به نام خدا

گزارش تمرین سیستم های توزیعی

پیاده سازی یک سیستم فراخوانی متد از راه دور(RMI)

استاد درس : دکتر کلباسی

نگارنده : فاطمه غلام زاده – 99131003

پیش فرض ها

- چند کلاینت می توانند به طور هم زمان به سرور متصل باشند.
- کلاینت ها از آیپی و پورت سرور مطلع هستند.
- برای اجرای برنامه ابتدا سرور ران می شود و سپس کلاینت ها.
- کلاس سرور و کلاینت در دو برنامه جداگانه اجرا می شوند اما در هر دو برنامه باید در پکیجی با نام یکسان باشند (به دلیل بازسازی آبجکت های (RMIMessage) ارسال شده)
- اینترفیس های ایجاد شده باید از اینترفیس Remote ارث بری کنند.
- حساب ها و موجودی آن ها در هر بار ران شدن معنا دارند و اگر دوباره کد اجرا شود همان مقادیر اولیه را می گیرند (چون موجودی حساب ها در یک HashMap سمت سرور نگهداری می شود و از فایل خوانده نمی شود)

شرح جزئیات

- برای ارتباطات از سوکت استفاده شده است و آیپی و پورت به صورت hard code در کدها تعبیه شده است . برای ارسال نام اینترفیس، نام متد و پارامترهای ورودی و مقادیر بازگشتی (return values) متدها، یک کلاس به نام RMIMessage تعریف شده است. نام ها و پارامترها در قالب یک RMIMessage بسته بندی و ارسال می شوند.
- برای طراحی از ارث بری استفاده شده است بدین صورت که یک اینترفیس به نام Remote وجود دارد و هر اینترفیسی که برای پیاده سازی متدها ایجاد می شود هم در سمت سرور و هم در سمت کلاینت باید از اینترفیسRemote ارث بری کند.

- برای اجرای برنامه ابتدا سرور ران می شود و در طول اجرای برنامه سرور همواره در حال ران شدن است و از طریق کلاس Registry در یک حلقه while true روی پورت مورد نظر در حال گوش دادن می باشد.
- سرور بعد از دریافت درخواست از هر کلاینت یک thread جداگانه برای پاسخگویی به آن کلاینت درست می کند، بدین منظور یک کلاس وجود دارد که اینترفیس Runnable را implement می کند.
- در thread ایجاد شده، پیام دریافت شده باز می شود، متد درخواست شده شناسایی شده و با استفاده از پارامترهای ورودی که کلاینت داده است، فراخوانی می شود و مقدار بازگشتی متد مجددا بسته بندی شده و به سمت کلاینت ارسال می گردد.
- در سمت کلاینت برای دریافت نام متد و پارامترهای ورودی متد از پروکسی استفاده شده است که در کتابخانه ی java.lang.reflect.Proxy از جاوا وجود دارد.

