[逆天通用水印支持Winform，WPF，Web，WP，Win10。支持位置选择（9个位置 ==》[X]）](http://www.cnblogs.com/dunitian/p/4939369.html)

好几天没上QQ了，今天上了个QQ，，额....额...貌似消息还挺多，没及时回复的还请见谅~~刚好昨天无聊把水印这快封装出来了，支持图片水印，文字水印，索引图水印和非索引图水印，支持位置选择（9个位置 ==》[X]）。好吧不高兴屁话了，软件是WPF的（只满足我的需求，如果有需要可以出第个二版本），水印核心代码封装了个Helper类（通用：支持Winform，WPF，Web，WP，Win10）软件共享一下==><http://pan.baidu.com/s/1qWDwv4C> 源码看下面  


水印类

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113 | /// <summary>  /// 水印类（重要参数：原图类型，水印类型，水印位置，图片水印路径，文字水印内容）  /// </summary>  public class WaterMark  {      #region 基础属性设置      private string \_photoType;      /// <summary>      /// 原图类型      /// </summary>      public string PhotoType      {          get { return \_photoType; }          set { \_photoType = value; }      }        private WaterMarkTypeEnum \_waterMarkType = WaterMarkTypeEnum.NoneMark;      /// <summary>      /// 水印类型      /// </summary>      public WaterMarkTypeEnum WaterMarkType      {          get { return \_waterMarkType; }          set { \_waterMarkType = value; }      }      #endregion        #region 水印效果设置      private WaterMarkLocationEnum \_waterMarkLocation = WaterMarkLocationEnum.BottomRight;      /// <summary>      /// 水印位置      /// </summary>      public WaterMarkLocationEnum WaterMarkLocation      {          get { return \_waterMarkLocation; }          set { \_waterMarkLocation = value; }      }        private float \_transparency = 0.7f;      /// <summary>      /// 水印透明度      /// </summary>      public float Transparency      {          get { return \_transparency; }          set { \_transparency = value; }      }      #endregion        #region 图片水印设置      private string \_imgPath;      /// <summary>      /// 图片水印路径      /// </summary>      public string ImgPath      {          get { return \_imgPath; }          set { \_imgPath = value; }      }      #endregion        #region 文字水印设置      private string \_text = "dunitian";      /// <summary>      /// 文字水印内容      /// </summary>      public string Text      {          get { return \_text; }          set { \_text = value; }      }        private string \_fontFamily = "微软雅黑";      /// <summary>      /// 文字字体      /// </summary>      public string FontFamily      {          get { return \_fontFamily; }          set { \_fontFamily = value; }      }        private Brush \_brushesColor = Brushes.Black;      /// <summary>      /// 文字颜色      /// </summary>      public Brush BrushesColor      {          get { return \_brushesColor; }          set { \_brushesColor = value; }      }        private  FontStyle \_fontStyle = FontStyle.Regular;      /// <summary>      /// 字体样式      /// </summary>      public FontStyle FontStyle      {          get { return \_fontStyle; }          set { \_fontStyle = value; }      }        private float \_fontSize = 14f;      /// <summary>      /// 字体大小      /// </summary>      public float FontSize      {          get { return \_fontSize; }          set { \_fontSize = value; }      }      #endregion  } |

　水印位置枚举

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | /// <summary>  /// 水印位子  /// </summary>  public enum WaterMarkLocationEnum  {      /// <summary>      /// 顶部居左      /// </summary>      TopLeft = 1,      /// <summary>      /// 顶部居中      /// </summary>      TopCenter = 2,      /// <summary>      /// 顶部居右      /// </summary>      TopRight = 3,      /// <summary>      /// 中部居左      /// </summary>      CenterLeft = 4,      /// <summary>      /// 中部居中      /// </summary>      CenterCenter = 5,      /// <summary>      /// 中部居右      /// </summary>      CenterRight = 6,      /// <summary>      /// 底部居左      /// </summary>      BottomLeft = 7,      /// <summary>      /// 底部居左      /// </summary>      BottomCenter = 8,      /// <summary>      /// 底部居左      /// </summary>      BottomRight = 9,  } |

