Version 1.1

|  |  |  |  |
| --- | --- | --- | --- |
| تاريخ | نسخه | توضيحات | نويسنده |
| 24/8/1388 | Version 1.1 | پردازش ترددها | محمد صلواتی |

Contents

[1. مقدمه 3](#_Toc291491370)

[2. :DoMap Method 3](#_Toc291491371)

[خط54 5](#_Toc291491372)

[خطوط 62-67 5](#_Toc291491373)

[خطوط 67-76 6](#_Toc291491374)

[خط 77 6](#_Toc291491375)

[خطوط 90-79 6](#_Toc291491376)

[خط 91 6](#_Toc291491377)

[3. BasicTrafficController 7](#_Toc291491378)

[4. نیمه شب مجازی Virtual mid night 8](#_Toc291491379)

[5. ProceedTraficController 9](#_Toc291491380)

[5-T Methods 9](#_Toc291491381)

[T1 5-1: 10](#_Toc291491382)

[خط 103 11](#_Toc291491383)

[خطوط 137-108 11](#_Toc291491384)

[خطوط 153-140 12](#_Toc291491385)

[خط157 12](#_Toc291491386)

[T2: 12](#_Toc291491387)

[T3: 13](#_Toc291491388)

[T4: 16](#_Toc291491389)

[T5: 17](#_Toc291491390)

[T6: 18](#_Toc291491391)

[T7: 19](#_Toc291491392)

[T9: 20](#_Toc291491393)

[T10: 21](#_Toc291491394)

# مقدمه

فرایند پردازش تردد بمعنی انجام یکسری عملیات بر روی ترددهای پایه بمنظور رسیدن به اهداف زیر میباشد:

* دسته بندی این ترددهای پایه بصورت زوج مرتب با توجه به پیشکارت آنها
* معین کردن تاریخ این زوجها با توجه به شیفت و پیشکارت آنها

این فرایند در کلاس TrafficMapper صورت میگیرد و دارای سه بخش کلی مباشد.

* DoMap Method
* BasicTrafficController
* ProceedTrafficController
* T Methods
* Private Service

که به توضیح هریک از این اجزا میپردازیم:

# :DoMap Method

در زیر فلوچارت مربوط به این متد آمده است و اما قبل از آن لازم است به چند نکته که برای فهم فلوچارت مورد نیاز است اشاره کنیم:

در این متد ما به لیستی از ترددهای پردازش نشده مربوط به یک شخص را نیاز داریم.برای سهولت در استفاده از این لیست یک Controller در کلاس Person تعریف کرده ایم .لذا برای دسترسی به این لیست تنها از این کنترولر استفاده میکنیم.این کنترل دارای سرویسهایی است که توضیحات آن در قسمت مربوط به خود آمده است اما نکته قابل توجه این که در فلوچارت زیر این کنترولر با عنوان BasicTrafficController یاد شده است.

در فلوچارت از یک سری متد استفاده شده است که این متد‌ها با توجه‌ به دسته‌بندی بالا در قسمت Private Services توضیح داده خواهند شد

قوانینی که در پردازش ترددها استفاده میشود به دو دسته تقسیم میشوند:

* قوانینی که برروی تردد پردازش نشده اعمال میشوند که هدف آنها پیدا کردن زوج یک تردد است
* قوانینی که بر روی زوج مرتبهای یک تردد پردازش شده اعمال میشوند مانند خذف ترددهایی با طول زیر 6 دقیقه

در فلوچارت منظور از RuleOnProceedTrafficRunner متدی است که قوانین مربوط به دسته دوم بند بالا را اجرا میکند

در فلوچارت منظور از RuleOnBasicTrafficRunner متدی است که قوانین مربوط به دسته اول بند بالا را اجرا میکند

## DoMap Flowchart2.jpg



### خط54

مقداردهی اولیه متغیرها ,reset کردن proceedTraffic بمنظور اضافه کردن زوجها به آن , مرتب سازی ترددهای پردازش نشده و دریافت نزدیکترین تاریخ شیفت (GetNearShiftDate) به آیتم جاری کنترلر تردد پردازش نشده و ذخیره آن در متغیری بنام nearShiftDate. .

### خطوط 62-67

اگر nearShiftDate برابر null نباشد به این معنی است که توانسته‌ایم برای اولینbasicTraffic یک تاریخ پیدا کنیم.در مرحله بعد نوبت به پردازش ترددها میرسد و یک به یک آنها را پیمایش میکنیم.برای این منظور وارد حلقه while میشویم و تا زمانی که آیتمی وجود داشته باشد که به ترددی انتساب داده نشده باشد این کار را ادامه میدهیم.

در خط بعدی بازای هر آیتم (که ایتم جاری کنترلر میباشد) تابع GetNearShiftDate صدا میزنیم تا یک تاریخ برای ما محاسبه کند(چگونگی این محاسبه در قسمت جداگانه امده است) و در tmpShiftDate ذخیره میکنیم.

