## Lukas Lipinski Coding Challenge

Thank you for the challenge.   
During my work on this task, I have focused mostly on showing different aspects of creating testable, multithreading projects, thus it might be overengineered in some places.   
  
There are three projects in my solution:

* **PowerReport**

Here are all classes necessary for generating report. Main class is **InterDayReportGenerator** where is implemented process of creating report. In Threading and Extensions directories there are classes supporting multithreading, and testability, based on solutions I have used in the other projects. Class **InterDayReportReportDateProvider** provides dates required for report name and during calling to PowerService.

* **PowerReportService**

This is the start project, and compiles to executable. Application can be start directly from Visual Studio, or by installing Windows Service and running from command line or Services Management.

All available command-line options can be seen after running “PowerReportService.exe help”. In this project, there is also definition of **SettingsService** and **SchedulerService** responsible for setting up and running report generation.

* **PowerReportTest**  
    
  Test class for **PowerReport** project. In the couple of tests asynchrony is eliminated by using Immediate Scheduler.

**Source Code:**

**Solution is available on GitHub:**

<https://github.com/scrmorph/CC_Petro>

**It’s a public repository.**

**Any comments are welcome!  
  
Installation:  
“PowerReportService.exe help” – all available command-line options.   
“PowerReportService.exe install” – installs service (should be seen in services management console).  
  
  
External packages:**

Topshelf - <http://topshelf-project.com/>

CsvHelper - <https://joshclose.github.io/CsvHelper/>

NLog - <http://nlog-project.org/>

Nunit - <http://nunit.org/>

Fluent Assertions - <http://fluentassertions.com/>