　水印类型枚举

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | /// <summary>      /// 水印类型      /// </summary>      public enum WaterMarkTypeEnum      {          /// <summary>          /// 文字          /// </summary>          Text = 1,          /// <summary>          /// 图片          /// </summary>          Image = 2,          /// <summary>          /// 无水印          /// </summary>          NoneMark = 3      } |

　水印帮助类（自己封装的，有bug可以像我反馈）

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243 | /// <summary>  /// 水印帮助类  /// </summary>  public class WaterMarkHelper  {      #region 设置水印      /// <summary>      /// 设置水印      /// </summary>      /// <param name="imgPath"></param>      /// <param name="model"></param>      /// <returns></returns>      public static Image SetWaterMark(string imgPath, WaterMark model)      {          //获取背景图          Image imgSource = Image.FromFile(imgPath);          //获取水印图片          Image markImg = null;            //水印检验（文字，图片[路径下是否存在图片]）          #region 水印校验+水印处理          if (model == null)          {              return null;          }          //看看原图是否存在          if (!System.IO.File.Exists(imgPath))          {              return null;          }          //根据水印类型校验+水印处理          switch (model.WaterMarkType)          {              case WaterMarkAPP.Enums.WaterMarkTypeEnum.Text:                  if (string.IsNullOrEmpty(model.Text))                  {                      return null;                  }                  else                  {                      markImg = TextToImager(model);//水印处理-如果是文字就转换成图片                  }                  break;              case WaterMarkAPP.Enums.WaterMarkTypeEnum.Image:                  if (!System.IO.File.Exists(model.ImgPath))                  {                      return null;                  }                  else                  {                      //获得水印图像                      markImg = Image.FromFile(model.ImgPath);                  }                  break;              case WaterMarkAPP.Enums.WaterMarkTypeEnum.NoneMark:                  return imgSource;          }          #endregion            #region 创建颜色矩阵          //创建颜色矩阵          float[][] ptsArray ={                               new float[] {1, 0, 0, 0, 0},                               new float[] {0, 1, 0, 0, 0},                               new float[] {0, 0, 1, 0, 0},                               new float[] {0, 0, 0, model.Transparency, 0}, //注意：0.0f为完全透明，1.0f为完全不透明                               new float[] {0, 0, 0, 0, 1}};          ColorMatrix colorMatrix = new ColorMatrix(ptsArray);          //新建一个Image属性          ImageAttributes imageAttributes = new ImageAttributes();          //将颜色矩阵添加到属性          imageAttributes.SetColorMatrix(colorMatrix, ColorMatrixFlag.Default, ColorAdjustType.Default);          #endregion            //原图格式检验+水印          #region 原图格式检验+水印            //判断是否是索引图像格式          if (imgSource.PixelFormat == PixelFormat.Format1bppIndexed || imgSource.PixelFormat == PixelFormat.Format4bppIndexed || imgSource.PixelFormat == PixelFormat.Format8bppIndexed)          {              #region 索引图片,转成位图再加图片              //转成位图,这步很重要              Bitmap bitmap = new Bitmap(imgSource.Width, imgSource.Height);              Graphics graphic = Graphics.FromImage(bitmap);                #region 缩放处理              //如果原图小于水印图片 等比缩放水印图              if (markImg.Width >= imgSource.Width || markImg.Height >= imgSource.Height)              {                  markImg = ImageShrink(imgSource, markImg);              }              #endregion                #region 水印位置              //水印位置              int x;              int y;              WaterMarkLocations(model, imgSource, markImg, out x, out y);              #endregion                //将原图画在位图上              graphic.DrawImage(imgSource, new Point(0, 0));                //将水印加在位图上              graphic.DrawImage(markImg, new Rectangle(x, y, markImg.Width, markImg.Height), 0, 0, markImg.Width, markImg.Height, GraphicsUnit.Pixel, imageAttributes);                graphic.Dispose();              return bitmap;              #endregion          }          else          {              #region 非索引图片,直接在上面加上水印              Graphics graphic = Graphics.FromImage(imgSource);                #region 缩放处理              //如果原图小于水印图片 等比缩放水印图              if (markImg.Width >= imgSource.Width || markImg.Height >= imgSource.Height)              {                  markImg = ImageShrink(imgSource, markImg);              }              #endregion                #region 水印位置              //水印位置              int x;              int y;              WaterMarkLocations(model, imgSource, markImg, out x, out y);              #endregion                //将水印加在原图上              graphic.DrawImage(markImg, new Rectangle(x, y, markImg.Width, markImg.Height), 0, 0, markImg.Width, markImg.Height, GraphicsUnit.Pixel, imageAttributes);                graphic.Dispose();              return imgSource;              #endregion          }          #endregion      }      #endregion        #region 水印处理-文字转图片      /// <summary>      /// 水印处理-文字转图片      /// </summary>      /// <param name="model"></param>      /// <returns></returns>      private static Image TextToImager(WaterMark model)      {          Font f = new Font(model.FontFamily, model.FontSize, model.FontStyle);          Bitmap fbitmap = new Bitmap(Encoding.GetEncoding("GBK").GetByteCount(model.Text) / 2 \* f.Height, f.Height);          Graphics gh = Graphics.FromImage(fbitmap);//创建一个画板;          gh.SmoothingMode = SmoothingMode.AntiAlias;          gh.DrawString(model.Text, f, model.BrushesColor, 0, 0);//画字符串          return fbitmap as Image;      }      #endregion        #region 水印位置      /// <summary>      /// 水印位置      /// </summary>      /// <param name="model"></param>      /// <param name="imgSource"></param>      /// <param name="markImg"></param>      /// <param name="x"></param>      /// <param name="y"></param>      private static void WaterMarkLocations(WaterMark model, Image imgSource, Image markImg, out int x, out int y)      {          x = 0;          y = 0;          switch (model.WaterMarkLocation)          {              case WaterMarkLocationEnum.TopLeft:                  x = 0;                  y = 0;                  break;              case WaterMarkLocationEnum.TopCenter:                  x = imgSource.Width / 2 - markImg.Width / 2;                  y = 0;                  break;              case WaterMarkLocationEnum.TopRight:                  x = imgSource.Width - markImg.Width;                  y = 0;                  break;              case WaterMarkLocationEnum.CenterLeft:                  x = 0;                  y = imgSource.Height / 2 - markImg.Height / 2;                  break;              case WaterMarkLocationEnum.CenterCenter:                  x = imgSource.Width / 2 - markImg.Width / 2;                  y = imgSource.Height / 2 - markImg.Height / 2;                  break;              case WaterMarkLocationEnum.CenterRight:                  x = imgSource.Width - markImg.Width;                  y = imgSource.Height / 2 - markImg.Height / 2;                  break;              case WaterMarkLocationEnum.BottomLeft:                  x = 0;                  y = imgSource.Height - markImg.Height;                  break;              case WaterMarkLocationEnum.BottomCenter:                  x = imgSource.Width / 2 - markImg.Width / 2;                  y = imgSource.Height - markImg.Height;                  break;              case WaterMarkLocationEnum.BottomRight:                  x = imgSource.Width - markImg.Width;                  y = imgSource.Height - markImg.Height;                  break;          }      }      #endregion        #region 缩放水印      /// <summary>      /// 等比缩放水印图（缩小到原图的1/3）      /// </summary>      /// <param name="imgSource"></param>      /// <param name="successImage"></param>      /// <returns></returns>      private static Image ImageShrink(Image imgSource, Image markImg)      {          int w = 0;          int h = 0;            Image.GetThumbnailImageAbort callb = null;            //对水印图片生成缩略图,缩小到原图的1/3(以短的一边为准)          if (imgSource.Width < imgSource.Height)          {              w = imgSource.Width / 3;              h = markImg.Height \* w / markImg.Width;          }          else          {              h = imgSource.Height / 3;              w = markImg.Width \* h / markImg.Height;          }          markImg = markImg.GetThumbnailImage(w, h, callb, new System.IntPtr());          return markImg;      }      #endregion  } |

