



Úvod do programovacích stylů Manuál ke knihovně proto

verze z 23. října 2023

Knihovnu tvoří jediný soubor `proto.py`. Soubor se musí nalézat v téže adresáři jako program, který jej používá. Pro použití stačí importovat modul:

```
from proto import obj
```

Hodnota proměnné `obj` se nazývá *základní objekt*. Nové objekty vytváříme *klonováním* existujících objektů.

Objektu můžeme pouze zasílat zprávy. Výraz

```
receiver.message(arg1, arg2, ...) => value
```

zašle objektu *receiver* zprávu *message* s argumenty *arg1*, *arg2*, ... Hodnota *value* je návratovou hodnotou zasláné zprávy.

Objekt je popsán *sloty*, kde slot je pojmenovaná položka objektu. Objekt *parent* nazýváme *rodičem* objektu *object*, jestliže *parent* je hodnotou slotu *parent* objektu *object*. Všechny objekty kromě `obj` mají rodiče.

Po zaslání zprávy *message* objektu *receiver* s argumenty *arg1*, *arg2*, ... se nejprve hledá *obsluha* zprávy a poté se obsluha *vyhodnotí*. Pokud se nalezne obsluha zprávy, říkáme, že objekt zprávě *rozumí*, jinak zaslání zprávy skončí chybou.

Hledání obsluhy zprávy *message* v objektu *object* probíhá následovně.

1. Pokud objekt *object* má slot *message*, pak je obsluhou hodnota slotu.
2. Pokud objekt *object* nemá slot *message*, ale má rodiče *parent*, pak se rekurzivně hledá obsluha zprávy v rodiči.
3. Jinak zpráva nemá obsluhu.

Vyhodnocení obsluhy *handler* nalezené v objektu *owner* pro zprávu *message* s argumenty *arg1*, *arg2*, ... zasláné objektu *receiver* probíhá následovně.

1. Pokud je *handler* funkce, pak se *handler* zavolá s argumenty *call_super*, *receiver*, *arg1*, *arg2*, ... Hodnota *call_super* je funkce bez argumentů, která přepošle zprávu *message* zaslánou *receiver* rodiči objektu *owner* s argumenty *arg1*, *arg2*, ...

2. Jinak je hodnotou obsluhy přímo *handler*.

Přeposlání zprávy *message* zaslané objekt *receiver* s argumenty *arg1*, *arg2*, ... objektu *object* probíhá tak, že se nejprve nalezne obsluha zprávy *message* v objektu *object* a poté se obsluha vyhodnotí.

Následuje popis zpráv, kterým rozumí základní objekt.

```
object.clone() => clone
```

Vytvoří nový objekt *clone*, který bude mít jediný slot *parent* s hodnotou *object*.

```
object.set_slot(slot, value) => object
```

Nastaví hodnotu slotu *slot* objektu *object* na *value*. Pokud zadaný slot neexistuje, vytvoří se nový.

```
object.add_slot(slot) => object
```

Přidá sloty *slot* s hodnotou *None* a *set_slot* s hodnotou *set_method* do objektu *object*, kde *set_method* je metoda očekávající jeden argument *value*, která provede nastavení slotu *slot* na hodnotu *value*.

```
object.print() => object
```

Vytiskne objekt *object* včetně jeho slotů.

```
object.understand(message) => bool
```

Rozhodne, zda objekt *object* rozumí zprávě *message*.

```
object.send(message, arg1, arg2, ...) => value
```

Zašle objektu *object* zprávu *message* s argumenty *arg1*, *arg2*, ... Hodnota *value* je návratovou hodnotou zaslané zprávy.

```
object.name() => name
```

Řetězec *name* je názvem objektu *object*.

```
object.set_name(name) => object
```

Nastaví slot `name` objektu *object* na hodnotu *name*.