نقاط قوت و ضعف سیستم طراحی شده

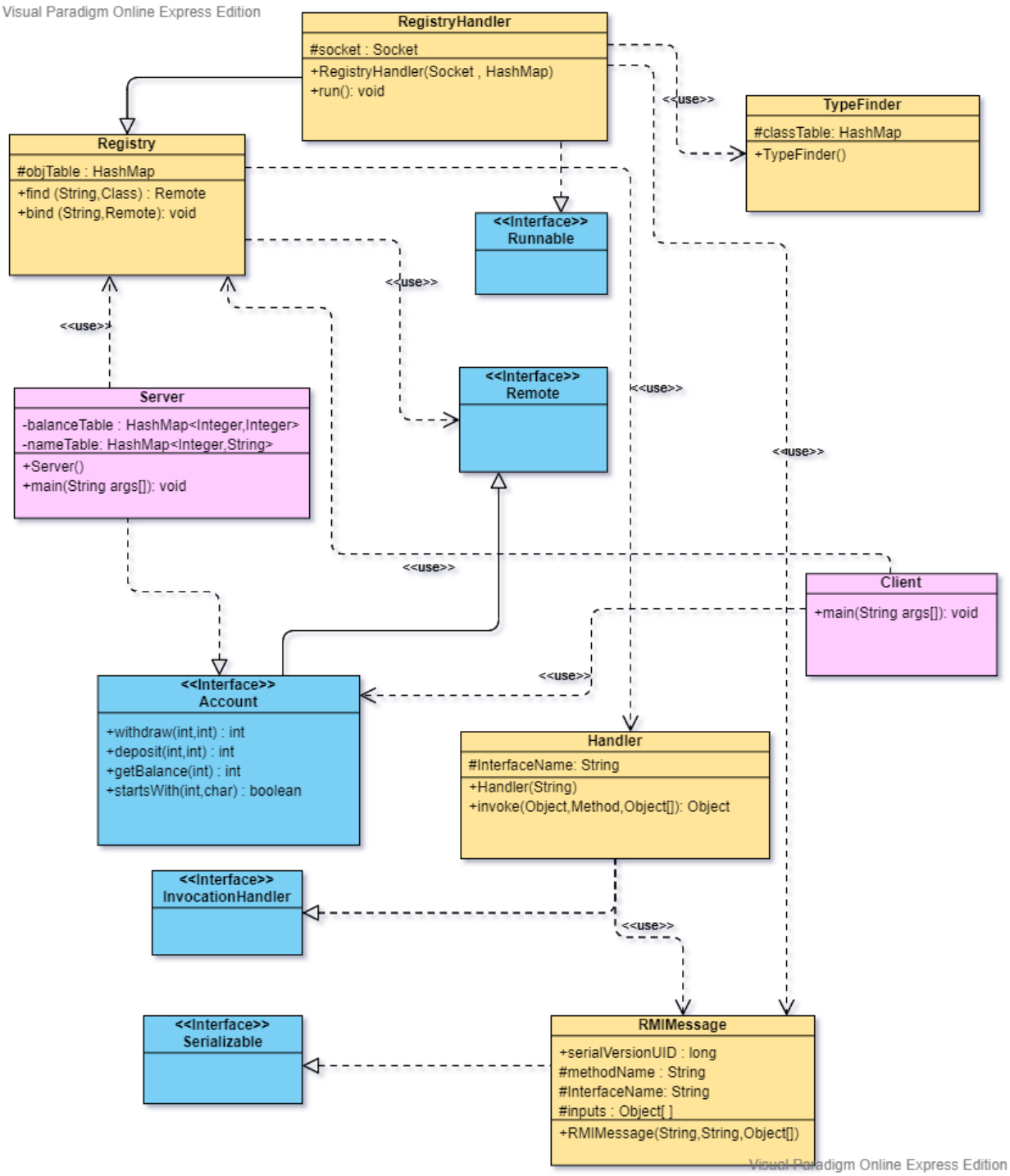
نقاط قوت :

- سیستم قابلیت این را دارد که چند کلاینت به طور هم زمان از آن استفاده کنند.
- انواع داده متعدد شامل int و char و Boolean و String و long و short و double و float و byte می توانند توسط این سیستم به عنوان پارامتر ورودی و خروجی متدها رد و بدل شوند.
- کلاینت بدون پیاده سازی و آگاهی از بدنه متدها می تواند از آن ها استفاده کند و متد را روی اشیا دور فراخوانی کند، دقیقا مشابه اینکه این متدها به صورت محلی پیاده سازی شده اند.
- سرور برای پاسخگویی به هر کلاینت یک thread جداگانه می سازد پس امکان موازی سازی هست.

نقاط ضعف :

- از dynamic binding استفاده نشده و آیپی و پورت به صورت hard code آورده شده است.
- آبجکت هایی که خود کاربر تعریف کند نمی توانند به عنوان پارامتر ورودی متدها یا مقدار خروجی آن ها رد و بدل شوند.
- سیستم برای تعداد خیلی زیاد کلاینت ها تست نشده است پس امکان وجود مشکلات احتمالی برای تعداد زیاد کلاینت وجود دارد.

❖ نمودار کلاس (class diagram)



❖ نمودار توالی (sequence diagram)

نمودار توالی برای فراخوانی متد withdraw توسط کلاینت ترسیم شده است. لازم به ذکر است سایر متدها نیز به همین شکل صدا زده می شوند و نمودار توالی آن ها به همین صورت خواهد بود.