Welcome to C#

Introduction

During this task, you will start to build a fundament for a project, on which you will work your entire semester. In the end of the semester, you will get a completed project for your portfolio. For now, let's focus on creation of it;)

Task

You got the task for creating a managing program of electronic devices. All devices have next common information:

- Id
- Name
- Is device turned on

All these devices can be turned on or turned off.

Table for device type presentation in file	
Manufacture type	Line in file starts with
Smartwatch	SW
Personal computer	P
Embedded device	ED

Smartwatches (SW)

Smartwatches contain information about battery percentage. Implement interface ***IPowerNotifier*** and call implemented function in case if value for setting battery percentage is less 20%.

Don't forget to handle situations when user attempt to set battery percentage less than zero or greater than one hundred.

If user tries to turn on device with lest than 11% in battery, throw ***EmptyBatteryException***. If operation turning on is successful, reduce battery percentage by 10.

Personal Computer (P)

All personal computers can have operation systems. However, it doesn't mean that it's installed by default. If operation system is not specified, it can't be launched. If user tries to launch computer, ***EmptySystemException*** should be thrown.

Embedded devices (ED)

All embedded devices have IP address and network name. On IP address set, check with Regex. If it has wrong format - throw ***ArgumentException***.
Implement function "Connect", where will be checked if network name value contains "MD Ltd.". If it's not, ***ConnectionException*** should be thrown. Function "Connect" should be used in "TurnOn" function.

Device manager

You must create class which will store all devices. In constructor of this class, you must accept parameter "filePath". From the specified path, read lines of file with built-in static class "File". You must check if file exists. After reading all files, parse all recorded devices to list of devices. There's a chance of corrupted lines, so do checks before creating new instance of class. If there's a wrong line – ignore it. Remember, that storage for devices is finite (max count is fifteen). You must check capacity before adding device to storage.
Also, class should provide next functions:

- Add device
- Remove device
- Edit device data

    - For this functionality you'll need technique called "boxing/unboxing". For more information, look at additional materials on Teams.
- Turn on device
- Turn off device
- Show all devices
- Save data to file

All exceptions should be caught, and their messages should be displayed to user.

Unit tests

To test functionality, implement unit tests, that will show correct work of mentioned functions above in device manager class.

Additional information

All child classes should override "ToString" function, where you should return detailed information (based on state of device).
It's prohibited to use any functionality that was not discussed on tutorial or wasn't mentioned here.
Also, it would be nice if you track you progress with commits in Git. Of course, I won't reduce points if you will have one-two commits, but it's better to see the progress (and possibility to revert changes for you;)).
Use .gitignore file to remove .idea folder or other unnecessary folders. You can generate file on web resources or generate with dotnet command.
Please, use .net8.0 as a target framework.