**بسه تعالی**

**سند راهنمای پروژه سیستم پرداخت حقوق**

**تهیه کننده: محمد باقر رفیعیان**

**شرح پروژه:**

در این سیستم قصد ایجاد فرآیندی را داریم که به وسیله ی آن اطلاعات مربوط به حقوق کارکنان را دریافت کند و پس از پردازش، عملیات واریز را انجام دهد. این پروژه با بهره گیری از فایل ها پیاده سازی گردید. این فایل ها به شرح زیر هستند:

فایل payment: فایل پرداخت ما است که حساب های بستانکار و بدهکار و مقادیر آنها مشخص می شود.

فایل balance: فایل موجودی حساب ها است که بعد از پردازش عملیات به روزرسانی می شود و مقدار موجودی حساب ها را نگهداری می کند.

فایل transaction: فایلی که اطلاعات تراکنش های صورت گرفته را نگهداری می کند.

**مفاهیم مورد استفاده:**

java.nio: از این پکیج که بعد از پکیج java.io ارئه شد و حاوی کلاس ها و واسط های جدیدی است برای ایجاد فایل های مورد نیاز و همچنین برای خواندن و نوشتن در آنها استفاده گردید.

multiThread: از ترد ها برای تردبندی کردن عملیات پردازش استفاده شد تا عملیات پردازش را بین چند ترد تقسیم کنیم که به طور همزمان این فرایند اجرا شده و سرعت عملیات بالاتر رود.

Enum: یک نوع داده (Data Type) در جاوا بوده که از مجموعه‌ای Constant ها تشکیل شده است. Enum ها معمولا برای داده‌های ثابت به کار می‌روند .در این پروژه برای مشخص کردن نوع شماره حساب و همچنین نوع فایل از دو Enum مختلف استفاده شده است.

Exception: در زبان جاوا به رویدادی گفته می شود که در پروسه اجرای یک برنامه یا اپلیکیشن بوجود می آید و از اجرای طبیعی برنامه جلوگیری به عمل می آورد. در این پروژه عدم موجودی کافی حساب بدهکار چک می شود و در صورت عدم موجودی کافی Exception ایجاد شده throw می شود.

در ادامه به ترتیب پکیج ها، کلاس ها و متد های مورد استفاده شرح داده می شوند.

**پکیج بندی:**

پکیج dto: این پکیج حاوی کلاس های معادل با داده های balance، payment و transaction می باشد.

پکیج exception: این پکیج حاوی کلاس Exception ایجاد شده برای handle کردن عدم موجودی کافی حساب بدهکار می باشد.

پکیج main: این پکیج حاوی کلاس Main ما می باشد.

پکیج service: این پکیج حاوی کلاس SalaryPaymentServer می باشد که حاوی متدهای سرویس ما است.

پکیج util: این پکیج حاوی کلاس هایی است که به عنوان ابزارهای مورد استفاده هستند مثل Enumها ، کلاس Thread و...

**کلاس ها:**

Main: کلاس اصلی برنامه که اجرای کد از آن آغاز می شود.

BalanceDto: این کلاس در واقع برای هر رکورد موجود در فایل موجودی استفاده می شود که حاوی شماره حساب و مقدار موجودی حساب است.

PaymentDto: این کلاس در واقع برای هر رکورد فایل پرداخت استفاده می شود که حاوی نوع حساب، شماره حساب و مقدار قابل واریز یا برداشت برای آن حساب است.

TransactionDto: این کلاس برای هر رکورد فایل تراکنش استفاده می شود که حاوی شماره حساب بدهکار، شماره حساب بستانکار و مقدار مبلغ انتقال داده شده است.

LackSufficientBalanceException: این کلاس برای تعریف یک exception جدید برای handle کردن عدم موجودی کافی در حساب بدهکار می باشد.

SalaryPaymentServer: این کلاس حاوی متدهای اصلی برنامه می باشد که در کلاس Main استفاده شده است.

ReadAndWriteFile: این کلاس حاوی متدهای خواندن و نوشتن در فایل ها می باشد.

SalaryPaymentTheared: کلاس ترد ما می باشد که حاوی متدهای مورد استفاده برای هر نخ در پردازش داده ها است.

Enumها:

DepositTypeEnum: این Enum برای تعیین نوع حساب (بدهکار، بستانکار) استفاده شده است.

FileTypeEnum: این Enum برای تعیین نوع فایل (موجودی، پرداخت و یا تراکنش) استفاده شده است.

**متد ها:**

salaryPaymentThreadsExecuter: متدی که وظیفه ترد بندی پردازش مورد نیاز، با تقسیم رکوردهای خوانده شده از فایل پرداخت بین task ها و ارسال آنها به ThreadPool را انجام می دهد. در این متد همچنین قبل از انجام هر عملیاتی موجودی کافی حساب بدهکار نیز بررسی می شود.

به دلیل اینکه فرایند ایجاد و اتمام یک نخ جدید پیچیده و پرهزینه است. برای مدیریت این کار، الگوریتم های مناسبی پیاده سازی شده است، به این اشیاء که ایجاد و اجرای نخ ها را مدیریت می کنند، اجرا گر(Executer) می گویند. کلاس های مختلفی به عنوان Executer طراحی شده اند. این اشیاء معمولا یک خزانه نخ(thread pool) دارند که تعداد m کار را به طور همزمان در n نخ اجرا می کنند.

در این پروژ، ما از کلاس کمکی Executers برای ایجاد یک threadPool با 5 نخ استفاده کرده ایم. برای ایجاد تسک ها نیز ما پردازش هر 100 سطر از فایل را به یک تسک واگذار کردیم که با توجه به تعداد سطر های فایل پرداخت، تعداد تسک ها مشخص می گردد. سپس این تسک ها به خزانه نخ ارسال شده و در آنجا در صورتی که نخ بیکاری وجود داشته باشد پردازش می شوند، در غیر این صورت منتظر می مانند تا کار یکی از نخ ها به اتمام برسد و سپس پردازش این نخ را شروع کند.

createFile: این متد برای ایجاد فایل های پرداخت و موجودی در کلاس Main استفاده می شود، در این متد از متدهای کلاس ReadAndWriteFile استفاده شده است.

:calculateSumPayments متدی برای محاسبه جمع مقادیر حساب های بستانکار در فایل پرداخت می باشد.

timeCreationFileWriter: متدی برای نوشتن در فایل ها در هنگام ایجاد فایل ها می باشد.

:createPaymentFile متدی برای ایجاد فایل پرداخت به همراه داده های مورد نیاز آن می باشد. این متد در متد timeCreationFileWriter استفاده شده است.

createBalanceFile: متدی برای ایجاد فایل موجودی به همراه داده های آن می باشد. این متد در متد timeCreationFileWriter استفاده شده است.

paymentFileReader: متدی برای خواندن داده های فایل پرداخت و تبدیل آنها به لیست می باشد.

balanceFileReader: متدی برای خواندن داده های فایل موجودی و تبدیل آنها به لیست می باشد.

balanceWriter: متدی برای نوشتن در فایل موجودی در هنگام پردازش می باشد.

transactionWriter: متدی برای نوشتن در فایل تراکنش در هنگام پردازش می باشد.