ISSN 2307-9851

НАУКОВО-МЕТОДИЧНИЙ ЖУРНАЛ

Виходить 8 разів на рік

Видається з лютого 1998 року

Засновники:
Інститут педагогіки НАПН України, Інститут інформаційних технологій і засобів навчання НАПН України, Редакція журналу

Журнал видається за сприяння Міністерства освіти і науки України
Свідоцтво про реєстрацію серія КВ №12217–1101ПР від 17.01.2007

Передплатний індекс 74248

Журнал включено до Переліку наукових фахових видань України у галузі педагогічних наук, Наказ МОН України від 29.09.2014 року №1081

Журнал внесений до наукометричної бази даних РИНЦ

Журнал індексується бібліометричним сервісом Google Scholar

Затверджено Вченою радою Інституту педагогіки НАПН України, протокол №14 від 8 жовтня 2018 р.

E-mail: csf22101@ukr.net

Тел. 044 481 37 39

Офіційний сайт журналу:
www.csf221.wordpress.com

КОМП'ЮТЕР у школі та сім'ї

№5 (149) ∈ 2018

ЗМІСТ

На першій і другій сторінках обкладинки:
Веб-конференція «Учені НАПН України – українським вчителям»
27-28 серпня 2018 року

УДК 372.8

ГРАФІКА З ВИКОРИСТАННЯМ МОВИ С++ І БІБЛІОТЕКИ WXWIDGETS

Рудик Олександр Борисович

кандидат фізико-математичних наук, доцент
Інституту післядипломної педагогічної освіти
Київського університету імені Бориса Грінченка,
rudykob@gmail.com,
ORCID ID 0000-0003-3676-0688

Анотація. У статті висвітлено питання роботи з графікою при використанні мови С++ і бібліотеки wxWidgets у формі коротких прокоментованих прикладів кодів.

Ключові слова:С++, CodeBlocks, wxWidgets, графіка, приклади кодів.

У публікації [1] проведено аналіз проблем, пов'язаних зі зміною парадигми вивчення інформатики у школі: істотним зростанням кількості навчальних годин на вивчення програмування. Там обгрунтовано вибір мови С++, середовища програмування CodeBlocks і бібліотеки wxWidgets. Дана робота є безпосереднім продовженням публікації [1].

Робота з графікою при використанні бібліотеки wxWidgets передбачає використання об'єкту визначення пристрою (device context) [2]. Такий об'єкт у wxWidgets називають wxDC. Його не призначено для безпосереднього використання, замість нього потрібно вибрати один з похідних класів:

- WxScreenDC використовують для малювання скрізь на екрані;
- WxWindowDC використовують для малювання у всьому вікні (лише для Windows). Це включає також оформлення вікон;
- WxClientDC використовують для малювання у клієнтській області вікна, тобто області вікна без його декорацій (заголовку й меж);
- WxPaintDC використовують для малювання у клієнтській області таким застереженням:
  - WxPaintDC потрібно використовувати лише з wxPaintEvent;
  - WxClientDC непотрібно використовувати з wxPaintEvent;
- WxMemoryDC використовують для нанесення графіки на растрові зображення;
- WxPostScriptDC використовують для запису на файли PostScript на будь-якій платформі;
- WxPrinterDC використовують для доступу до принтера (лише для Windows).

Охочі можуть продовжити знайомитися з описом класу wxDC на сторінці [3] офіційного сайту проекту wxWidgets. Альтернатива, реалізована у цій публікації, полягає в ознайомленні з конкретними прокоментованими прикладами кодів, які істотно коротші від перелічених в описі [4] на сайті проекту. Ці приклади створено у результаті опрацювання джерел [5–8]. Наявність ілюстрацій — зображень, зображень, побудованих у результаті виконання коду — дає можливість зрозуміти принципи роботи, читаючи лише друкований текст, без використання ПК. Досконале опанування передбачає щонайменше модифікацію поданих шести прикладів.