حال tmpShiftDate و nearShiftDate را باهم مقایسه میکنیم. در صورتی که این دو برابر باشند بدین معنی است که باید زوج مرتب بدست آمده را در همان proceedTraffic قبلی ذخیره کرد , بعبارتی این زوج مرتب با سایر زوج مرتبهای داخل proceedTraffic در یک روز یا شیفت ثبت شده‌است.در صورتی که این دو باهم برابر نباشند بدین معنی است که زوج مرتی جدید باید در دسته‌بندی مجزا از زوجهای قبلی قرار گیرد

### خطوط 67-76

اگر tmpShiftDate و nearShiftDate برابر نباشند آنگاه نوبت به اجرای این خطوط میرسد.در مرحله اول نوبت Raise شدن Event مربوط به اجرای قوانین تردد پردازش شده است.سپس proceedTraffic در person ذخیره میشود.دقت شود متغیر proceedTraffic حاوی لیتی از جوابها که همان ترددهای پردازش شده هستند میباشد.

در مرحله بعد متغیر nearShiftDate بروز رسانی میشود و سپس proceedTraffic مقداردهی اوله(Reset) میشود.

### خط 77

proceedTraffic.ProceedTrafficPaireController.ImportBasicTraffic(basicController.CurrentBasicItem, basicController.NextItem);

همانطور که میدانید proceedTraffic نیز دارای لیستی از زوج مرتبها میباشد.در این قسمت نیز برای سهولت در کار یک کنترولر برای این لیست تعریف کرده ایم که در قسمت مخصوص به آن توضیح میدهیم.

در این جا به کنترولر proceedTraffic میگوییم که آیتم جاری و آیتم بعدی پردازش نشده را با عنوان آیتم در حال ویرایش در خود import کن.کلیه قوانین که بر روی BasicTraffic عمل میکنند , تغییرات خود را برروی این آیتم در حال ویرایش اعمال میکنند تا نتایج همه قوانین در یکجا اعمال شود.

### خطوط 90-79

ابتدا بررسی میشود که آیتم جاری قبلا استفاده نشده باشد.سپس آنرا استفاده شده در نظر میگیرد.

در خط 74 Event مربوط به قوانینی که روی ترددهای پایه یا همان پردازش نشده اعمال میشوند Raise میشود.

در خط 76 تابع ImportBaseTraffic صدا زده میشود تا به ProceedTraffic Controller بگوید اجرای قوانین به پایان رسیده و آیتم در حال ویرایش را به لیست زوجهایت اضافه کن.

### خط 91

در نهایت پس از اتمام آیتم‌ها و بیرون آمدن از حلقه , proceedTaffic بررسی میشود تا در صورت دارا بودن زوج مرتب ذخیره نشده , این جوابها ذخیره شوند.

# BasicTrafficController

همانطور که در بالا اشاره شد , ما در حین عملیات پردازش با لیستی از ترددهای پایه که همان BasicTraffic ها هستند مواجه هستیم.برای سهولت کار با این لیست یک کلاس کنترولر تعریف کرده‌ایم که ما را از درگیری با جزئیات لیست دور میکند.این کلاس API های زیر را ارائه میدهد.

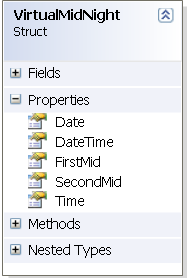
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BasicTrafficController Attributes | | | | |
| CurrentBasicItem | Property | BasicTraffic | | آیتم جاری در حال پردازش |
| NextItem | Property | BasicTraffic | آیتم بعدی | |
| BeforeItem | Property | BasicTraffic | آیتم قبلی | |
| HasNextItem | Property | Bool | آیا آیتم بعدی وجود دارد | |
| HasAfterNextItem | Property | Bool | آیا آیتم دوتا جلوتر وجود دارد | |
| CurrentIndex | Property | Int | ایندکس آیتم فعلی در آرایه | |
| AfterNextItem | Property | BasicTraffic | دوتا آیتم جلوتر | |
| Finished | Property | Bool | آیا آیتم استفاده نشده باقی مانده است | |
| MoveCurrentToNextUnusedItem | Method | Void | آیتم جاری برابر آیتم بعدی که هنوز استفاده نشده‌است قرار میگیرد | |
| InsertTrafficAfterCurentItem | Method | Void | یک آیتم بعد از آیتم جاری در لیست درج میکند | |

# نیمه شب مجازی Virtual mid night

بعلت وجود شیفتهایی که از یک روز شروع میشوند و در روز بعد به پایان میرسد نیمه شب مجازی تعریف کرده‌ایم.تعریف اصلی نیمه شب مجازی بر مبنای زیر میباشد:

هر نیمه شب مجازی بصورت منطقی شامل دو قسمت بنام های ***نیمه اول*** و **نیمه دوم** میباشد.نیمه شب مجازی باید طوری تعریف شود که

