Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «МОП ЭВМ»

РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ

по дисциплине «Операционные системы и оболочки»

## 

Студент группы 6ИСб-1 И. Нозимзода

Преподаватель В.Я Столяров

2019

**Содержание**

[Задание 3](#_Toc533107147)

[Ход выполнения работы 4](#_Toc533107148)

[Список использованных источников 7](#_Toc533107149)

Задание

Следует выполнить разработку программы на языке C++ по технологии Win API, которая бы работала так же, как и прилагаемый образец (обзор изображений в текущей папке и возможность установки их как обоев рабочего стола).

Ход выполнения работы

Образец программы из задания показан на рисунке 1.

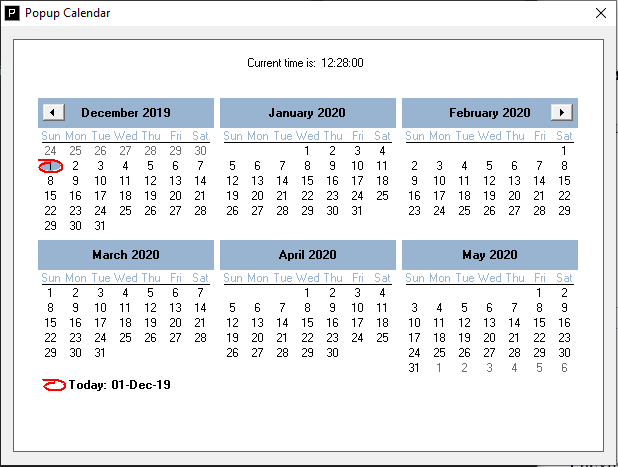


Рисунок 1 – Календарь

По примеру программы индивидуального задания была реализована программа «Календарь», основными функциями которой является переход между месяцами, переход к текущему дню и отображение текущего времени. Так же требовалось реализовать иконку в системном меню с выпадающим списком.

Текст программы приведен в листинге 1.

