PROGRAMAÇÃO DE COMPUTADORES 2

TRABALHO AVALIATIVO – SOFTWARE DE MONITORAMENTO DE TEMPERATURA

Criar um software que monitore a temperatura a partir de um simulador. Esse simulador é executado pela DLL simuladorTermico32.dll, para desenvolvimento em 32bits, e simuladorTermico64.dll, para 64 bits. O acesso a esse simulador é feito através da classe temperatura, fornecido com o trabalho. A classe temperatura possui 3 métodos, o principal deles é o método enviarComando, que permite enviar um comando a uma ventoinha, um resistor e uma lâmpada.

```
float enviarComando(int n); //Recebe um codigo numerico conforme a tabela abaixo e retorna um codigo de confirmacao
```

Código	Ação	Resposta
1	Leia temperatura	Retorna a temperatura
2	Aciona Ventoinha	200+n em caso de sucesso
3	Desliga Ventoinha	
4	Aciona Resistor	
5	Desliga Resistor	
6	Liga Lâmpada	
7	Desliga lâmpada	

Tabela 1 : Ações disponíveis para controle do simulador térmico.

Além do método ***enviaComando***, há também os métodos ***inicializa()*** que deve ser chamado para iniciar a simulação e ***double lerTemp()***, que executa a leitura da temperatura corrente.

A biblioteca ***Temperatura.h*** tem seus códigos em ***Temperatura.cpp*** que também será fornecida juntamente com a DLL acima citada, assim basta incluir a **temperatura.h** para utilizar o método acima. Um projeto inicial completo para o Eclipse CDT está disponível na página da disciplina no Moodle, assim como alternativas para *vscode* e um projeto portável com o Dev-Cpp Portable.

O software deve ler a temperatura, e armazená-la em uma lista estática de 40 mil entradas disponíveis, juntamente com o estado do resistor e da ventoinha. Um arquivo de log deve ser salvo com a respectiva data e hora de salvamento no nome, com o seguinte no formato (AAAMMDD_hhmmss). Informações de como obter data e hora do sistema neste ***link***.

O software deve prover as seguintes funcionalidades:

1 - Monitoramento:

Permitir a definição de número de leituras e intervalo entre elas em menu listando a temperatura, estado da ventoinha e resistor, ocorridos durante o processo de leitura, após a leitura deve voltar ao menu. Deverá ser possível também limpar os dados da última leitura.

2 - Salvar e Recuperar Dados:

Deverá ser possível armazenar as leituras em arquivos no formato *CSV*, com todos os dados da lista atual. Já quando o software for aberto deve-se poder buscar os dados da execução anterior informando o nome do arquivo, realimentando a lista em memória com todo o histórico do arquivo lido.

3 - Controle de temperatura:

Um limiar de temperatura máxima e mínima, configuráveis via menu. Que caso a temperatura máxima seja atingida ligue a ventoinha, caso a temperatura mínima seja atingida deve-se ligar o resistor.

4 - Análise estatística da leitura atual:

A temperatura máxima e mínima, média e mediana dos dados de leitura atual. Além da quantidade de temperatura acima dos limites máximos e mínimos definidos, considerando os limiares do momento, quando definidos.

O menu principal deverá apresentar as quatro principais funcionalidades e um submenu para aos parâmetros necessários em cada funcionalidade, sendo sempre possível voltar ao menu principal. É necessário validar toda a informação de entrada, solicitando nova entrada caso o usuário informe dados inválidos, tanto em relação as opções de menu como entradas de parâmetros, como por exemplo entrada de textos em campos numéricos.

Deverão ser modeladas as classes necessárias que provenham uma eficiente modularização e que evitem ao máximo duplicação de código, no mínimo de 3 classes desenvolvidas pelo aluno. É proibido o uso de variáveis globais (as variáveis devem ser passadas as classes ou funções por parâmetro ou referência), também proibido o uso de classe STL ou derivadas destas no trabalho.

A entrega deverá ser feita via Moodle, enviando um arquivo com o link do repositório do gitlab até o dia **20/09 (23:59)**, versões submetidas ao repositório após esta data não serão avaliadas. Após a entrega deverá ser agendada a apresentação do trabalho, que ocorrerá fora do horário de aula da disciplina até o dia **23/09**. A apresentação consiste da explicação do código seguida de respostas aos questionamentos que o professor julgar necessário sobre o desenvolvimento do mesmo.