

Wstęp do programowania w języku C

Marek Piotrów - Wykład 10
Wprowadzenie do GTK+ - biblioteki okienkowego interfejsu
w C

10 grudnia 2014

Biblioteka GTK+

- GTK+ jest darmową biblioteką przeznaczoną do tworzenia graficznego (okienkowego) interfejsu użytkownika.
- Została napisana w C na potrzeby programu GIMP (stąd nazwa Gimp ToolKit), ale ma strukturę obiektową.
- Można z niej korzystać w wielu językach programowania (C, C++, C#, Ada, Perl, itd.).
- Programy napisane z użyciem tej biblioteki są przenośne pomiędzy różnymi systemami operacyjnymi.
- GTK+ korzysta z m.in. bibliotek: GLib, GObject, GIO, Pango, ATK i GDK (trzeba je zainstalować, aby można było kompilować programy używające GTK+).
- Strona WWW projektu: <http://www.gtk.org>

GTK - Okienko z przyciskiem

```
#include <gtk/gtk.h>

int main(int argc, char *argv[])
{
    gtk_init (&argc, &argv);

    GtkWidget *window=gtk_window_new(GTK_WINDOW_TOPLEVEL);

    gtk_window_set_title(GTK_WINDOW(window), "Zaczynamy z GTK+ 3.0");
    gtk_window_set_position(GTK_WINDOW(window), GTK_WIN_POS_CENTER);
    gtk_container_set_border_width(GTK_CONTAINER(window), 30);

    g_signal_connect(G_OBJECT(window), "destroy", G_CALLBACK(gtk_main_quit), NULL);

    GtkWidget *button=gtk_button_new_with_label("Witamy w świecie interfejsów GTK+");
    gtk_container_add(GTK_CONTAINER(window), button);
    gtk_widget_show(button);

    gtk_widget_show(window);

    gtk_main ();

    return 0;
}
```

GTK-test: definiowanie kontrolerek I

```
#include <string.h>
#include <gtk/gtk.h>

static struct przycisk {
    char *opis, *wyjscie;
    int posX, lenX, posY, lenY;
} tab[] = {
    {"NumLock", "", 0, 1, 0, 1}, {" / ", " / ", 1, 1, 0, 1}, {" * ", " * ", 2, 1, 0, 1}, {" - ", " - ", 3, 1, 0, 1},
    {" 7 ", " 7 ", 0, 1, 1, 1}, {" 8 ", " 8 ", 1, 1, 1, 1}, {" 9 ", " 9 ", 2, 1, 1, 1}, {" + ", " + ", 3, 1, 1, 2},
    {" 4 ", " 4 ", 0, 1, 2, 1}, {" 5 ", " 5 ", 1, 1, 2, 1}, {" 6 ", " 6 ", 2, 1, 2, 1},
    {" 1 ", " 1 ", 0, 1, 3, 1}, {" 2 ", " 2 ", 1, 1, 3, 1}, {" 3 ", " 3 ", 2, 1, 3, 1}, {" Enter ", " \n ", 3, 1, 3, 2},
    {" 0 ", " 0 ", 0, 2, 4, 1}, {" . ", " . ", 2, 1, 4, 1}
};

static GtkWidget *text;

static void test_nacisniecia( GtkWidget *widget, gpointer data)
{
    g_print("GTK-test - %s zostal nacisniety\n", (gchar *) data);
}

static void wyjscie_przyciskiem( GtkWidget *widget, gpointer data)
{
    gtk_main_quit();
}
```

GTK-test: definiowanie kontrolerek II

```
static void wypisz_wprowadzony_tekst( GtkWidget *widget, GtkWidget *text)
{
    const gchar *wejscie;
    wejscie = gtk_entry_get_text (GTK_ENTRY (text));
    g_print ("GTK-test wprowadzono: %s\n", wejscie);
}

static void dodaj_do_text(GtkWidget *widget, gpointer data) {
    gint tmp_pos = gtk_entry_get_text_length(GTK_ENTRY(text));

    if(strcmp((gchar*)data, "\n") == 0)
        gtk_entry_set_text(GTK_ENTRY(text), "");
    else
        gtk_editable_insert_text(GTK_EDITABLE(text), (gchar*)data, -1, &tmp_pos);
}

int main( int  argc, char *argv[] )
{
    gtk_init(&argc, &argv);

    GtkWidget *window = gtk_window_new(GTK_WINDOW_TOPLEVEL);

    gtk_window_set_title(GTK_WINDOW(window), "GTK - panel numeryczny");
    gtk_window_set_position(GTK_WINDOW(window), GTK_WIN_POS_CENTER);
    gtk_container_set_border_width(GTK_CONTAINER(window), 10);
```