Подані на початку статті дані видаються зайвими. Але це лише на перший погляд — до спроби створити програму поточкового опрацювання побудованого зображення (див. останній приклад щодо симетрії відносно горизонталі).

Приклади побудови графічних примітивів

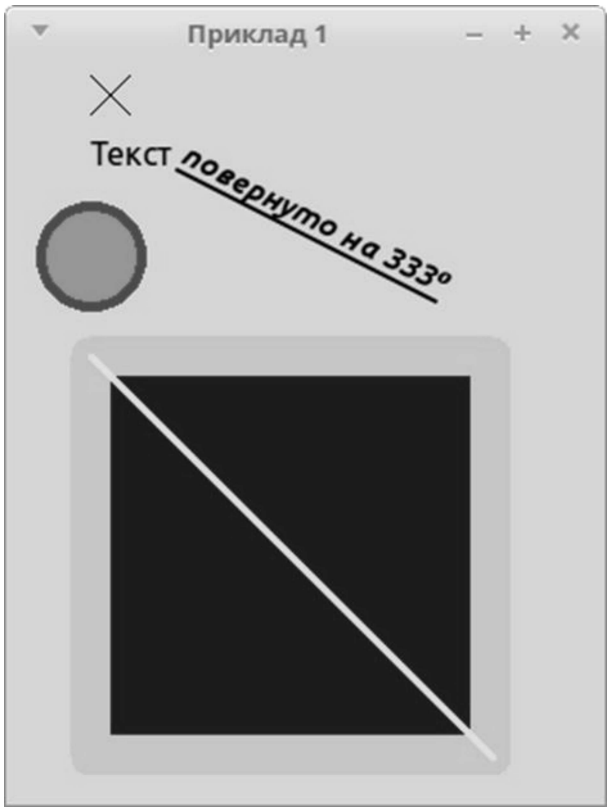

Рис. 1. Приклад побудови графічних примітивів

Зображення отримано у результаті виконання такого коду (подано усю програму, див. вказівки наприкінці коду).

```
#include "wx/wx.h"
#include "wx/sizer.h"
class BasicDrawPane : public wxPanel
```

```
{     public:     BasicDrawPane(wxFrame*
parent);
  void paintEvent(wxPaintEvent & evt);
  void paintNow();
  void render(wxDC& dc);
  /* Перелік корисних подій -
  подано лише у цьому коді:
  void mouseMoved(wxMouseEvent& event);
  void mouseDown(wxMouseEvent& event);
  void     mouseWheelMoved(wxMouseEvent&
event);
  void       mouseReleased(wxMouseEvent&
event);
  void rightClick(wxMouseEvent& event);
  void     mouseLeftWindow(wxMouseEvent&
event);
  void keyPressed(wxKeyEvent& event);
  void keyReleased(wxKeyEvent& event);
  */
  DECLARE_EVENT_TABLE()
};

class MyApp: public wxApp
{ bool OnInit();
  wxFrame *frame;
  BasicDrawPane * drawPane;
  public:
};

IMPLEMENT_APP(MyApp)

bool MyApp::OnInit()
{ wxBoxSizer* sizer =
    new wxBoxSizer(wxHORIZONTAL);
  frame = new wxFrame((wxFrame *)NULL, -
1,
    wxT("Приклад 1"), // заголовок вікна
    wxPoint(50,50),    // верхній лівий
кут
    wxSize(300,400)); // розміри вікна
  drawPane =
    new         BasicDrawPane((wxFrame*)
frame );
  sizer->Add(drawPane, 1, wxEXPAND);
  frame->SetSizer(sizer);
  frame->SetAutoLayout(true);
  frame->Show();
  return true;
}

BEGIN_EVENT_TABLE(BasicDrawPane,  wxPan-
el)
  /* Перелік корисних подій -
     подано лише у цьому коді:
  EVT_MOTION(BasicDrawPane::mouseMoved)
  EVT_LEFT_DOWN
(BasicDrawPane::mouseDown)
  EVT_LEFT_UP
(BasicDrawPane::mouseReleased)
  EVT_RIGHT_DOWN
(BasicDrawPane::rightClick)
  EVT_LEAVE_WINDOW(BasicDrawPane::
                    mouseLeftWindow)
  EVT_KEY_DOWN
(BasicDrawPane::keyPressed)
  EVT_KEY_UP(BasicDrawPane::keyReleased)
  EVT_MOUSEWHEEL(BasicDrawPane::
                 mouseWheelMoved)
  */
  // перехлення події малювання
  EVT_PAINT(BasicDrawPane::paintEvent)
END_EVENT_TABLE()

/* Перелік корисних подій -
   подано лише у цьому коді:
void BasicDrawPane::
     mouseMoved(wxMouseEvent& event) {}
void BasicDrawPane::
     mouseDown(wxMouseEvent& event) {}
void BasicDrawPane::
     mouseWheelMoved(wxMouseEvent&
event){}
void BasicDrawPane::
     mouseReleased(wxMouseEvent&  event)
{}
void BasicDrawPane::
     rightClick(wxMouseEvent& event) {}
void BasicDrawPane::
     mouseLeftWindow(wxMouseEvent&
event) {}
void BasicDrawPane::
     keyPressed(wxKeyEvent& event) {}
void BasicDrawPane::
     keyReleased(wxKeyEvent& event) {}
*/

BasicDrawPane::
BasicDrawPane(wxFrame* parent):
       wxPanel(parent) {}