"اگر در فلان روز **شیفت** تعریف شده باشد آنگاه نیمه شب مجازی مربوط به آن روز قادر به **پوشش کامل** شیفت باشد با این خصیصه که تاریخ نیمه شب مجازی برابر ابتدای شروع شیفت باشد."کلاس نیمه شب مجازی بصورت زیر میباشد:



به نمونه‌های زیر توجه فرمایید:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| نیمه دوم | نیمه اول | نیمه شب مجازی | شیفت | تاریخ |
|  | 00:00 تا 23:59 1389/7/1 | 00:00 1/7/1389 | 7:00-16:00 | 1/7/1389 |
|  | 00:00 تا 23:59 1389/7/2 | 00:00 2/7/1389 | 7:00-16:00 | 2/7/1389 |
|  | 00:00 تا 23:59 1389/7/3 | 00:00 3/7/1389 | تعطیل شیفت | 3/7/1389 |
|  | 00:00 تا 23:59 1389/7/4 | 00:00 4/7/1389 | 7:00-16:00 | 4/7/1389 |
|  | 00:00 تا 23:59 1389/7/5 | 00:00 5/7/1389 | 7:00-16:00 | 5/7/1389 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| نیمه دوم | نیمه اول | نیمه شب مجازی | شیفت | تاریخ |
| 06:59 تا 00:00 2/7/1389 | 23:59 تا 07:00 1389/7/1 | 07:00 \/1389 | 16:00 - +7:00 | 1/7/1389 |
| 06:59 تا 00:00 3/7/1389 | 23:59 تا 07:00 1389/7/2 | 07:00 2/7/1389 | 16:00 - +7:00 | 2/7/1389 |
| 06:59 تا 00:00 4/7/1389 | 23:59 تا 07:00 1389/7/3 | 07:00 3/7/1389 | تعطیل شیفت | 3/7/1389 |
| 06:59 تا 00:00 5/7/1389 | 23:59 تا 07:00 1389/7/4 | 07:00 4/7/1389 | 16:00 - +7:00 | 4/7/1389 |
| 06:59 تا 00:00 6/7/1389 | 23:59 تا 07:00 1389/7/5 | 07:00 5/7/1389 | 16:00 - +7:00 | 5/7/1389 |
| نیمه دوم | نیمه اول | نیمه شب مجازی | شیفت | تاریخ |
| 06:59 تا 00:00 2/7/1389 | 23:59 تا 07:00 1389/7/1 | 07:00 1/7/1389 | 18:30 - +7:00 | 1/7/1389 |
| 06:59 تا 00:00 3/7/1389 | 23:59 تا 07:00 1389/7/2 | 07:00 2/7/1389 | تعطیل شیفت | 2/7/1389 |
| 06:59 تا 00:00 4/7/1389 | 23:59 تا 07:00 1389/7/3 | 07:00 3/7/1389 | 7:00-19:00 | 3/7/1389 |
| 06:59 تا 00:00 5/7/1389 | 23:59 تا 07:00 1389/7/4 | 07:00 4/7/1389 | 18:30- +7:00 | 4/7/1389 |
| 06:59 تا 00:00 6/7/1389 | 23:59 تا 07:00 1389/7/5 | 07:00 5/7/1389 | تعطیل شیفت | 5/7/1389 |

# ProceedTraficController

همانطور که میدانید proceedTraffic نیز دارای لیستی از زوج مرتبها میباشد.در این قسمت نیز برای سهولت در کار یک کنترولر برای این لیست تعریف کرده ایم. این کنترولر سرویسهای زیر را ارائه میدهد:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ProceedTrafficController Attributes | | | | |
| CurrentEdingItem | Property | ProceedTrafficPair | | آیتم جاری در حال ویرایش |
| GapBetweenCurrentAndNextItem | Property | Int | فاصله آخرین زوج و ایتم در حال ویرایش | |
| Count | Property | Int | تعداد زوجهای ذخیره شده | |
| LastItem | Property | ProceedTraficPair | آیتم آخر لیست | |
| HasNextItem | Property | Bool | آیا آیتم بعدی وجود دارد | |
| NextPaireItem | Property | ProceedTraficPair | زوج مرتب بعدی | |
| CurrentPaireItem | Property | ProceedTraficPair | زوج مرتب جاری | |
| Finished | Property | Bool | آیا آیتم استفاده نشده باقی مانده است | |
| AddBallanceTraffic | Method | Void | یک تردد 1 دقیقه ای بمنظور ایجاد تعادل ثبت میکند | |
| EditFinish | Method | Void | فیلد در حال ویرایش را به لیست اضافه میکند | |
| ImportBasicTraffic | Method | Void | مقداردهی اولیه آیتم در حال ویرایش از روی دو تردد پایه که بعنوان پارامتر دریافت میکند | |
| MergeCurrentItemWithNextItem | Method | Void | ایتم جاری و بعدی را ادغام میکند | |

## 5-T Methods

در فرایند پردازش ترددهای پایه بیشترین نقش را قوانین ایفا میکنند.این قوانین در آینده این قابلیت را دارند که توسط کاربر کم , زیاد و ویرایش شوند.در حال حاضر بصورت Template قانونینی در سیستم تعریف شده است که تعئائ آنها به 10 تا میرسد.این قوانین با پیشوند T نامگذاری شده‌اند.همچنین در بالا به این نکته اشاره شد که قوانین تردد به دو دسته "قوانین تردد پایه" و "قوتنین تردد پردازش شده" تقسیم‌بندی میشوند.درزیر قوانین را مرور میکنیم:

### T1 5-1:

#### عنوان:

اگر آیتم جاری از نوع روزانه و در آن روز ترددهای دیگری ثبت شده بود آنگاه آیتم روزانه را با ساعت از نقطه شروع تا انتهای شیفت ثبت کن همچنین تا تردد بعدی دیده شده این آیتم روزانه را هر روز تکرار کن.

