Win API

1. 다음과 같이 문자열을 출력하시오

화면 크기를 변경하여도 자동으로 정렬하여야 함

[윈도우 관리 메시지, 문자열 출력 파트]

|  |  |  |
| --- | --- | --- |
| 좌측상단 | 상단 | 우측상단 |
| 좌측 | 중앙 | 우측 |
| 좌측 하단 | 하단 | 우측하단 |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam) {  HDC hDC;  PAINTSTRUCT ps;  static RECT rt;    switch (iMsg) {  case WM\_DESTROY:  PostQuitMessage(0);  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  SetTextAlign(hDC, TA\_LEFT);  TextOut(hDC, rt.left, rt.top, TEXT("좌측상단"), 8);  TextOut(hDC, rt.left, rt.bottom / 2, TEXT("좌측"), 4);  SetTextAlign(hDC, TA\_LEFT | TA\_BOTTOM);  TextOut(hDC, rt.left, rt.bottom, TEXT("좌측하단"), 8);  SetTextAlign(hDC, TA\_CENTER);  TextOut(hDC, rt.right / 2, rt.top, TEXT("상단"), 4);  TextOut(hDC, rt.right / 2, rt.bottom /2 , TEXT("중앙"), 4);  SetTextAlign(hDC, TA\_CENTER | TA\_BOTTOM);  TextOut(hDC, rt.right / 2, rt.bottom, TEXT("하단"), 4);  SetTextAlign(hDC, TA\_RIGHT);  TextOut(hDC, rt.right, rt.top, TEXT("우측상단"), 8);  TextOut(hDC, rt.right, rt.bottom / 2, TEXT("우측"), 4);  SetTextAlign(hDC, TA\_RIGHT | TA\_BOTTOM);  TextOut(hDC, rt.right, rt.bottom, TEXT("우측하단"), 8);  break;  case WM\_SIZE:  GetClientRect(hWnd, &rt);  InvalidateRect(hWnd, NULL, TRUE);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } | | |

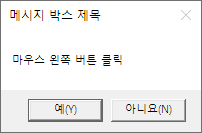
1. 다음과 같이 도형을 출력하시오

[도형 출력 파트]

|  |
| --- |
|  |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam)  {  HDC hDC;  PAINTSTRUCT ps;  switch (iMsg)  {  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  Rectangle(hDC, 100, 100, 1100, 500);  Rectangle(hDC, 150, 150, 300, 300);  Rectangle(hDC, 400, 150, 550, 300);  Rectangle(hDC, 650, 150, 800, 300);  Rectangle(hDC, 900, 150, 1050, 300);  Ellipse(hDC, 300, 400, 500, 600);  Ellipse(hDC, 700, 400, 900, 600);    MoveToEx(hDC, 250, 450, nullptr);  LineTo(hDC, 300, 360);  LineTo(hDC, 500, 360);  LineTo(hDC, 550, 450);  MoveToEx(hDC, 650, 450, nullptr);  LineTo(hDC, 700, 360);  LineTo(hDC, 900, 360);  LineTo(hDC, 950, 450);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |

1. 프로그램 우측 상단 X키를 눌렀을 때 다음 메시지 박스를 띄우고 ‘예’를 눌렀을 때만 종료시키시오

[메시지 박스 파트]

http://www.soen.kr/lecture/win32api/lec3/Image251.gif

프로그램을 종료합니까?

프로그램 종료

|  |
| --- |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam) {  HDC hDC;  PAINTSTRUCT ps;  switch (iMsg) {  case WM\_CLOSE:  if (MessageBox(hWnd, TEXT("프로그램을 종료합니까?"), TEXT("프로그램 종료"), MB\_YESNO) == IDNO)  return 0;  else {  break;  }  case WM\_DESTROY:  PostQuitMessage(0);  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  TextOut(hDC, 100, 100, TEXT("우왕"), 4);  EndPaint(hWnd, &ps);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |

1. 방향키를 누르면 문자 A가 상하좌우로 움직이고 (200,200) 좌표에 위치한 문자 B에 A가 겹쳐졌을 때 프로그램이 종료되도록 하시오

[문자열 출력, 키보드 입력 파트]

|  |
| --- |
| A  B |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam) {  HDC hDC;  PAINTSTRUCT ps;  static INT x = 0;  static INT y = 0;  switch (iMsg) {  case WM\_DESTROY:  PostQuitMessage(0);  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  SetTextAlign(hDC, TA\_LEFT | TA\_TOP);  TextOut(hDC, 200, 200, TEXT("B"), 1);  TextOut(hDC, x, y, TEXT("A"), 1);  if (x == 200 && y == 200) SendMessage(hWnd, WM\_DESTROY, 0, 0);  EndPaint(hWnd, &ps);  break;  case WM\_KEYDOWN:  switch (wParam) {  case VK\_LEFT:  x -= 10;  if (x < 0) x = 0;  break;  case VK\_RIGHT:  x += 10;  break;  case VK\_UP:  y -= 10;  if (y < 0) y = 0;  break;  case VK\_DOWN:  y += 10;  break;  }  InvalidateRect(hWnd, NULL, TRUE);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |

1. 화면을 더블클릭하면 문자 A가 더블클릭한 좌표로 이동하도록 하시오

이동 방법(순간이동, 조금씩 이동)은 자유

[문자열 출력, 마우스 입력 파트]

|  |
| --- |
| A  더블클릭!  A |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam) {  HDC hDC;  PAINTSTRUCT ps;  INT x = 0, y = 0;  switch (iMsg) {  case WM\_LBUTTONDBLCLK:  x = LOWORD(lParam);  y = HIWORD(lParam);  InvalidateRect(hWnd, NULL, TRUE);  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  TextOut(hDC, x, y, TEXT("A"), 1);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  } |

1. 타이머 프로시저를 이용하여 매 1초 간격으로 알아서 랜덤한 방향으로 움직이는 문자 A를 화면에 출력하시오

[문자열 출력, 타이머 콜백 함수 파트]

|  |
| --- |
| A A    A |
| 답안  LRESULT CALLBACK WndProc(HWND hWnd, UINT iMsg, WPARAM wParam, LPARAM lParam) {  HDC hDC;  PAINTSTRUCT ps;  time\_t mytime;  static HANDLE hTimer;  INT move = 0;  static INT x = 100, y = 100;  switch (iMsg) {  case WM\_CREATE:  SetTimer(hWnd, 1, 1000, NULL);  SendMessage(hWnd, WM\_TIMER, 1, 0);  break;  case WM\_TIMER:  move = rand() % 4;  switch (move) {  case 0:  x -= 10;  break;  case 1:  x += 10;  break;  case 2:  y -= 10;  break;  case 3:  y += 10;  break;  }  InvalidateRect(hWnd, NULL, TRUE);  break;  case WM\_PAINT:  hDC = BeginPaint(hWnd, &ps);  TextOut(hDC, x, y, TEXT("A"), 1);  EndPaint(hWnd, &ps);  break;  case WM\_DESTROY:  KillTimer(hWnd, 1);  PostQuitMessage(0);  break;  }  return DefWindowProc(hWnd, iMsg, wParam, lParam);  }  void CALLBACK TimerProc(HWND hWnd, UINT iMsg, UINT idEvent, DWORD dwTime) {  HDC hDC;  INT i;  }  TimerProc를 어떻게 쓸지 모르겠습니다. |

1. (심화)창 상단에서 랜덤으로 떨어지는 문자 B와 플레이어가 좌우 방향키로 움직일 수 있는 창 하단에 위치한 문자 A를 만들고 프로그램을 시작하고 흐른 시간을 문자열로 표시해준다. 문자 A가 B에 겹쳐지면 흐른 시간을 메시지 박스로 알려주고 프로그램을 종료시킨다.

[키보드 입력, 원도우 관리 메시지,타이머,문자열 출력 응용]

|  |
| --- |
| B B B 시간 : 48초  B B  B B B  B  B B B  B B  B B  B B B  B B B  B  B B B  A |
| 답안 |