/* Буде викликано, коли панель
потребуватиме перемалювання.
Можна запустити цей виклик
за допомогою Refresh()/Update().
*/

void BasicDrawPane::
     paintEvent(wxPaintEvent & evt)
{ wxPaintDC dc(this);
  render(dc);
}
/* Можна також використати clientDC для
   малювання на панелі у будь-який час.
   Такий спосіб не звільняє від потреби
   перехоплювати подій малювання, коли
   це можливо. Наприклад, менеджер вікон
   відкидає малюнком, коли вікно йде на
   задній план, і очікує на перемалюван-
ня,
   коли вікно повертається назад.
   У більшості випадків це не потрібно;
   достатньо перехоплювати події малю-
вання
   й викликати Refresh() для оновлення.
*/
```

```
void BasicDrawPane::paintNow()
{ wxClientDC dc(this);
  render(dc);
}

/* Далі - поточна візуалізація окремим
   методом. Окремим для того, щоб він
   спрацював незалежно від використано-
   го типу DC (наприклад, wxPaintDC чи
   wxClientDC).У наступних прикладах
   буде змінено лише розміри вікна і
   наступну частину коду. Для
   журнальної публікації замість усього
   коду буде подано лише цю частину.
*/

void BasicDrawPane::render(wxDC&  dc)
{ // чорний контур ширини 1 піксель
  dc.SetPen(wxPen(wxColor(0,0,0),1));
  for (int j = 0; j<20; j++)
  {// зафарбування точки (40+j,10+j)
      dc.DrawPoint(40+j,10+j);
      dc.DrawPoint(40+j,29-j);
  }
  wxFont font1(          // опис шрифту
    12,                  // розмір
    wxFONTFAMILY_DEFAULT,//родина
    wxFONTSTYLE_NORMAL,  //стиль  напи-
сання
    wxFONTWEIGHT_NORMAL,//наповнення
    false,              // підкреслити?
    "Ubuntu");          // гарнітура
  dc.SetFont(font1);// призначення шри-
фту

  // чорний колір шрифту
  dc.SetTextForeground(wxColor(0,0,0));

  // прозоре тло для виведення тексту
  dc.SetBackgroundMode(wxTRANSPARENT);

  // Виведення тексту з точки (40, 40)
  dc.DrawText(wxT("Текст"),40,40);
    wxFont font2 (12,   // розмір
    wxFONTFAMILY_DEFAULT,
    wxFONTSTYLE_ITALIC,//стиль написан-
ня
    wxFONTWEIGHT_BOLD, //наповнення
    true,
    "Ubuntu");      // гарнітура
  dc.SetFont(font2);//призначення
  шрифту
  // Виведення повернутого тексту
  dc.DrawRotatedText(
    wxT("повернуто на
333°"),90,40,333);
  // заповнення зеленим кольором
  dc.SetBrush(*wxGREEN_BRUSH);

  // червоний контур ширини 5 пікселів
  dc.SetPen(wxPen(wxColor(255,0,0),5));
  // намалювати коло радіусом 25
  // і центром (40, 100)
  dc.DrawCircle(wxPoint(40,100), 25);

  // заповнення блакитним кольором
  dc.SetBrush(*wxBLUE_BRUSH);

  // рожевий контур ширини 20 пікселів
  dc.SetPen(wxPen(wxColor
(255,175,175),20));

  // намалювати прямокутник з верхньою
лівою
  //  вершиною  (40,150)  і  розмірами
200x200
  dc.DrawRectangle( 40, 150, 200, 200);

  // жовтий контур ширини 3 пікселі
  dc.SetPen(wxPen(              wxColor
(255,255,0),3));

  // сполучити відрізком прямої точки
  // (40,150) і (240,350)
  dc.DrawLine(40,150,240,350);
}
```

