

1) فرض کنید قرار است یک شرکت را مدل کنیم. شرکت یک مدیر، تعدادی برنامه‌نویس و تعدادی کارمند بخش اداری دارد. با استفاده از مفاهیم Object Oriented یک کلاس برای هر یک از افراد تعریف کنید. هر فرد یک نام دارد و تابعی که نام آن شخص را نمایش می‌دهد. همچنین موقع ایجاد کلاس باید نام فرد دریافت شود و هنگام Destruct شدن نام و Position فرد نمایش داده شود. مدیر یک فیلد به نام تعداد کارمند دارد. برنامه‌نویس و کارمند هم یک فیلد به اسم "نام مدیر" دارند و تابعی که این‌ها را نمایش دهد. هر فرد یک تابع به نام WritePosition دارد که باید در آن صرفاً در کنسول نوشته شود که مدیر است، برنامه‌نویس است یا کارمند بخش اداری. این تابع باید برای تمامی افراد فراخوانده شود. هم چنین یک سیستم اطلاع‌رسانی داریم که به دو روش می‌تواند اطلاع‌رسانی را انجام دهد. از طریق Email یا از طریق SMS. برای هر کدام یک کلاس در نظر بگیرید که باید تابعی به نام inform داشته باشند که صرفاً در آن فقط نمایش داده شود که از کدام روش اطلاع‌رسانی استفاده شده است.

2) یک کلاس به نام HugeInteger ایجاد کنید که از یک آرایه‌ی 40 تایی برای ذخیره اعداد تا 40 رقم استفاده کند. این کلاس باید امکان ذخیره‌ی اعداد مثبت و منفی را داشته باشد.

برای این کلاس متدها و Property های زیر را ایجاد کنید:

- دو Constructor به صورت زیر:
 - یک Default Constructor که یک HugeInteger با مقدار صفر را ایجاد کند.
 - یک Constructor که یک string دریافت کند و آن را در HugeInteger ذخیره کند. باید بررسی شود که فرمت ورودی درست باشد و فقط شامل ارقام و علامت + یا - در ابتدای آن باشد. همچنین بررسی شود که تعداد ارقام از 40 رقم بیشتر نباشد. در صورت درست نبودن ورودی باید پیغام مناسبی نمایش داده شده و مقدار HugeInteger برابر صفر در نظر گرفته شود.
- متد ToString را Override کنید که HugeInteger را به فرمت string در بیاورد.
- 3 متد زیر برای مقایسه‌ی HugeInteger ها که باید true یا false برگردانند.
 - Equals
 - IsGreaterThan
 - IsLessThan

پس از نوشتن کلاس HugeInteger در تابع Main برنامه‌ای بنویسید که یک Loop داشته باشد که هر بار دو عدد از ورودی دریافت کند و برای هر کدام یک HugeInteger ایجاد کرده و آن‌ها را با هم مقایسه کند و نتایج را نشان دهد. برنامه در صورتی خاتمه پیدا می‌کند که یکی از اعداد ورودی empty string باشد.

3) برنامه‌ای بنویسید که دو ماتریس دریافت کند و ضرب آن‌ها را انجام دهد. ضرب ماتریس‌ها باید به صورت موازی انجام شود به این صورت که برای محاسبه‌ی هر درایه از ماتریس جواب یک Task یا Thread ایجاد شود. برای انجام این مورد از Task استفاده کرده‌اید یا Thread؟ دلیل انتخاب خود را توضیح دهید.

4) کلاسی به نام ConcurrentIntegerList بنویسید که list از اعداد را نگهداری کرده و Thread Safe باشد. به این معنی که اگر به صورت همزمان چندین دسترسی به لیست از نوع خواندن و نوشتن وجود داشته باشد، مشکلی برای داده‌های ذخیره شده به وجود نیاید. استفاده از Concurrent Collection ها (System.Collections.Concurrent) مجاز نیست. در مورد مفاهیم انحصار متقابل (Mutual Exclusion) و Critical Section توضیح دهید.