　　简单调用：

文字水印调用：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | /// <summary>  /// 文字水印调用  /// </summary>  /// <param name="sender"></param>  /// <param name="e"></param>  private void Button\_Click(object sender, RoutedEventArgs e)  {      //给水印对象赋对应的值      WaterMark waterMark = new WaterMark();      waterMark.WaterMarkType = Enums.WaterMarkTypeEnum.Text;      waterMark.Transparency = 0.7f;      waterMark.Text = "dunitian.cnblogs.com";      waterMark.FontStyle = System.Drawing.FontStyle.Bold;      waterMark.FontFamily = "Consolas";      waterMark.FontSize = 20f;      waterMark.BrushesColor = System.Drawing.Brushes.YellowGreen;      waterMark.WaterMarkLocation = Enums.WaterMarkLocationEnum.CenterCenter;        //调用      Image successImage = WaterMarkHelper.SetWaterMark("text.png", waterMark);      //保存      successImage.Save("text1.png", System.Drawing.Imaging.ImageFormat.Png);        MessageBox.Show("请查看软件根目录", "成功");  } |

　　图片水印调用：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | /// <summary>          /// 图片水印调用          /// </summary>          /// <param name="sender"></param>          /// <param name="e"></param>          private void Button\_Click\_1(object sender, RoutedEventArgs e)          {              //给水印对象赋对应的值              WaterMark waterMark = new WaterMark();              waterMark.WaterMarkType = Enums.WaterMarkTypeEnum.Image;              waterMark.ImgPath = "水印.png";              waterMark.WaterMarkLocation = Enums.WaterMarkLocationEnum.CenterCenter;              waterMark.Transparency = 0.7f;                //调用              Image successImage = WaterMarkHelper.SetWaterMark("text.png", waterMark);              //保存              successImage.Save("text2.png", System.Drawing.Imaging.ImageFormat.Png);                MessageBox.Show("请查看软件根目录","成功");          } |