Пояснимо, які значення можна надати властивостям об'єктів, зображеним на попередньому малюнку вище.

Родини шрифтів

wxFONTFAMILY_DEFAULT
wxFONTFAMILY_DECORATIVE
wxFONTFAMILY_ROMAN
wxFONTFAMILY_SCRIPT
wxFONTFAMILY_SWISS
wxFONTFAMILY_MODERN
wxFONTFAMILY_TELETYPE
wxFONTFAMILY_MAX

Стилі написання

wxFONTSTYLE_NORMAL
wxFONTSTYLE_ITALIC
wxFONTSTYLE_SLANT
wxFONTSTYLE_MAX

Види наповнення

wxFONTWEIGHT_NORMAL
wxFONTWEIGHT_LIGHT
wxFONTWEIGHT_BOLD
wxFONTWEIGHT_MAX

Pen (перо) *використовують для зображення ліній (прямих і кривих), контурів многокутників і еліпсів.*

Налаштування Pen здійснюють такою вказівкою:

```
wxPen(const wxColour& colour,
  int width = 1, int style = wxSOLID)
```

Вона має такі три агрументи (перелічено у порядку запису, вказано значення як усталено): колір, ширина і стиль. Вище в описі вказівки wxPen вказано значення як усталено. У коді вище стиль не задано.

Стилі **Pen**:

wxPENSTYLE_SOLID — суцільний;
wxPENSTYLE_DOT — крапки;
wxPENSTYLE_LONG_DASH — довгі риски;
wxPENSTYLE_SHORT_DASH — короткі риски;
wxPENSTYLE_DOT_DASH — риски й крапки;
wxPENSTYLE_TRANSPARENT — прозорий.