#### توضیح:

آیتم روزانه آیتمی است که پیشکارت آن برابر یکی از موارد زیر باشد:

غیبت روزانه

مرخصی استعلاجی روزانه

مرخصی روزانه

مرخصی بیماری بدون حقوق روزانه

مرخصی بدون حقوق روزانه

ماموریت روزانه

و ...

#### کد:



#### توضیح کد:

### خط 103

وقتی یکی از قوانین کاری انجام میدهد که لازم است آن دوره از قوانین اجرا نشوند ,متغیر Reset را Set میکند.همه قوانین نیز در ابتدا مقدار این متغیر را بررسی میکنند و سپس کار خود را شروع میکنند.میتوان فرض کرد که متغیر Reset مانند دستور Continue در حلقه عمل میکند.

### خطوط 137-108

در صورتی که آیتم روزانه باشد و ترددهای دیگری در آن تردد پردازش شده موجود باشد آنگاه آن آیتم روزانه بمنظور انجام محاسبات آینده با ساعت ذخیره میگردد.یعنی از ساعت ثبت آن تا انتهای شیفت.در غیر این صورت آیت روزانه را بصورت معمول ذخیره میکنیم.

### خطوط 153-140

به تعداد روزهایی که شخص پس از ثبت یک آیتم روزانه در شرکت حضور ندارد(تردد ثبت نشده) برای شخص این آیتم روزانه هر روز تکرار میشود.توجه شود که در حالت معمول هیچ کارمندی اجازه و دسترسی به ثبت پیشکارت روزانه را ندارد و این عمل مستقیما توسط اپراتور سیستم در ترددهای پردازش شده درج میگردد.ولی اگر شخص چنین دسترسی را دارا باشد یعنی در ترددهای پایه آیتم روزانه موجود باشد آنگاه با آن اینچنین برخورد میشود.

### خط157

با قرار دادن Reset برابر “true” دیگر قوانین بر روی آیتم فعلی اجرا نمیشوند.

### T2:

#### عنوان:

اگر پیشکارت خروج از نوعی است که شخص دوباره به اداره بازمیگردد

یا

از نوع روزانه بود

آنگاه آیتم بعدی را استفاده نشده در نظر بگیر

#### توضیح:

برخی پیشکارتها طوری است که مطمئن هستیم شخص به شرکت باز‌میگردد ,لذا تردد بعدی باید با پیشکارت عادی و با همان زمان ورود شخص به اداره منظور شود.بطور مثال فرض کنید شخص دارای ترددهای پایه زیر میباشد:

|  |  |  |
| --- | --- | --- |
| زمان | پیشکارت | توضیح پیشکارت |
| 9:30 | 51 | ماموریت ساعتی |
| 10:50 | 51 | ماموریت ساعتی |
| 13:00 | 0 | عادی |

در این شرایط تردد پردازش شده بصورت زیر ثبت میگردد:

|  |  |  |
| --- | --- | --- |
| From | To | پیشکارت |
| 9:30 | 10:50 | 51 |
| 10:50 | 13:00 | 0 |

#### کد:

if (\_person.BasicTrafficController.HasNextItem)

{

BasicTraffic basicTraf = \_person.BasicTrafficController.NextItem;

if (\_person.BasicTrafficController.NextItem.ID > 0)

{

if (IsWorkingPrecard((int)basicTraf.Precard.ID)

|| IsMounthlyORYearly((int)basicTraf.Precard.ID))

{

\_person.BasicTrafficController.NextItem.Used = false;

}

else

{

\_person.BasicTrafficController.NextItem.Used = true;

}

#### توضیح کد:

اگر خصیصه Used مربوط به آیتم بعدی برابر false قرار گیرد این آیتم استفاده نشده در نظر گرفته میشود.

### T3:

#### عنوان:

اگر آیتم فعلی و آیتم بعدی در یک تردد نگنجد آنگاه تردد را ناقص ثبت کن

یکی در میان ورود وخروج در نظر بگیر

تبصره : در دو حالت زیر تردد را ناقص ثبت نمیکنیم:

* آیتم بعدی داخل شیفتی باشد که به شیفت فعلی چسبیده است
* آیتم بعدی با پیشکارت خروج ثبت شده باشد

#### توضیح:

شرایط زیر را در نظر بگیرید:

|  |  |  |  |
| --- | --- | --- | --- |
| زمان | پیشکارت | توضیح | تاریخ |
| 13:20 | 0 | عادی | 1388-01-01 |
| +09:20 | 0 | عادی | 1388-01-02 |

بسته به شرایط وعوامل مختلف میتوان تصمیم گیری‌های مختلفی برای آیتم اول مبنی بر زوج شدن یا ناقص ثبت شدن آن اتخاذ کرد.