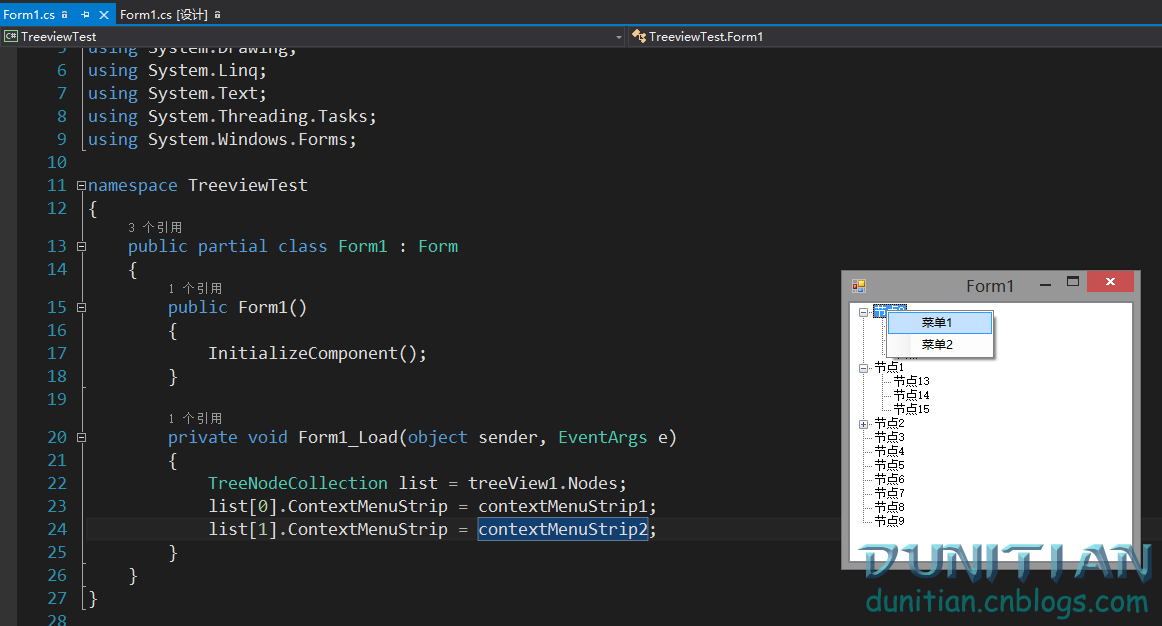
　　软件源码：（helper类不是最新，用上文的helper类）

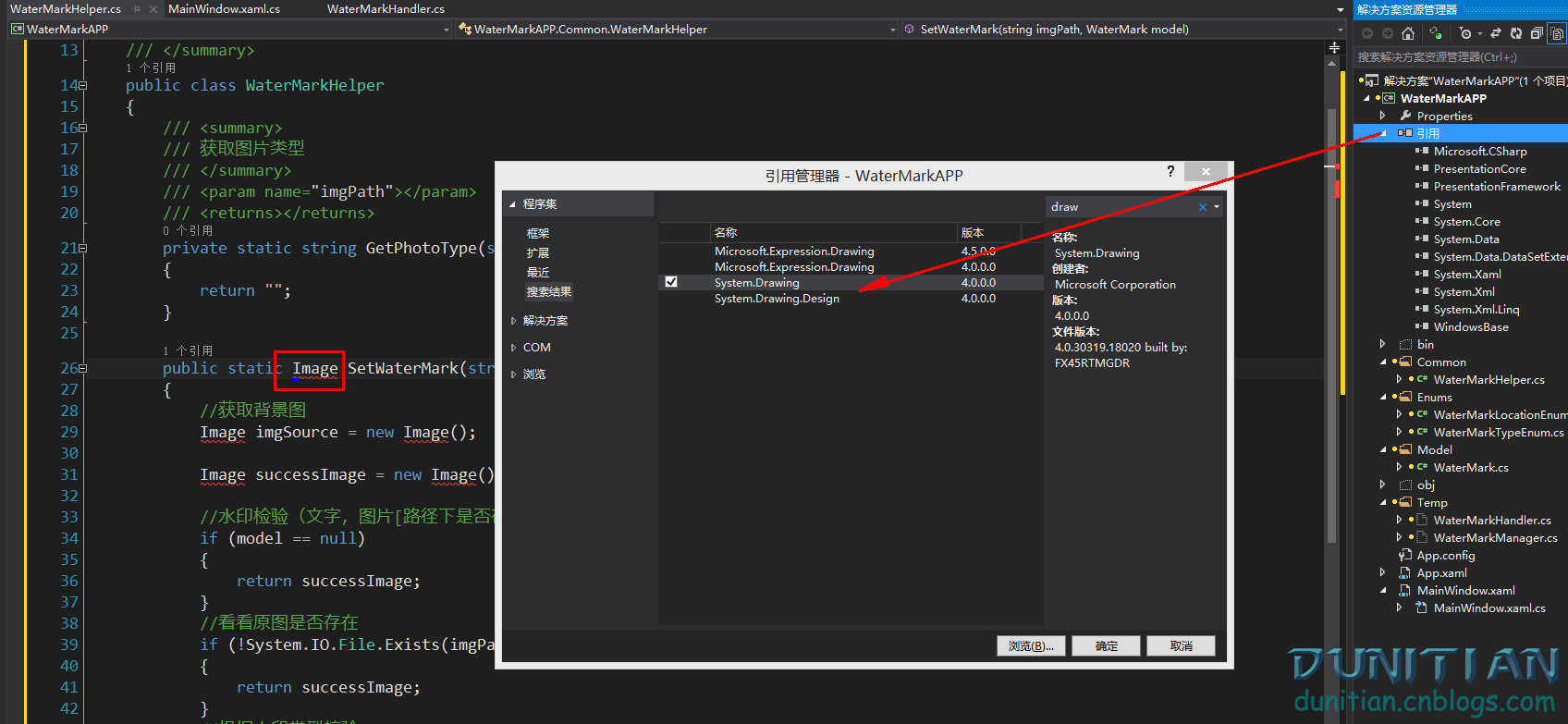
(调用部分我做了一个通用apidemo)==><http://pan.baidu.com/s/1gd1300z>