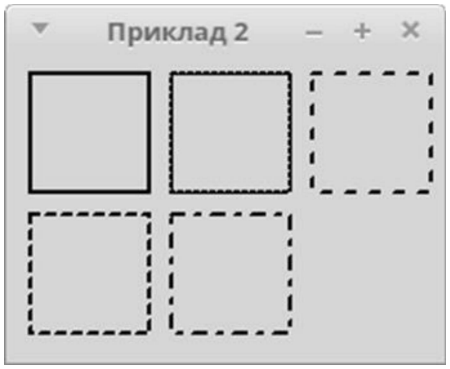

Рис. 2. Приклад використання стилів Pen

Зображення на рис.2.. отримано у результаті виконання такого коду (подано лише кінець програми).

```
…
wxSize(220,180)); // розміри вікна
…
void BasicDrawPane::render(wxDC&  dc)
{ dc.SetBrush(*wxTRANSPARENT_BRUSH);

  dc.SetPen(wxPen(wxColor(0,0,0), 2,
                  wxPENSTYLE_SOLID));
  dc.DrawRectangle(10, 10, 60, 60);

  dc.SetPen(wxPen(wxColor(0,0,0), 2,
                   wxPENSTYLE_DOT));
  dc.DrawRectangle(80, 10, 60, 60);

  dc.SetPen(wxPen(wxColor(0,0,0), 2,
              wxPENSTYLE_LONG_DASH));
  dc.DrawRectangle(150, 10, 60, 60);

  dc.SetPen(wxPen(wxColor(0,0,0), 2,
             wxPENSTYLE_SHORT_DASH));
  dc.DrawRectangle(10, 80, 60, 60);

  dc.SetPen(wxPen(wxColor(0,0,0), 2,
               wxPENSTYLE_DOT_DASH));
  dc.DrawRectangle(80, 80, 60, 60);

  dc.SetPen(wxPen(wxColor(0,0,0), 2,
            wxPENSTYLE_TRANSPARENT));
  dc.DrawRectangle(150, 80, 60, 60);
}
```

У цьому коді в якому кольору заповнення надано прозорості (*wxTRANSPARENT_BRUSH). Це зроблено з метою не відволікати увагу від стилю контура. Але колір заповнення може бути іншим.

Вбудовані (стандартні) кольори заповнення

wxNullBrush
wxBLACK_BRUSH
wxBLUE_BRUSH
wxCYAN_BRUSH
wxGREEN_BRUSH
wxYELLOW_BRUSH
wxGREY_BRUSH
wxLIGHT_GREY_BRUSH
wxMEDIUM_GREY_BRUSH
wxRED_BRUSH
wxTRANSPARENT_BRUSH
wxWHITE_BRUSH

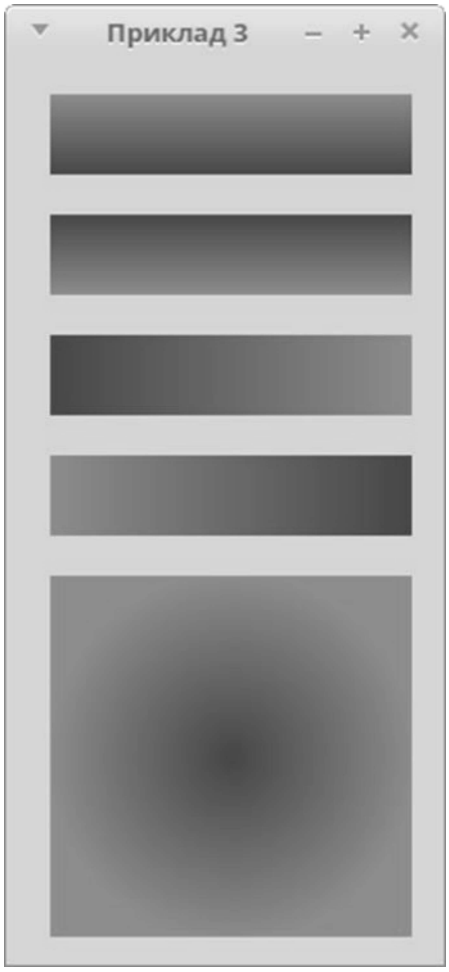

Рис. 3. Приклад використання градієнтів

Зображення на рис. 3. отримано у результаті виконання такого коду (подано лише кінець програми).

```
…
wxSize(220,480)); // розміри вікна
…
void BasicDrawPane::render(wxDC&  dc)
{ wxColour c1, c2;
  c1.Set(wxT("#ee0000"));//червоний
колір
  c2.Set(wxT("#00ee00"));//     зелений
колір
                         //         лінійні
градієнти
  dc.GradientFillLinear(
  wxRect(20,20,180,40),//об'єкт
          c1,       // початковий колір
          c2,       //   кінцевий колір
          wxNORTH); //     напрям вгору
  dc.GradientFillLinear(
  wxRect(20, 80, 180,
40),c1,c2,wxSOUTH);
  dc.GradientFillLinear(
  wxRect(20, 140, 180,
40),c1,c2,wxEAST);
  dc.GradientFillLinear(
```

```
  wxRect(20, 200, 180,
40),c1,c2,wxWEST);

    // круговий градієнт
  dc.GradientFillConcentric(
    wxRect(20,260,180,180),    //
об'єкт
    c1, c2,       // кольори заповнення
   wxPoint(90, 90));// початкова точка
}
```