##### اول حالت:

|  |  |  |
| --- | --- | --- |
| Shift | | |
| From | To | Date |
| 11:00 | 23:59 | 1388-01-01 |
| 00:00 | 11:59 | 1388-01-02 |

##### نتیجه:

|  |  |  |  |
| --- | --- | --- | --- |
| From | To | پیشکارت | تاریخ |
| 13:20 | 23:59 | 0 | 01-01-1388 |
| 00:01 | 09:20 | 0 | 02-01-1388 |

حالت دوم:

|  |  |  |  |
| --- | --- | --- | --- |
| زمان | پیشکارت | توضیح | تاریخ |
| 13:20 | 0 | عادی | 1388-01-01 |
| +09:20 | 5 | خروج | 1388-01-02 |

##### نتیجه:

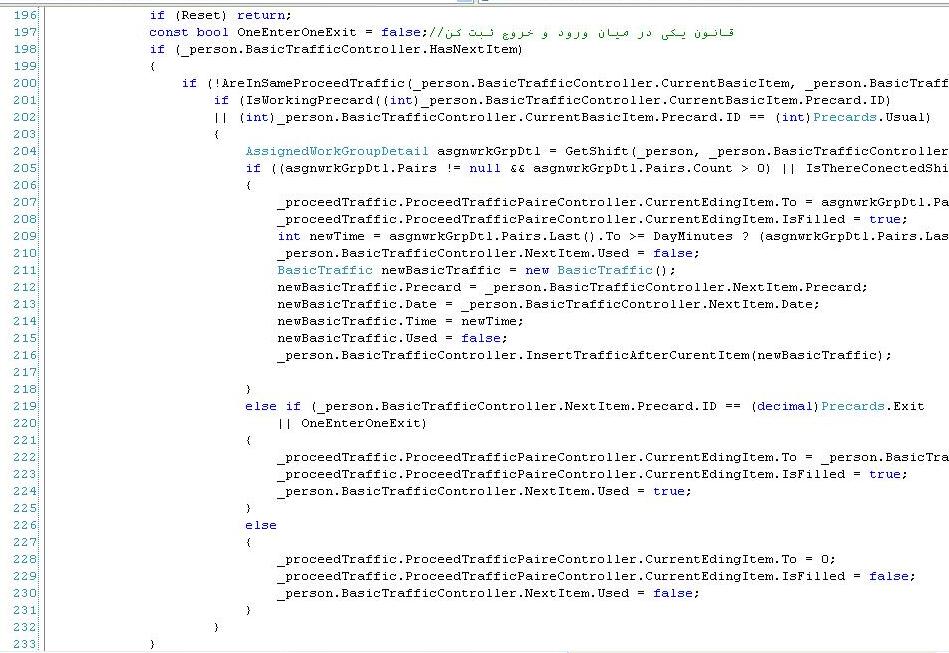
|  |  |  |  |
| --- | --- | --- | --- |
| From | To | پیشکارت | تاریخ |
| 13:20 | +09:20 | 0 | 01-01-1388 |

اما مثلا در شرایط زیر باید تردد راناقص ثبت کنیم:

|  |  |  |
| --- | --- | --- |
| Shift | | |
| From | To | Date |
| 15:00 | 23:59 | 1388-01-01 |
| 11:00 | 23:59 | 1388-01-01 |

|  |  |  |  |
| --- | --- | --- | --- |
| From | To | پیشکارت | تاریخ |
| 13:20 | 0 |  | 01-01-1388 |
| 09:20 | To | 0 | 02-01-1388 |

#### کد:



#### توضیح کد:

#### خطوط 200-202

سه خط اول بررسی میکنند که آیا ترددهای پایه مورد بررسی شامل این قانون میگردند یا خیر

#### خطوط 204 و 205

شیفت آیتم جاری را در صورتی که تردد واقعا درون شیفت باشد برمیگرداند البته در صورت عدم وجود شیفت null برمیگرداند

در خط 205 دو شرطی که در قسمت قبل ذکر شد بررسی میشود

#### خطوط 207-216

در این قسمت تردد مجازی ثبت با فاصله یک دقیقه ثبت میگردد.

#### خطوط 219-224

شرط یک در میان ورود وخروج و پیشکارت خروج بررسی میشود.در صورت درست بودن با بعدی جفت میشود

#### خطوط 228-231

در صورتی که شروط فوق صدق نکند تردد ناقص ثبت میشود

### T4:

#### عنوان:

اگر پیشکارت آخرین زوج مرتب ثبت شده به عنوان تردد پردازش شده از نوعی است که شخص به بیرون اداره رفته و دوباره بازمیگردد و همچنین پیشکارت آن با پیشکارت تردد پایه جاری برابر باشد آنگاه پیشکارت زوج مرتب جاری با نوع عادی ذخیره میگردد.