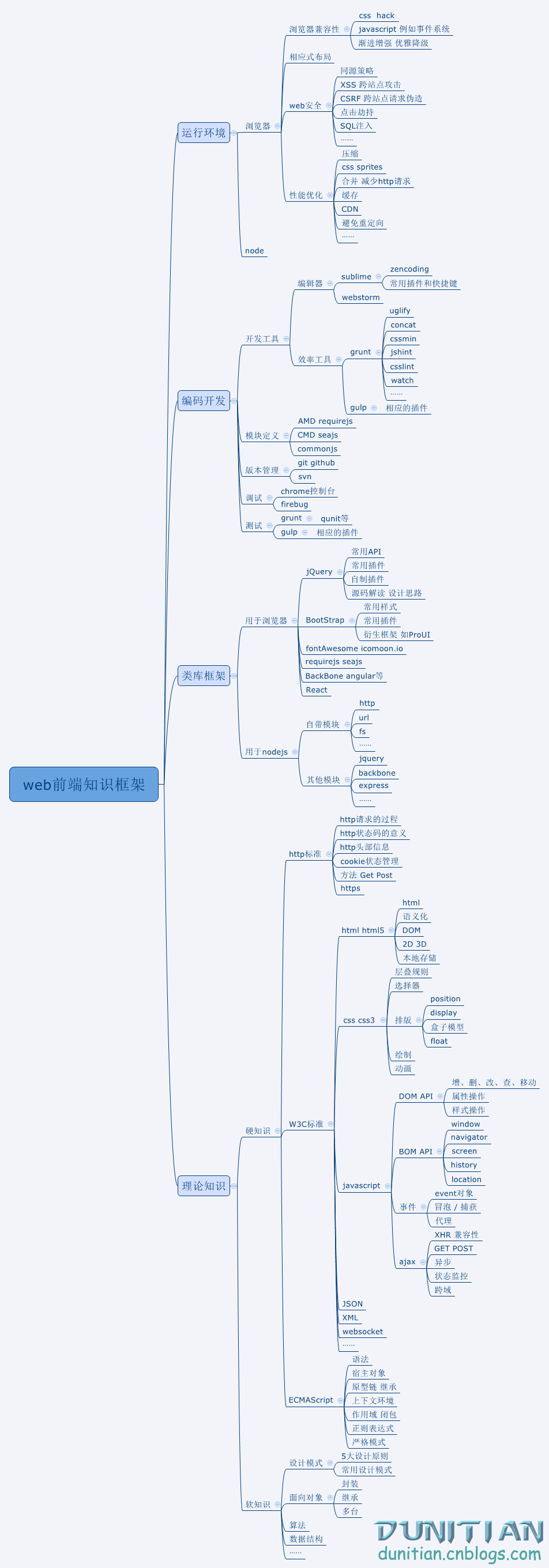
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249 | namespace WaterMarkAPP  {      /// <summary>      /// MainWindow.xaml 的交互逻辑      /// </summary>      public partial class MainWindow : Window      {          public MainWindow()          {              InitializeComponent();          }            #region 按钮事件          /// <summary>          /// 单文水印          /// </summary>          /// <param name="sender"></param>          /// <param name="e"></param>          private void BtnFont\_Click(object sender, RoutedEventArgs e)          {              WaterMark waterMark = WaterMarkFont();              DIVWaterMark(waterMark);          }            /// <summary>          /// 批文水印          /// </summary>          /// <param name="sender"></param>          /// <param name="e"></param>          private void BtnFonts\_Click(object sender, RoutedEventArgs e)          {              WaterMark waterMark = WaterMarkFont();              DIVWaterMarks(waterMark);          }            /// <summary>          /// 单图水印          /// </summary>          /// <param name="sender"></param>          /// <param name="e"></param>          private void BtnImage\_Click(object sender, RoutedEventArgs e)          {              WaterMark waterMark = WaterMarkImage();              DIVWaterMark(waterMark);          }            /// <summary>          /// 批图水印          /// </summary>          /// <param name="sender"></param>          /// <param name="e"></param>          private void BtnImages\_Click(object sender, RoutedEventArgs e)          {              WaterMark waterMark = WaterMarkImage();              DIVWaterMarks(waterMark);          }          #endregion            #region 关闭程序          /// <summary>          /// 关闭程序          /// </summary>          /// <param name="sender"></param>          /// <param name="e"></param>          private void Button\_Click(object sender, RoutedEventArgs e)          {              Environment.Exit(0);          }          #endregion            #region 窗体拖动          /// <summary>          /// 窗体拖动          /// </summary>          /// <param name="sender"></param>          /// <param name="e"></param>          private void DragWindow(object sender, MouseButtonEventArgs e)          {              if (e.LeftButton == MouseButtonState.Pressed)              {                  DragMove();              }          }          #endregion            #region 版权系列          private void TextBlock\_MouseLeave(object sender, MouseEventArgs e)          {              Process.Start("<http://dunitian.cnblogs.com/>");          }            private void TextBlock\_MouseLeave\_1(object sender, MouseEventArgs e)          {              Process.Start("<http://tieba.baidu.com/f?kw=>毒逆天");          }            private void TextBlock\_MouseLeave\_2(object sender, MouseEventArgs e)          {              Process.Start("<http://1054186320.qzone.qq.com/>");          }          #endregion            #region 软件代码（水印核心代码请看Helper类,最简洁的调用请看APIDemo文件夹里面的内容）            #region 水印预设          /// <summary>          /// 水印文字预设          /// </summary>          /// <returns></returns>          private static WaterMark WaterMarkFont()          {              WaterMark waterMark = new WaterMark();              waterMark.WaterMarkType = Enums.WaterMarkTypeEnum.Text;              waterMark.Transparency = 0.7f;              waterMark.Text = "dunitian.cnblogs.com";              waterMark.FontStyle = System.Drawing.FontStyle.Bold;              waterMark.FontFamily = "Consolas";              waterMark.FontSize = 20f;              waterMark.BrushesColor = System.Drawing.Brushes.YellowGreen;              waterMark.WaterMarkLocation = Enums.WaterMarkLocationEnum.CenterCenter;              return waterMark;          }            /// <summary>          /// 图片水印预设          /// </summary>          /// <returns></returns>          private static WaterMark WaterMarkImage()          {              WaterMark waterMark = new WaterMark();              waterMark.WaterMarkType = Enums.WaterMarkTypeEnum.Image;              waterMark.ImgPath = "水印.png";              waterMark.WaterMarkLocation = Enums.WaterMarkLocationEnum.BottomRight;              waterMark.Transparency = 0.7f;              return waterMark;          }          #endregion            #region 水印操作          /// <summary>          /// 单个水印操作          /// </summary>          /// <param name="waterMark"></param>          private static void DIVWaterMark(WaterMark waterMark)          {              #region 必须参数获取              OpenFileDialog dialog = new OpenFileDialog              {                  Filter = "png(\*.png)|\*.png|jpg(\*.jpg)|\*.jpg|bmp(\*.bmp)|\*.bmp|gif(\*.gif)|\*.gif|jpeg(\*.jpeg)|\*.jpeg",                  Title = "打开一张图片"              };              if (dialog.ShowDialog() != true)              {                  return;              }                //图片路径              string filePath = dialog.FileName;              //文件名              string fileName = System.IO.Path.GetFileNameWithoutExtension(filePath);              //图片所处目录              string dirPath = System.IO.Path.GetDirectoryName(filePath);              //存放目录              string savePath = dirPath + "\\DNTWaterMark";              //是否存在，不存在就创建              if (!Directory.Exists(savePath))              {                  Directory.CreateDirectory(savePath);              }              #endregion                #region 水印操作              Image successImage = WaterMarkHelper.SetWaterMark(filePath, waterMark);              if (successImage != null)              {                  //保存图片（不管打不打开都保存）                  successImage.Save(savePath + "\\" + fileName + ".png", System.Drawing.Imaging.ImageFormat.Png);                  //是否打开目录                  MessageBoxResult result = MessageBox.Show("水印成功！是否打开目录？", "逆天友情提醒", MessageBoxButton.YesNo);                  if (result == MessageBoxResult.Yes)                  {                      System.Diagnostics.Process.Start("explorer.exe ", savePath);//打开保存后的路径                  }              }              else              {                  MessageBox.Show("水印失败！请检查原图和水印图！", "逆天友情提醒");              }              #endregion          }            /// <summary>          /// 批量水印操作          /// </summary>          /// <param name="waterMark"></param>          private void DIVWaterMarks(WaterMark waterMark)          {              System.Windows.Forms.FolderBrowserDialog dialog = new System.Windows.Forms.FolderBrowserDialog              {                  Description = "选择你要批量水印的图片目录"              };              if (dialog.ShowDialog() == System.Windows.Forms.DialogResult.OK)              {                  string[] files = Directory.GetFiles(dialog.SelectedPath);                  if (files.Length <= 0)                  {                      return;                  }                    #region 存储专用                  //图片所处目录                  string dirPath = System.IO.Path.GetDirectoryName(files[0]);                  //存放目录                  string savePath = dirPath + "\\DNTWaterMark";                  //是否存在，不存在就创建                  if (!Directory.Exists(savePath))                  {                      Directory.CreateDirectory(savePath);                  }                  #endregion                    int num = 0;//计数用                  foreach (string filePath in files)                  {                      //文件名                      string fileName = System.IO.Path.GetFileNameWithoutExtension(filePath);                        Image successImage = WaterMarkHelper.SetWaterMark(filePath, waterMark);                      if (successImage != null)                      {                          //保存图片                          