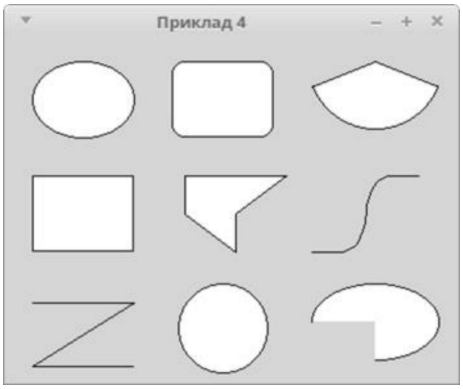

Рис. 4. Приклад побудови фігур

Зображення на рис. 4. отримано у результаті виконання такого коду (подано лише кінець програми).

```
…
wxSize(360,300)); // розміри вікна
…
void BasicDrawPane::render(wxDC&  dc)
{ wxPoint lines[] =  // вершини ламаної
  { wxPoint( 20,210),wxPoint(100,210),
    wxPoint( 20,260),wxPoint(100,260)};

  wxPoint  polygon[]  =//  вершини  5-
кутника
  { wxPoint(140,140),wxPoint(180,170),
    wxPoint(180,140),wxPoint(220,110),
    wxPoint(140,110)};

  // опорні точки сплайну
  wxPoint splines[] =
  { wxPoint(240,170),wxPoint(280,170),
    wxPoint(285,110),wxPoint(325,110)};

  dc.DrawEllipse(20,20,  //  координати
ВЛК
                 80,60);//      розміри
еліпса
  dc.DrawRoundedRectangle(
               130,  20,  //  координати
ВЛК
                80, 60, // розміри
         10);//   радіус   заокруглення
кутів
  dc.DrawArc( // дуга кола
      240,  40,  //  координати  початку
дуги
      340,  40,  //  координати   кінця
дуги
      290,  20);//  центр  відповідного
кола
  // рух дугою - у напрямку, протилеж-
ному
  // до руху годинникової стрілки

  dc.DrawRectangle(20,110,80,60);
  dc.DrawPolygon(5,polygon);    //    5-
кутник
  dc.DrawSpline(4,splines);  // сплайн

  dc.DrawLines(4, lines);     // ламана
  dc.DrawCircle(170, 230, 35);// коло
  dc.DrawEllipticArc(          //  дуга
еліпса
    240,195,//координати  ВЛК  прямокут-
ника
    100,60,//розміри  описаного  прямо-
кутника
270,180);//  кутові  координати  кінців
дуги
}
```