GTK-test: definiowanie kontrolerek III

```
g_signal_connect(G_OBJECT(window), "destroy",G_CALLBACK(gtk_main_quit), NULL);

GtkWidget *box1 = gtk_box_new(GTK_ORIENTATION_VERTICAL, 0);

gtk_container_add(GTK_CONTAINER(window), box1);

GtkWidget *button = gtk_button_new_with_label("Kliknij tu");
g_signal_connect(G_OBJECT(button), "clicked",
                 G_CALLBACK(test_nacisniecia),(gpointer)"Przycisk");

gtk_box_pack_start(GTK_BOX(box1), button, TRUE, TRUE, 0);

text = gtk_entry_new();
gtk_entry_set_max_length(GTK_ENTRY(text), 50);
g_signal_connect(G_OBJECT(text), "activate",
                 G_CALLBACK(wypisz_wprowadzony_tekst),
                 (gpointer) text);
gtk_entry_set_text(GTK_ENTRY(text), "Wprowadz");
gint tmp_pos = gtk_entry_get_text_length(GTK_ENTRY(text));
gtk_editable_insert_text(GTK_EDITABLE(text), "  jakis tekst", -1,&tmp_pos);
gtk_editable_select_region(GTK_EDITABLE(text),
                           0, gtk_entry_get_text_length(GTK_ENTRY(text)));
gtk_box_pack_start(GTK_BOX(box1), text, TRUE, TRUE, 0);

GtkWidget *grid = gtk_grid_new();
gtk_grid_set_row_spacing(GTK_GRID(grid), 5);
gtk_grid_set_row_homogeneous(GTK_GRID(grid), TRUE);
```

GTK-test: definiowanie kontrolerek IV

```
gtk_grid_set_column_spacing(GTK_GRID(grid), 5);
gtk_grid_set_column_homogeneous(GTK_GRID(grid), TRUE);
gtk_box_pack_start(GTK_BOX(box1), grid, TRUE, TRUE, 0);

button=gtk_label_new(tab[0].opis);
gtk_grid_attach(GTK_GRID(grid), button, tab[0].posX, tab[0].posY, tab[0].lenX, tab[0].lenY);

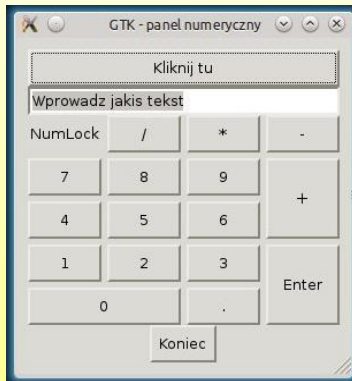
for(int i = 1; i < 17; i++) {
    button = gtk_button_new_with_label(tab[i].opis);
    g_signal_connect(G_OBJECT(button), "clicked", G_CALLBACK(dodaj_do_text),
        (gpointer) tab[i].wyjscie);
    gtk_grid_attach(GTK_GRID(grid), button, tab[i].posX, tab[i].posY, tab[i].lenX, tab[i].lenY);
}

GtkWidget *box2 = gtk_box_new(GTK_ORIENTATION_HORIZONTAL, 0);
button = gtk_button_new_with_label("Koniec");
g_signal_connect(G_OBJECT(button), "clicked", G_CALLBACK(wyjscie_przyciskiem), NULL);
gtk_box_pack_start(GTK_BOX(box2), button, TRUE, FALSE, 0);
gtk_box_pack_start(GTK_BOX(box1), box2, TRUE, TRUE, 0);

gtk_widget_show_all(window);
gtk_main();

return 0;
}
```

Uzyskany efekt



Układanie kontrolerek w pudełkach (parametry *expand* i *fill* w funkcjach pakujących)

```
GtkWidget *gtk_box_new( GtkOrientation orientation, gint spacing );  
  
void gtk_box_pack_start( GtkWidget *box, GtkWidget *child,  
                        gboolean expand, gboolean fill, gint padding );
```

