky, jako prvky

se v něm musí střídát *x*ové a *y*ové souřadnice vrcholů polygonu. Kreslí se barvou uloženou v kreslicím parametru `:foreground` okna *window*. Kreslicí parametr `:filledp` okna *window* udává, zda se bude polygon kreslit vyplněný. Pokud není nastaven, budou se kreslit pouze úsečky spojující jednotlivé vrcholy polygonu čarou, jejíž tloušťka je uložena v kreslicím parametru `:thickness` okna *window*. Kreslicí parametr `:closedp` okna *window* určuje, zda se má nakreslit i úsečka spojující poslední bod polygonu s prvním. Pokud je nastaven kreslicí parametr `:filledp`, kreslicí parametr `:closedp` se ignoruje.

Všechny kreslicí funkce je povoleno používat jen uvnitř zpětného volání `:display` nebo makra `mg:drawing`.

## 2 Zpětná volání

```
(mg:get-callback window callback-name) => callback
```

*window*: hodnota vrácená funkcí `mg:display-window`  
*callback-name*: symbol, přípustné hodnoty viz níže

Vrací zpětné volání názvu *callback-name* instalované v okně *window*. *callback-name* musí být název zpětného volání (např. `:display` nebo `:mouse-down`). Zpětná volání lze také nastavit pomocí funkce `mg:set-callback`.

Popis základních typů zpětných volání:

`:display`: funkce s jedním argumentem. Knihovna tuto funkci volá kdykoliv je nutno okno překreslit. Jako argument použije odkaz na okno (vrácený funkcí `mg:display-window`).

`:mouse-down`: funkce akceptující čtyři argumenty: odkaz na okno (vrácený funkcí `mg:display-window`), označení stisknutého tlačítka myši (jeden ze symbolů `:left`, `:center`, `:right`) a vodorovnou a svislou souřadnici myši. Zpětné volání knihovna zavolá při stisku tlačítka myši v okně.

```
(mg:set-callback window callback-name callback) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`  
*callback-name*: symbol, přípustné hodnoty viz výše

Nastavuje zpětné volání *callback-name* okna *window* na funkci *callback*. Popis zpětných volání najdete u funkce `mg:get-callback`.

```
(mg:invalidate window) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`

Oznámí knihovně `micro-graphics`, že okno *window* má neaktuální obsah. Knihovna pak ve vhodný moment zavolá zpětné volání `:display`.

### 3 Další funkce

```
(mg:get-size window) => (width height)
```

*window*: hodnota vrácená funkcí `mg:display-window`  
*width*, *height*: čísla

Vrátí ve dvouprvkovém seznamu šířku a výšku okna *window*.

```
(mg:set-size window width height) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`  
*width*, *height*: čísla

Nastaví šířku a výšku okna *window*.

```
(mg:get-drawing-size window) => (width height)
```

*window*: hodnota vrácená funkcí `mg:display-window`  
*width*, *height*: čísla

Vrátí ve dvouprvkovém seznamu šířku a výšku kreslicí plochy okna *window* (obecně jsou to menší hodnoty než u funkce `mg:get-size`).

```
(mg:close-window window) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`

Uvolní systémové prostředky alokované pro okno *window* a okno vymaže z obrazovky. Předtím zavolá zpětné volání `:destroy` (viz níže).

```
(mg:get-string-extent string)  
=> (left top right bottom)
```

*string*: řetězec

Vrátí rozměry, které by měl řetězec *string* po vykreslení do okna na souřadnice (0 0). Hodnota *left* je většinou rovna nule, hodnota *top* záporná.

```
(mg:draw-string window string x y &optional phi) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`

*string*: řetězec

*x, y*: čísla

*phi*: číslo; nepovinný argument s defaultní hodnotou 0

Vykreslí řetězec *string* do okna *window* na souřadnice *x y*. Řetězec může mít více řádků, ale odřádkovávat je nutné standardním způsobem (např. direktivou `~%`). Je-li zadáno *phi*, udává úhel, pod kterým bude řetězec vykreslen.

Všechny kreslicí funkce je povoleno používat jen uvnitř zpětného volání `:display` nebo makra `mg:drawing`.

```
(mg:path name) => path
```

*name*: řetězec

Vrátí cestu k souboru, jehož název je *name*. Chybějící části cesty se doplní podle cesty k souboru, ve kterém se vyhodnocovaný výraz `(mg:path name)` vyskytuje. V řetězci *name* lze použít, jak je běžné, lomítka `/` (nikoliv zpětná) jako oddělovače a dvě tečky `..` jako přechod o úroveň výš. Aby funkce vracela správný výsledek, je povoleno používat ji jen v kombinaci s makrem `defvar` (nebo `defparameter`) takto:

```
(defvar *my-picture* (mg:path "my-picture.jpg"))
```

nebo takto:

```
(defvar *my-picture*)  
(setf *my-picture* (mg:path "my-picture.jpg"))
```

Funkci `mg:path` lze použít k nalezení cesty k souborům s obrázkem nebo zvukem, které jsou ve stejném adresáři jako zdrojový soubor.