#### توضیح:

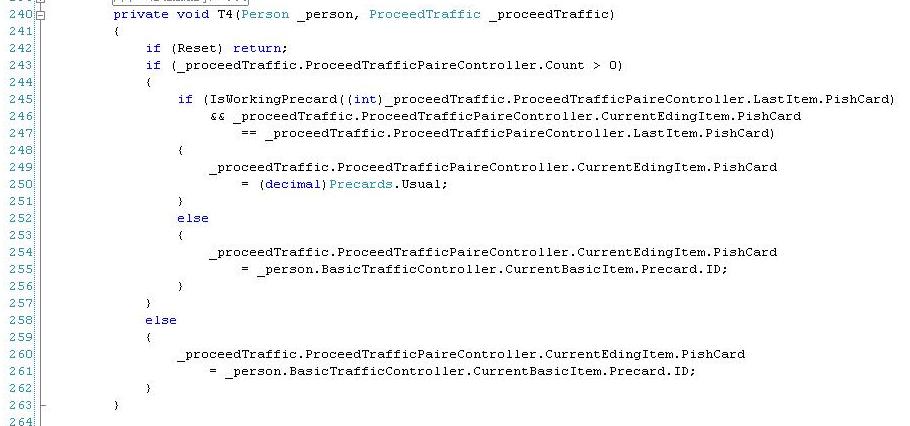
به جدول تردد‌های پایه زیر توجه کنید:

|  |  |  |  |
| --- | --- | --- | --- |
| زمان | پیشکارت | توضیح | تاریخ |
| 10:10 | 51 | ماموریت ساعتی | 1388-05-01 |
| 13:00 | 51 | ماموریت ساعتی | 1388-05-01 |
| 15:20 | 0 | عادی | 01-05-1388 |

از جدول بالا استنباط میشود که شخص ساعت 10:10 برای ماموریت شرکت را ترک کرده و ساعت 13:00 بازگشته است.پس تردد ساعت 13:00 از طرفی به عنوان پایان ماموریت و از طرفی به عنوان شروع حضور شخص در شرکت باید در نظر گرفته شود.قانون بالا بر این نکته تاکید دارد که در چنین شرایطی بدون توجه به پیشکارت تردد دوم(ساعت 13:00) , پیشکارت تردد را "عادی" در نظر بگیرید.

|  |  |  |  |
| --- | --- | --- | --- |
| From | To | پیشکارت | تاریخ |
| 10:00 | 13:00 | 51 | 01-05-1388 |
| 13:00 | 15:20 | 0 | 01-05-1388 |

#### کد:



#### توضیح کد:

این به بررسی شرط‌های مطرح شده در بالا میپردازد و سپس بعد از تصمیم گیری پیشکارت آیتم جاری را تعیین میکند

### T5:

#### عنوان:

اگر یک تردد از نوع روزانه بود انگاه آنرا لحاظ کن

#### توضیح:

در محاسبات مربوط به پردازش ترددها هرگاه به یک آیتم روزانه مانند مرخصی روزانه یا غیبت روزانه برسیم باید موارد زیر را اعمال کنیم:

تردد پایه‌ی بعدی را استفاده نشده در نظر بگیریم

در تردد پردازش شده مشخص شود که این تردد زوج مرتبی نیست

#### کد:

if (IsDaily((int)\_person.BasicTrafficController.CurrentBasicItem.Precard.ID))

{

\_proceedTraffic.ProceedTrafficPaireController.CurrentEdingItem.To = 0;

\_proceedTraffic.ProceedTrafficPaireController.CurrentEdingItem.From = 0;

\_proceedTraffic.ProceedTrafficPaireController.CurrentEdingItem.IsFilled = true;

\_proceedTraffic.ProceedTrafficPaireController.CurrentEdingItem.PishCard =

\_person.BasicTrafficController.CurrentBasicItem.Precard.ID;

\_person.BasicTrafficController.NextItem.Used = false;

\_proceedTraffic.IsPairly = false;

}

#### توضیح کد:

پس از حصول اطمینان از روزانه بودن آیتم جاری,From و To آیتم جاری در حال ویرایش را برابر صفر قرار داده و همچنین IsFilled آن برابر مقدار "درست" در نظر گرفته میشود تا در آینده برای ما قابل تشخیص باشد.

سوال:اگر این تردد از نوع زوج مرتبی نیست پس چرا برای آن زوج مرتب(0,0) را ثبت کردیم؟

جواب:این کار بدلیل محدودیتهای برنامه نویسی انجام شده است و بصورت ایده‌ال نباید زوجی برای آن ثبت شود.