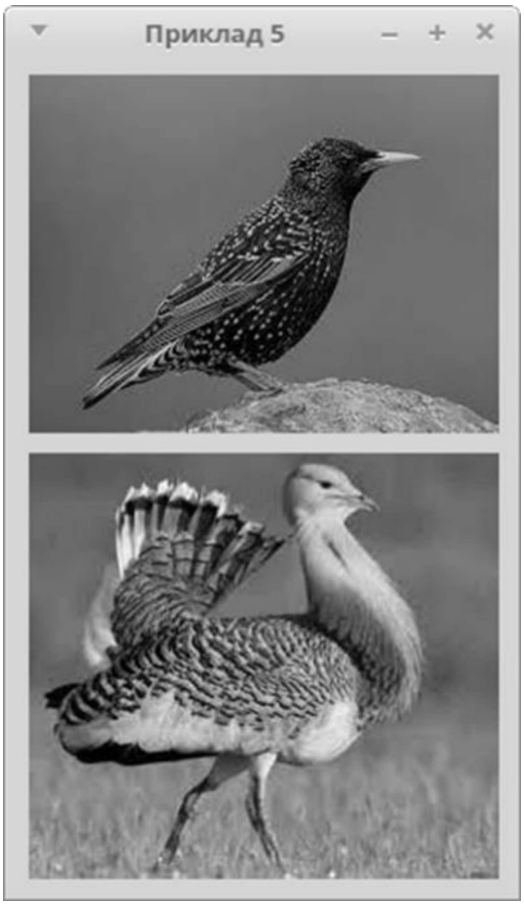

Рис. 5. Приклад відображення малюнків

Зображення на рис. 5. отримано у результаті виконання такого коду (подано лише кінець програми).

```
…
wxSize(260,450));// розміри вікна
…
void BasicDrawPane::render(wxDC&  dc)
{ dc.DrawBitmap(wxBitmap(
```

```
    wxT("10.jpeg"),     // шпак
    wxBITMAP_TYPE_JPEG),// тип
              10, 10, //координати
               true );
  dc.DrawBitmap( wxBitmap(
    wxT("02.jpeg"),     // дрофа
    wxBITMAP_TYPE_JPEG),10,200,true);
}
```

Рис. 6. Приклад поточкового опрацювання побудованого малюнку

Зображення на рис. 6 отримано у результаті виконання такого коду (подано лише кінець програми).

```
…
wxSize(260,420));// розміри вікна
…
void BasicDrawPane::render(wxDC&  dc)
{ // задати білий колір тла
    dc.SetBackground(*wxWHITE_BRUSH);

  // очистити: отримати тло білого кольору,
  // а не сірого - передбаченого
  dc.Clear();   // налаштуваннями ОС

  dc.DrawBitmap(wxBitmap(// зображення
    wxT("10.jpeg"),     // фото шпака
    wxBITMAP_TYPE_JPEG),// тип
            10, 10, // координати
             true);

  // виділення пам'яті для зображення
  // і вирізання його з копії екрану
  wxBitmap bitmap(236, 180);
  wxMemoryDC memDC;
  memDC.SelectObject(bitmap);
  memDC.Blit(0, 0, 236, 180,& dc, 10,
10);
  memDC.SelectObject(wxNullBitmap);

  //конвертація для можливості
опрацювання
  wxImage   img = bitmap.ConvertToImage
();

  //поточкове опрацювання й виведення
  for (int j=0; j<236; j++)
  for (int k=0; k<180; k++)
  { dc.SetPen(wxPen(wxColor(
       img.GetRed(j,k),
       img.GetGreen(j,k),
       img.GetBlue(j,k)),1));
       dc.DrawPoint(10+j,380-k);
  }
}
```

Список використаних джерел

1.Рудик О. Налаштування взаємодії wxWidgets і CodeBlocks для забезпечення осучаснення навчання інформатики. — Комп'ютер у школі та сім'ї. — № 3 (147), 2018. — С. 16–20.

2.Визначення пристрою у wxWidgets. — Режим доступу: http://zetcode.com/gui/wxwidgets/gdi/

3.Довідник класу wxDC. — Режим доступу: http://docs.wxwidgets.org/3.1/classwx_d_c.html#a13978b2624116987a59ff729c4f81a96

4.Огляд прикладів. — Режим доступу: http://docs.wxwidgets.org/3.1/page_samples.html

5.Малювання на панелі з використанням DC. — Режим доступу: https://wiki.wxwidgets.org/Drawing_on_a_panel_with_a_DC

6.Панель зображення. — Режим доступу: https://wiki.wxwidgets.org/An_image_panel

7.Малювання на панелі в C++ з wxWidgets. — Режим доступу: http://www.informit.com/articles/article.aspx?p=405047

8.Руководство по wxWidgets. — Режим доступу: http://www.sl-alex.com.ua/ru/page/wxwidgets_tutorial_00_introduction

References. Translation and transliteration

1.Rudick O. Interaction settings wxWidgets and CodeBlocks to ensure the modernization of computer science. - Computer at school and family. - No. 3 (147), 2018. - P. 16-20

