SCALE FOR PROJECT PISCINE CPP (/PROJECTS/PISCINE-CPP) / RUSH01 (/PROJECTS/PISCINE-CPP-RUSH01)

Comments

This rush is the final activity of the Piscine C++, and as such, the subject leaves a lot of room for creativity.
GKrellM is a small system monitor I like to use that represents each monitored data as a module. Of course users are encouraged to write additional modules to extend the functionalites of the monitor via a plugin system. A plugin system can be easily represented by a C++ interface and is a very relevant real life exemple.

Guidelines

The Makefile must compile with clang++, with -Wall -Wextra -Werror, or you must flag the project as "Invalid compilation" .

Any of these means you must flag the projet as "Incomplete work"
- A function is implemented in a header (except in a template)
- A class is not in Coplien's form

Any of these means that you must flag the project as Cheat:
- Use of a "C" function (*alloc, *printf, free)
- Use of a function not allowed in the subject
- Use of "using namespace" or "friend" (Unless explictly allowed in the subject)
- Use of C++11 features (Unless explictly allowed in the subject)

Also remember that this is a C++ project. Therefore, please take some time to look at the actual code of the project, and if it is just "C+", meaning that it's essentially a C program with a few classes "just for show" and std::cout instead of printf (you know what I mean), just mark the project as "Incomplete work" and be done with it. As the subject says, to be relevant in this project, you have to actually make an effort to use the object-oriented properties of the language.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/2086/rush01.en.pdf)

The IMonitorModule and IMonitorDisplay interfaces

Check that the code declares and uses the two mendatory interfaces IMonitorModule and IMonitorDisplay. - IMonitorDisplay is an abstraction between the two available displays. Thus, at least two class should implement this interface : the class that handles the shell UI and the one that handles the graphical UI. - IMonitorModule is an abstraction to the available modules. Implementing this interface allows a class to behave like a module of the monitor, and allows the monitor to handle every modules in an unified way. This, students. This is Object Oriented Programming.

The IMonitorDisplay interface

Is this interface present and used as intended ?

☑ Yes ☒ No

The IMonitorModule interface

Is this interface present and used as intended ?

☑ Yes ☒ No

Step 1

Date/time module

Is this module present and does it work as intended ?

☑ Yes ☒ No

OS infos module

Is this module present and does it work as intended ?

☑ Yes ☒ No

Hostname/username module

Is this module present and does it work as intended ?

☑ Yes ☒ No

Step 2

RAM module

Is this module present and does it work as intended ?

☑ Yes ☒ No

CPU module

Is this module present and does it work as intended ?

☑ Yes ☒ No

Step 3

Network throughput module

Is this module present and does it work as intended ?

☑ Yes ☒ No

Step 4 - Bonus

Please remember that bonus are available if and only if the 3 previous steps are perfect.

Bonus modules

Add one step on the slider for each additional module you deem relevant.

Regarding the module with zaz's favorite animal, no, it is not the pony or the unicorn, it is indeed the cat. (Yes, how very pedestrian)

Rate it from 0 (failed) through 5 (excellent)

Preview!!!

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok

Empty work | Incomplete work | No author file | Invalid compilation

Norme | Cheat | Crash | Forbidden function

Conclusion

Leave a comment on this evaluation