### T6:

#### عنوان:

اگر همه آیتمهای یک ترددهای پردازش شده از نوع روزانه بود آنگاه آن تردد پردازش شده را روزانه فرض کن

#### توضیح:

گاهی ممکن است یک شیفت طولانی باشد(مثلا 72 ساعت) و در طول آن یک مرخصی روزانه به همراه چند زوج مرتب تردد ثبت شود.در چنین شرایطی درست نیست که ما فقط با دیدن یک تردد روزانه خصیصه زوج مرتب بودن تردد پردازش شده را برابر "نادرست" قرار دهیم.لذا لازم است قانونی باشد که با داشتن دید روی کلیه‌ی زوج مرتبهای یک تردد پردازش شده این تصمیم را اتخاذ کند.

نکته قابل توجه این که این قانون از جمله قوانین مربوط به تردد پردازش شده میباشد

قوانینی که در پردازش ترددها استفاده میشود به دو دسته تقسیم میشوند:

قوانینی که برروی تردد پردازش نشده اعمال میشوند که هدف آنها پیدا کردن زوج یک تردد است

قوانینی که بر روی زوج مرتبهای یک تردد پردازش شده اعمال میشوند مانند خذف ترددهایی با طول زیر 6 دقیقه

#### کد:

\_proceedTraffic.IsPairly = true;

foreach (ProceedTrafficPair p in \_proceedTraffic.Pairs)

{

if (IsDaily((int)p.PishCard))

{

\_proceedTraffic.IsPairly = false;

break;

}

}

if (IsDaily((int)\_proceedTraffic.ProceedTrafficPaireController.CurrentEdingItem.PishCard))

{

\_proceedTraffic.IsPairly = false;

}

#### توضیح کد:

همه زوجهای یک تردد پردازش شده را بررسی میکند و در مورد روزانه بودن آن تصمیم میگیرد به عباری حتی اگر یک زوج مرتب نیز ثبت شده باشد باید فیلد IsPailrly (که مشخص کننده روزانه بودن و نبودن-زوج مرتبی بودن یا نبودن) برابر "درست" باشد.

### T7:

#### عنوان:

اگر اولین زوج مرتب یک تردد پردازش شده طوری بود که شخص در شرکت حضور ندارد و تاخیر نداشت آنگاه یک تردد 1 دقیقه با پیشکارت عادی برای شخص در ابتدای وقت منظور کن.

اگر اولین زوج مرتب یک تردد پردازش شده طوری بود که شخص در شرکت حضور ندارد و تاخیر داشت آنگاه از ابتدای شیفت برای شخص تردد با همام پیشکارت ثبت کن.

#### توضیح:

گاهی پیش می‌آید که اولین تردد شخص در روز با پیشکارت مثلا 51(ماموریت ساعتی) است.در چنین شرایطی دو حالت پیش میآید

شحص تاخیر ندارد: در ابتدای روز برای شخص تردد 1 دقیقه‌ای بع معنای حضور در شرکت ثبت میکنیم.

شخص تاخیر دارد: از ابتدای شیفت برای شخص تردد با همام پیشکارت ثبت میکنیم.

#### کد:

#### T7.JPG

#### توضیح کد:

[ProceedTrafficController](#_ProceedTraficController) دارای سرویسی است که یک تردد مجازی 1 دقیقه‌ای برای شخص ثبت میکند.در این جا نیز این سرویس استفاده شده است.

### T9:

#### عنوان:

اگر مقدار بازه تردد از 6 دقیقه کوچکتر یا مساوی بود آنگاه انرا درنظر نگیر

#### توضیح:

ترددهای زیر را در نظر بگیرید:

|  |  |  |  |
| --- | --- | --- | --- |
| زمان | پیشکارت | توضیح | تاریخ |
| 10:10 | 0 | عادی | 1388-05-01 |
| 10:13 | 0 | عادی | 1388-05-01 |
| 10:16 | 0 | عادی | 01-05-1388 |
| 12:00 | 0 | عادی | 01-05-1388 |

که باید بصورت زیر بشود:

|  |  |  |  |
| --- | --- | --- | --- |
| From | To | پیشکارت | تاریخ |
| 10:10 | 12:00 | 0 | 01-05-1388 |

#### کد:



#### توضیح کد:

تمام زوجهای یک تردد پردازش شده پیمایش میشود.اندازه هر زوج در متغیر gap قرار داده میشود.در صورتی که کمتر از شیفت باشد , آن زوج با بعدی ادغام میشود.

### T10:

#### عنوان:

اگر به پیشکارت 28 یا 29 رسیدی تردد را ممتد در نظر بگیر و اگر تردد شامل چند شیفت شد ترددهای مجازی ثبت بکن.

#### توضیح:

در حالتی که به پیشکارت 28 یا 29 برسیم , آنقدر جلو میرویم تا به زوج آن برسیم.اگر در این مسیر از شیفتی هم عبور کنیم برای آن شیفت تردد مجازی ثبت میکنیم.