2.Define the device in wxWidgets. - Mode access: http://zetcode.com/gui/wxwidgets/gdi/

3.The wxDC class guide. - Access mode: http://docs.wxwidgets.org/3.1/classwx_d_c.html#a13978b2624116987a59ff729c4f81a96

4.Examples Overview - Access mode: http://docs.wxwidgets.org/3.1/page_samples.html

5.Draw on a panel using DC. - Access mode: https://wiki.wxwidgets.org/Drawing_on_a_panel_with_a_DC

6.Image panel. - Access mode: https://wiki.wxwidgets.org/An_image_panel

7.Draw a panel in C ++ with wxWidgets. - Access mode: http://www.informit.com/articles/article.aspx?p=405047

8.WxWidgets Guide. - Access mode: http://www.sl-a l e x . c o m . u a / e n / p a g e / wxwidgets_tutorial_00_introduction

ГРАФИКА С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА С++ И БИБЛИОТЕКИ WXWIDGETS

Рудик Александр Борисович

кандидат физико-математических наук, доцент Института последипломного педагогического образования Киевского университета имени Бориса Гринченко,

rudykob@gmail.com

Аннотация. В статье освещены вопросы работы с графикой при использовании языка С ++ и библиотеки wxWidgets в форме коротких прокомментированных примеров кодов.

Ключевые слова*: С ++, CodeBlocks, wxWidgets, графика, примеры кодов. .*

GRAPHICS USING C ++ LANGUAGE AND WXWIDGETS LIBRARY

Rudyk Alexander Borisovich*,*

candidate of Sciences (Ph.D.), Associate Professor of the Institute of Postgraduate Pedagogical Education of the Boris Grinchenko University of Kyiv,

rudykob@gmail.com ",

Annotation. The article is devoted to working with graphics using the C ++ language and wxWidgets library in the form of short commented code samles.

Keywords: C ++, CodeBlocks, wxWidgets, graphics, code samles.

Продовження серії статей у наступних номерах

* * *

УДК 37.02

ОСВІТНЯ РОБОТОТЕХНІКА В ПОЗАУРОЧНІЙ НАВЧАЛЬНІЙ ДІЯЛЬНОСТІ

Кіт Ігор Володимирович –

учитель інформатики та технологій, Одеська спеціалізована школа I-III ступенів «Освітні ресурси та технологічний тренінг» №94 імені Володимира (Зеєва) Жаботинського з поглибленим вивченням івриту та інформатики Одеської міської ради, спеціаліст вищої категорії, igorkot3@gmail.com

Кіт Ольга Григорівна –

учитель інформатики та технологій, Одеська спеціалізована школа I-III ступенів «Освітні ресурси та технологічний тренінг» №94 імені Володимира (Зеєва) Жаботинського з поглибленим вивченням івриту та інформатики Одеської міської ради, спеціаліст вищої категорії , oollyyk@gmail.com

Анотація. У статті описано досвід застосування інструментів освітньої робототехніки як ефективного засобу розвитку творчого потенціалу учнів. Показано, що використовуючи елементи освітньої робототехніки, ми можемо зробити сучасну школу конкурентоспроможною, а навчання – по-справжньому ефективним і продуктивним. Зазначено, що використання елементів освітньої робототехніки підвищує мотивацію учнів до оволодіння новими знаннями, сприяє розвитку творчих здібностей, просторової уяви, конструктивного мислення і колективної взаємодії. Розглянуто можливість упровадження робототехніки у позаурочну навчальну діяльність шляхом організації роботи гуртка робототехніки. У основу побудови такого гуртка покладено ідеї системно-діяльнісного і компетентнісного підходів як сучасної парадигми освітнього процесу, результати психодидактичного розгляду особливостей суб'єктів навчання. Робота гуртка робототехніки розглядається в контексті трикомпонентної за цілями й видами діяльності системи: робот як об'єкт вивчення; робот як інструмент пізнання; робот як засіб навчання, розвитку та виховання. Особливу увагу приділено особливостям побудови гуртка: по-