```
(mg:draw-image window path x y) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`

*path*: řetězec

*x, y*: čísla

Vykreslí obrázek do okna *window*. Levý horní roh obrázku bude v bodě o souřadnicích *x, y*. Obrázek musí být uložen v souboru s cestou *path*. K vytvoření cesty je možné použít funkci `mg:path`.

Všechny kreslicí funkce je povoleno používat jen uvnitř zpětného volání `:display` nebo makra `mg:drawing`.

```
(mg:get-image-size window path) => (w h)
```

*window*: hodnota vrácená funkcí `mg:display-window`

*path*: řetězec

Vrátí rozměry obrázku uloženém v souboru s cestou *path*. K vytvoření cesty je možné použít funkci `mg:path`.

```
(mg:play-sound window path) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`

*path*: řetězec

Spustí přehrávání zvuku uloženého v souboru s cestou *path*. K vytvoření cesty je možné použít funkci `mg:path`. Zvuk je spuštěn asynchronně. Je svázán s oknem *window*, při zavření okna přestane hrát. Funkci lze použít pouze ve Windows a Mac OS. Přehrává pouze některé typy zvukových souborů.

```
(mg:stop-sound window path) => nil
```

*window*: hodnota vrácená funkcí `mg:display-window`

*path*: řetězec

Zastaví přehrávání zvuku uloženého v souboru s cestou *path*. K vytvoření cesty je možné použít funkci `mg:path`. Funkci lze použít pouze ve Windows a Mac OS.

```
(mg:translate-color object) => spec
```

*object*: libovolná hodnota

Vrátí reprezentaci barvy *object* použitelnou jako vstup pro funkce grafického systému LispWorks pracující s barvami (např. `color:color-hue`). Pokud *object* neoznačuje barvu, vrátí `nil`. Funkci lze použít k testování, zda hodnota *object* označuje barvu.

```
(mg:add-timer-callback window seconds callback)  
=> timer
```

*window*: hodnota vrácená funkcí `mg:display-window`

*seconds*: kladné reálné číslo

*callback*: funkce

Naplánuje zavolání funkce *callback* v čase daném hodnotou *seconds*. Jako výsledek vrátí hodnotu, kterou lze použít jako argument funkce *mg:remove-callback*. S touto hodnotou (jako druhým argumentem) bude také funkce *callback* volána. Prvním argumentem je okno. Funkce *callback* by měla jako výsledek vrátit kladné číslo nebo *nil*. V prvním případě dojde k jejímu dalšímu zavolání za počet sekund daný návratovou hodnotou. Takto jde zařídit volání funkce v daných intervalech donekonečna. Ve druhém případě se opakované volání funkce ukončí. To bude také ukončeno po zavření okna.

```
(mg:remove-timer-callback window timer) => nil
```

*window*: hodnota vrácená funkcí *mg:display-window*

*timer*: hodnota vrácená funkcí *mg:add-timer-callback*

Ukončí opakované volání funkce zaregistrované funkcí *mg:add-timer-callback*.

## 4 Další kreslicí parametry a zpětná volání

### 4.1 Kreslicí parametry

**:mask** Udává část okna, na kterou se omezí kreslení. Může být *nil* (bez omezení), nebo seznam definujících body polygonu (pro formát tohoto seznamu viz funkci *mg:draw-polygon*). Kreslení je pak omezeno na vnitřní body tohoto polygonu. Ovlivňuje všechny kreslicí funkce.

### 4.2 Zpětná volání

**:mouse-up**, **:mouse-move**, **:double-click**: stejný význam jako **:mouse-down**, liší se pouze situacemi, za nichž je knihovna volá. Zpětné volání **:mouse-move** může být navíc voláno s druhým argumentem rovným *nil*.

**:destroy**: funkce akceptující jeden argument. Knihovna tuto funkci zavolá při zavření okna uživatelem nebo funkcí *mg:close-window*. Jako parametr použije odkaz na okno (vrácený funkcí *mg:display-window*).

**:activate**: funkce akceptující dva argumenty. Knihovna tuto funkci zavolá při aktivaci resp. deaktivaci okna. O kterou z těchto možností jde, určuje druhý parametr (*Pravda*: aktivováno, *Nepravda*: deaktivováno). První parametr je okno.

**:resize**: funkce akceptující tři argumenty. Knihovna tuto funkci volá, kdykoliv dojde ke změně rozměrů okna. Jako argumenty použije okno (vrácené funkcí *mg:display-window*) a nové rozměry jeho obsahu (šířku a výšku v pixelech).

**:character**: funkce akceptující dva argumenty. Knihovna tuto funkci volá, kdykoliv uživatel stiskne klávesu. Prvním argumentem je okno, druhým znak odpoví-



dající klávese.