Листинг 1 – Текст файла Rgz.cpp

|  |
| --- |
| #include <windows.h>  #include "resource.h"  LRESULT CALLBACK PviewDlgProc(HWND hWnd, UINT wMsg, WPARAM wParam, LPARAM lParam);  int WINAPI WinMain  (  \_In\_ HINSTANCE hInstance, // дескриптор экземпляра приложения. Этот дескриптор содержит адрес начала кода программы в ее адресном пространстве. Дескриптор hInstance чаще всего требуется функциям, работающим с ресурсами программы  \_In\_opt\_ HINSTANCE, // дескриптор предыдущего экземпляра приложения. Этот дескриптор остался от старых версий Windows - скорее всего, вам он никогда не пригодится. Всегда равен 0  \_In\_ LPSTR, // указатель на начало командной строки, введенной при запуске программы  \_In\_ int // это значение содержит желаемый вид окна (например, свернутый или развернутый)  )  {  DialogBox(hInstance, MAKEINTRESOURCE(IDD\_DIALOG1), 0, PviewDlgProc);  return 0;  }  // Процедура обработки сообщений диалогового окна  LRESULT CALLBACK PviewDlgProc  (  HWND hWnd, // уникальный идентификатор окна от которого пришло сообщение  UINT wMsg, // сообщение, получаемое от ОС  WPARAM wParam, // дополнительная информация сообщения  LPARAM lParam // дополнительная информация сообщения  )  {  // Добавление двух категорий в меню иконки системного трея  if (lParam == WM\_RBUTTONDOWN)  {  SetForegroundWindow(hWnd);  POINT mouse;  GetCursorPos(&mouse);  HMENU hMenu = CreatePopupMenu();  AppendMenu(hMenu, MF\_ENABLED, SHOW, "Show Calendar");  AppendMenu(hMenu, MF\_SEPARATOR, 1000, "");  AppendMenu(hMenu, MF\_ENABLED, EXIT, "Exit Program");  TrackPopupMenu(hMenu, TPM\_RIGHTBUTTON, mouse.x, mouse.y, 0, hWnd, NULL);  }  switch (wMsg)  {  case WM\_PAINT:  {  // Отрисовка белой центровой области в окно, которую не получилось сделать через свойства  RECT rect;  rect.bottom = 422;  rect.top = 12;  rect.left = 13;  rect.right = 604;  PAINTSTRUCT ps;  BeginPaint(hWnd, &ps);  FillRect(ps.hdc, &rect, (HBRUSH)GetStockObject(COLOR\_WINDOW + 1));  EndPaint(hWnd, &ps);  }  break;  // Сообщение при инициализации диалогового окна  case WM\_INITDIALOG:  {  // Установка иконки у окна  HICON hIcon = LoadIcon(GetModuleHandle(NULL), (LPCSTR)IDI\_ICON2);  SendMessage(hWnd, WM\_SETICON, 1, (LPARAM)hIcon);  // Установка иконки в системном трее  NOTIFYICONDATA nid;  memset(&nid, 0, sizeof(NOTIFYICONDATA));  nid.cbSize = sizeof(NOTIFYICONDATA);  nid.hWnd = hWnd;  nid.uID = 1;  nid.uFlags = NIF\_ICON | NIF\_MESSAGE | NIF\_TIP;  nid.uCallbackMessage = WM\_USER + 200;  nid.hIcon = LoadIcon(GetModuleHandle(NULL), (LPCSTR)IDI\_ICON2);  lstrcpy(nid.szTip, "Popup Calendar");  Shell\_NotifyIcon(NIM\_ADD, &nid);  // Установка времени  char temp[12] = "";  GetTimeFormat(0, 0, 0, "hh':'mm':'ss", temp, 8);  SetDlgItemText(hWnd, IDC\_EDIT1, temp);  // Установка таймера для обновления поля времени  SetTimer(hWnd, 1, 1000, 0);  }  break;  case WM\_TIMER:  {  char temp[12] = "";  GetTimeFormat(0, 0, 0, "hh':'mm':'ss", temp, 8);  SetDlgItemText(hWnd, IDC\_EDIT1, temp);  }  break;  // Сообщение от элементов диалогового окна  case WM\_COMMAND:  {  switch (LOWORD(wParam))  {  case SHOW:  ShowWindow(hWnd, SW\_SHOW);  break;  case EXIT:  PostQuitMessage(0);  default:  return FALSE;  }  break;  }  // Удаляем иконку из трея, иначе сборщик мусора удалит её с задержкой  case WM\_DESTROY:  {  NOTIFYICONDATA nid;  memset(&nid, 0, sizeof(NOTIFYICONDATA));  nid.cbSize = sizeof(NOTIFYICONDATA);  nid.hWnd = hWnd;  nid.uID = 1;  Shell\_NotifyIcon(NIM\_DELETE, &nid);  }  break;  // Установка белого фона для остальных элементов окна  case WM\_CTLCOLORSTATIC:  {  SetBkColor((HDC)wParam, RGB(0xFF, 0xFF, 0xFF));  return (LRESULT)CreateSolidBrush(RGB(0xFF, 0xFF, 0xFF));  }  break;  // Сообщение при закрытии диалогового окна  case WM\_CLOSE:  ShowWindow(hWnd, SW\_HIDE);  break;  default:  return FALSE;  }  return TRUE;  } |

Результат работы программы показан на рисунке 1.

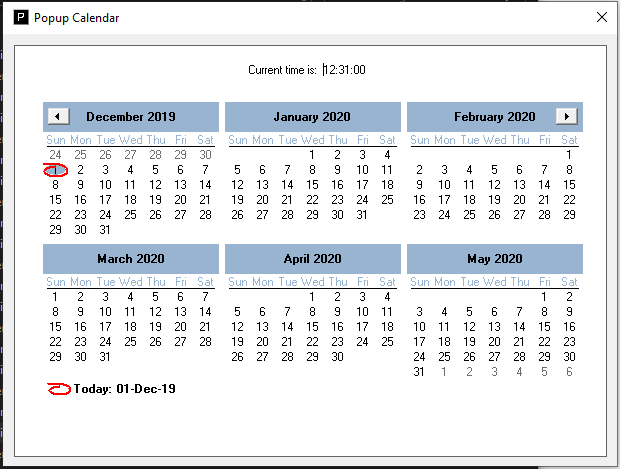


Рисунок 2 – Календарь

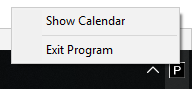


Рисунок 3 – Иконка в системном меню

Список использованных источников

1. Саймон, Р. Windows 2000 API Энциклопедия программиста / Р. Саймон. - М: DiaSoft, 2002