

یکی از مواردی که حین کار کردن با iTextSharp واقعا اعصاب خردکن است، طراحی نامناسب ثوابت این کتابخانه می‌باشد. برای مثال:

```
public class PdfWriter
{
    /** A viewer preference */
    public const int PageLayoutSinglePage = 1;
    /** A viewer preference */
    public const int PageLayoutOneColumn = 2;
    /** A viewer preference */
    public const int PageLayoutTwoColumnLeft = 4;
    /** A viewer preference */
    public const int PageLayoutTwoColumnRight = 8;
    /** A viewer preference */
    public const int PageLayoutTwoPageLeft = 16;
    /** A viewer preference */
    public const int PageLayoutTwoPageRight = 32;

    // page mode (section 13.1.2 of "iText in Action")

    /** A viewer preference */
    public const int PageModeUseNone = 64;
    /** A viewer preference */
    public const int PageModeUseOutlines = 128;
    /** A viewer preference */
    public const int PageModeUseThumbs = 256;
    /** A viewer preference */
    public const int PageModeFullScreen = 512;
    /** A viewer preference */
    public const int PageModeUseOC = 1024;
    /** A viewer preference */
    public const int PageModeUseAttachments = 2048;

    //...
    //...
}
```

6 ثابت اول مربوط به گروه PageLayout هستند و 6 ثابت دوم به گروه PageMode ارتباط دارند و این کلاس پر است از این نوع ثوابت (این کلاس نزدیک به 3200 سطر است!). این نوع طراحی نامناسب است. بجای گروه بندی خواص یا ثوابت با یک پیشوند، مثلا PageLayout یا PageMode، این‌ها را به کلاس‌ها یا در اینجا (حین کار با ثوابت عددی) به enumهای متناظر خود منتقل و Refactor کنید. مثلا:

```
public enum ViewerPageLayout
{
    SinglePage = 1,
    OneColumn = 2,
    TwoColumnLeft = 4,
    TwoColumnRight = 8,
    TwoPageLeft = 16,
    TwoPageRight = 32
}
```

- طبقه بندی منطقی ثوابت در یک enum و گروه بندی صحیح آنها، بجای گروه بندی توسط یک پیشوند
- استفاده بهینه از intellisense در visual studio
- منسوخ سازی استفاده از اعداد بجای معرفی ثوابت خصوصا عددی (در این کتابخانه شما می توانید بنویسید PdfWriter.PageLayoutSinglePage و یا 1 و هر دو صحیح هستند؛ این خوب نیست. ترویج استفاده از اصطلاحا magic numbers هم حین طراحی یک کتابخانه مذموم است).
- کم شدن حجم کلاس اولیه (مثلا کلاس PdfWriter در اینجا) و در نتیجه نگهداری ساده تر آن در طول زمان