successImage.Save(savePath + "\\" + fileName + ".png", System.Drawing.Imaging.ImageFormat.Png);                          num++;                      }                  }                    //是否打开目录                  MessageBoxResult result = MessageBox.Show("逆天友情提醒：已转换 " + num + " 张图片~是否打开目录？", "转换状态", MessageBoxButton.YesNo);                  if (result == MessageBoxResult.Yes)                  {                      System.Diagnostics.Process.Start("explorer.exe ", savePath);//打开保存后的路径                  }              }          }          #endregion            #endregion      }  } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51 | <Window x:Class="WaterMarkAPP.MainWindow"          xmlns="<http://schemas.microsoft.com/winfx/2006/xaml/presentation>"          xmlns:x="<http://schemas.microsoft.com/winfx/2006/xaml>"          Title="水印工具" Height="545" Width="298" Background="Transparent" AllowsTransparency="True"          WindowStyle="None" WindowStartupLocation="CenterScreen" MouseLeftButtonDown="DragWindow" Icon="1.ico">        <!--快乐玩耍：DIV一个手机-->      <!--手机整体-->      <Border BorderBrush="Green"              Background="Black"              BorderThickness="4"              CornerRadius="10,10,10,10">          <Grid>              <!--主屏幕-->              <Border BorderBrush="Gray"                  BorderThickness="2"                  Margin="10,50,10,40">                  <Grid Background="#FF151515">                      <Grid.RowDefinitions>                          <RowDefinition></RowDefinition>                          <RowDefinition></RowDefinition>                          <RowDefinition></RowDefinition>                          <RowDefinition></RowDefinition>                          <RowDefinition></RowDefinition>                          <RowDefinition></RowDefinition>                          <RowDefinition></RowDefinition>                      </Grid.RowDefinitions>                      <TextBlock FontFamily="微软雅黑">对不起，系统已挂请联系逆天重装Win10！</TextBlock>                      <Button Grid.Row="1" Opacity="0.6"  HorizontalAlignment="Center" VerticalAlignment="Center" Height="50" Width="100" Click="BtnFont\_Click">单文水印</Button>                      <Button Grid.Row="2" Opacity="0.6"  HorizontalAlignment="Center" VerticalAlignment="Center" Height="50" Width="100" Click="BtnImage\_Click">单图水印</Button>                      <Button Grid.Row="3" Opacity="0.6"  HorizontalAlignment="Center" VerticalAlignment="Center" Height="50" Width="100" Click="BtnFonts\_Click">批文水印</Button>                      <Button Grid.Row="4" Opacity="0.6"  HorizontalAlignment="Center" VerticalAlignment="Center" Height="50" Width="100" Click="BtnImages\_Click">批图水印</Button>                      <Button Grid.Row="5" Opacity="0.6"  HorizontalAlignment="Center" VerticalAlignment="Center" Height="50" Width="100" Click="Button\_Click">关闭程序</Button>                  </Grid>              </Border>              <!--底部-->              <Border BorderBrush="Black"                  BorderThickness="2"                  CornerRadius="0,0,10,10"                  VerticalAlignment="Bottom"                  Height="40">                  <!--底部按键-->                  <Grid>                      <TextBlock Text="毒" HorizontalAlignment="Left" VerticalAlignment="Center" Margin="80,0,0,0" Width="20" Foreground="Gray" MouseLeave="TextBlock\_MouseLeave" />                      <TextBlock Text="逆" HorizontalAlignment="Center" VerticalAlignment="Center" Width="20" Foreground="Gray"  MouseLeave="TextBlock\_MouseLeave\_1"/>                      <TextBlock Text="天" HorizontalAlignment="Right" VerticalAlignment="Center" Margin="0,0,80,0" Width="20" Foreground="Gray"  MouseLeave="TextBlock\_MouseLeave\_2"/>                  </Grid>              </Border>          </Grid>      </Border>  </Window> |

　效果：











　源码下载：<http://pan.baidu.com/s/1gd1300z>