

Není-li uvedeno jinak, zakroužkujte jedinou správnou odpověď. Všechny příklady jsou v jazyce Python 2.6 s knihovnou gtk2. Případné chyby zjevně plynoucí z rozdílů mezi verzemi pythonu nebudou počítány. OMG, test má i druhou stranu!

1) Spustili jsme Glade, nakreslili jsme v něm okno a v okně tlačítko se jménem `button1`. Pak jsme napsali program:

```
#!/usr/bin/env python
import gtk, sys
class Main:
    def __init__(self):
        self.window1 = gtk.Window()
        self.builder = gtk.Builder()
        self.builder.add_from_file(sys.path[0] + "/" + "gui.xml")
        self.builder.connect_signals(self)
        self.w("window1").connect('destroy', lambda w: gtk.main_quit())
        self.w("window1").show_all()
    def w(self, widgetname):
        return self.builder.get_object(widgetname)
    # GUI event handlers
    def on_button1_clicked(self, sender):
        print "KLIK!"
        pass
def myerr(exc_type, exc_value, tb):
    import traceback;
    traceback.print_exception(exc_type, exc_value, tb)
    raw_input()
sys.excepthook = myerr
MainInstance = Main()
gtk.main()
```

Jenomže program nefunguje, jak čekáme. Jde ho spustit, ale při kliknutí na tlačítko se neděje nic, do textového okna se nevypíše KLIK! ani nic jiného.

a) Zapomněli jsme do funkce `on_button1_clicked()` vložit volání funkce

`builder.connect_signals(self)`.

b) Zapomněli jsme v Glade definovat obsluhu události `clicked` pro tlačítko `button1`.

c) Takto to nepůjde, definice funkce `on_button1_clicked()` je na špatném místě v programu.

2) Co znamená, když řekneme, že je *kvantifikátor* `*` v *regulárních výrazech* takzvaně *hladový* (greedy)?

a) Musí být vždy uzavřen v hranatých závorkách, jinak se regulární výraz nevyhodnotí.

b) Program se snaží najít co nejvíce výskytů předchozího znaku, co reg. výraz dovoluje.

c) Program se snaží najít co nejméně výskytů předchozího znaku, co reg. výraz dovoluje.

d) Při použití v programu vám každá hvězdička sežere náhodný soubor na disku.

3) Podívejte se na následující program, který využívá regulární výrazy ke zpracování textu. Co program zřejmě dělá?

```
import re
a = "However, the novel also deals with broader antiwar themes: essentially
a series of absurdly comic episodes, it explores both the pointlessness and
futility of conflict in general and of military discipline, specifically
Austrian military discipline, in particular. Many of its characters,
especially the Czechs, are participating in a conflict they do not
understand on behalf of a country to which they have no loyalty."
b = 0
c = 0
for matchobject in re.finditer("\s[a-zA-Z][a-zA-Z]*\s", a):
    b += matchobject.end() - matchobject.start() - 2
    c += 1
print float(b)/c
```

- a) Program vypíše, který nejčastější znak je v textu.
- b) Program vypíše, kolik je v textu interpunkčních znaků (čárky, tečky...)
- c) Program vypíše průměrnou délku slova v řetězci.**

4) Vidíte, jak matoucí je, když v programu nejsou dobře pojmenované proměnné? (0 b.)

- a) Ano, je to hrozné.
- b) Ne.

5) Hovořili jsme o widgetech gtkTreeView a gtkTreeStore. K čemu slouží?

- a) TreeView zobrazuje tabulku/stromový seznam s daty, TreeStore pro něj skladuje hodnoty. TreeView musí mít určený svůj TreeStore.**
- b) TreeStore zobrazuje tabulku/stromový seznam s daty, TreeView pro něj skladuje hodnoty. TreeStore automaticky nalezne příslušný TreeView.
- c) TreeView i TreeStore dělají skoro totéž, ale TreeView dovede navíc vykreslovat obrázky.