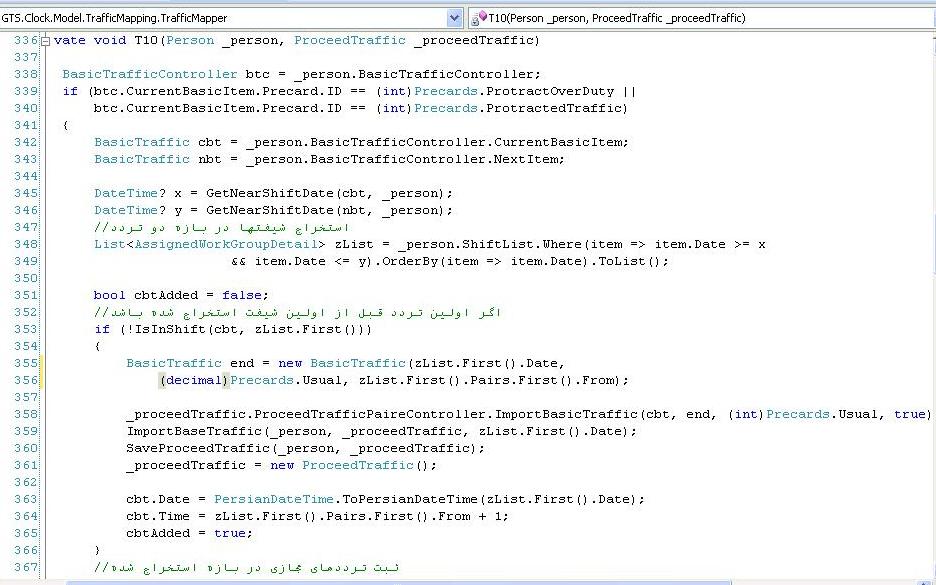
|  |  |  |
| --- | --- | --- |
| Shift | | |
| From | To | Date |
| 15:00 | 23:59 | 1388-05-01 |
| 13:00 | 09:00+ | 1388-05-02 |

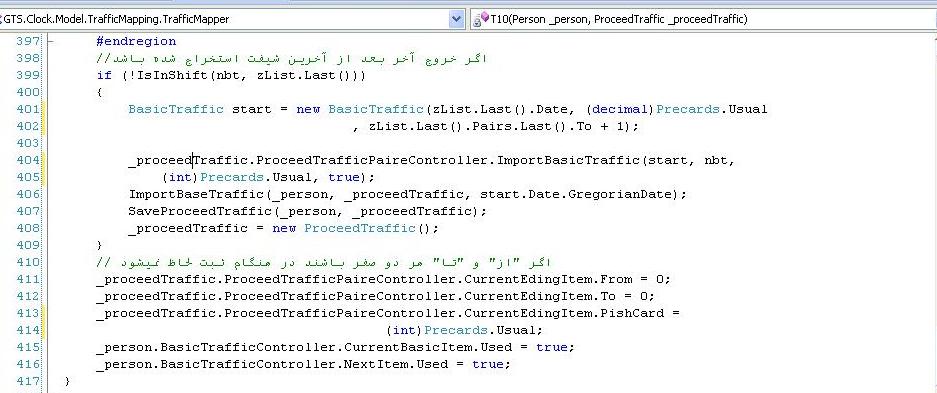
|  |  |  |  |
| --- | --- | --- | --- |
| زمان | پیشکارت | توضیح | تاریخ |
| 10:10 | 28 | ممتد | 1388-05-01 |
| 13:15 | 0 | عادی | 1388-05-03 |

نتیجه:

|  |  |  |  |
| --- | --- | --- | --- |
| From | To | پیشکارت | تاریخ |
| 10:10 | 15:00 | 0 | 01-05-1388 |
| 15:01 | 23:59 | 0 | 01-05-1388 |
| 00:00 | 13:00 | 0 | 1388-05-02 |
| 13:01 | +09:00 | 0 | 1388-05-02 |
| 09:01 | 13:15 | 0 | 1388-05-03 |

#### کد:





#### توضیح کد:

#### خطوط 342-350

آیتم فعلی و بعدی را مد نظر قرار میدهیم و با توجه به تاریخ آنها شیفتهایی را استخراج میکنیم که در این بازه قرار بگیرند

#### خطوط 353-365

اگر تردد اول قبل از اولین شیفت استخراج شده باشد باید از این تردد تا ابتدای اولین شیفت یک تردد مجازی ثبت کنیم

#### خطوط 368-396

سپس تمام شیفتهای استخراج شده را پیمایش میکنیم و با توجه به تردد دوم در صورت نیاز تردد مجازی ثبت میکنیم.

#### خطوط 398-409

همچنین اگر تردد دوم بعد از آخرین شیفت باشد از انتهای آخرین شیفت تا تردد دوم , تردد مجازی ثبت میکنیم

#### خطوط 409-416

در نهایت بدلیل اینکه ما در این قانون ترددهای لازم را ثبت کردیم دیگر آیتم جاری در حال ویرایش اعتباری ندارد پس آن را بی‌اعتبار میکنیم.همچنین آیتم جاری و بعدی را نیز مصرف شده در نظر میگیریم.