



UNIVERSIDADE FEDERAL DE SANTA CATARINA- UFSC
ENGENHARIA ELETRÔNICA

LUCAS ARMANDO CIELLO

Relatório preliminar projeto final

Florianópolis, 2018

LUCAS ARMANDO CIELLO

Relatório técnico apresentado como requisito parcial para obtenção de aprovação na disciplina Programação C++ para sistemas embarcados, no Curso de Engenharia Eletrônica, na Universidade Federal de Santa Catarina.

Prof. Dr. Eduardo Augusto Bezerra

Florianópolis, 2018

Sumário

Sumário	2
Resumo	3
Diagrama de Classes	4
Maquina de estado	4
Filas	4
Maquina de estado	5
Funcionamento da maquina de estado	5
Explicação Código	6
Fila de anúncios	6
Explicação Código	7
Funcionalidade adicional do sistema	7
Ferramentas utilizadas	8

Itens já avaliados

Maquina de estado pág. 5

Funcionamento da maquina de estado pág. 5

Explicação do código pág. 6

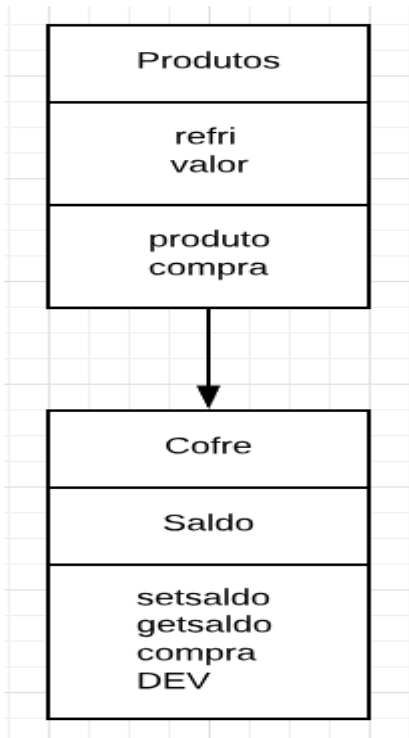
Os demais itens foram adicionados ou revisados e alterados.

Resumo

Este relatório apresenta o diagrama de classes preliminar revisado, o funcionamento da maquina de estado, as filas que serão utilizadas no projeto, bem como seus códigos em C++ e a descrição do funcionamento adicional da maquina de refrigerante.

Diagrama de Classes

Maquina de estado



O Diagrama é referente a maquina de estados, onde temos as classes produtos onde são armazenados os dados dos refrigerantes, bem como a classe cofre que é responsável pela gestão do dinheiro e da liberação do produto.

A classe cofre recebe a classe produto, tendo acesso aos seus atributos e métodos.

Filas

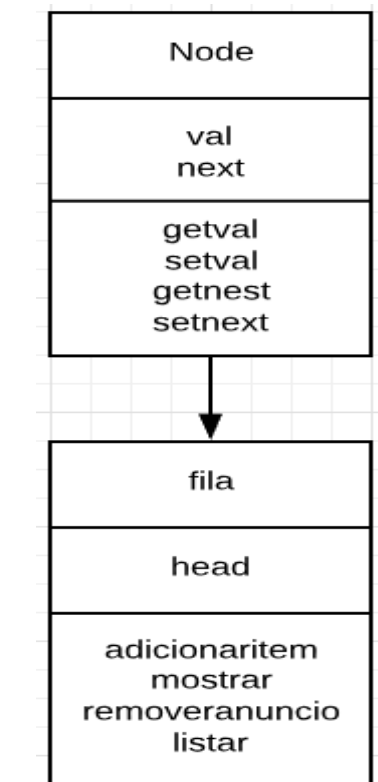


Diagrama referente às filas que serão utilizadas para implementação da dinâmica de propagandas.

A classe fila recebe a classe Node, tendo acesso aos seus atributos e métodos.

O programa final recebera as classes da Máquina de estado, da fila bem como as classes utilizadas para a funcionalidade extra do projeto, possibilitando a implementação de um programa que fara toda a gestão da maquina de refrigerante e em paralelo serão apresentadas as propagandas cadastradas pelo utilizador e a funcionalidade extra.

Maquina de estado

A máquina fornece dois tipos de refrigerantes, MEET e ETIRPS, que estão disponível para ser escolhidos no menu do programa. Ambos refrigerantes custam R\$1,50 e a máquina reconhece moedas de R\$1,00, R\$0,50 e R\$0,25, no código o usuário indica a moeda que está colocando, visto que nesta etapa não será utilizado nenhum tipo de sensor ou botão. A maquina devolve automaticamente o troco quando o saldo da maquina passa de R\$1,50, além disso, durante a compra, o usuário pode desistir da transação, a opção DEV devolve as moedas inseridas até o momento. Somente após acumular um crédito de R\$1,50 o usuário pode obter um refrigerante.

Funcionamento da maquina de estado

Estado atual	Comando de Entrada						
	Nada	M025	M050	M100	DEV	MEET	ETIRPS
S000	S000	S025	S050	S100	S000	S000	S000
S025	S025	S050	S075	S125	S000, D025	S025	S025
S050	S050	S075	S100	S150	S000, D050	S050	S050
S075	S075	S100	S125	S150, D025	S000, D075	S075	S075
S100	S100	S125	S150	S150, D050	S000, D100	S100	S100
S125	S125	S150	S150, D025	S150, D075	S000, D125	S125	S125
S150	S150	S150, D025	S150, D050	S150, D100	S000, D150	S000, DMEET	S000, DETIRPS

Através dessa tabela podemos verificar todas as possibilidades e o comportamento da maquina de estado.

Explicação Código

O código completo da maquina de estado está como anexo, bem como no github (<https://github.com/lucasciello/Projeto-final>).

A classe produto é utilizada para armazenar os dados dos refrigerantes, a classe cofre é filha da classe produto, tendo acesso a seus atributos e métodos. A classe cofre é responsável por cuidar do saldo e da liberação do refrigerante, Ambas as classes têm seus construtores que colocam o programa em estado inicial. A função setsaldo é responsável por adicionar o saldo sempre que o usuário informar que uma moeda foi inserida na maquina, adicionando o saldo anterior ao valor da moeda adicionada, sempre que o saldo ultrapassar o valor de R\$1,50, a função devolve o saldo excedente. A função compra faz um teste para verificar se o saldo é de R\$1,50, caso seja, ela libera o refrigerante escolhido e o valor do refrigerante é descontado do saldo. A função DEV devolve o saldo acumulado na maquina, caso ele exista. Este código é preliminar, podendo ser alterando conforme a necessidade ao longo do projeto.

Fila de anúncios

O sistema é composta por duas filas, a fila 2 é utilizada para armazenar os anúncios, enquanto a fila 1 é utilizada para mostrar os anúncios. Sempre que um anuncio novo e adicionado, ele é adicionado na fila 2, quando o usuário quiser apresentar esta anuncio ele carregar os anúncios incluídos na fila 2 na fila 1, então a fila 1 apresenta os anúncios atualizados. Sempre que um anuncio é apresentado ele é removido da fila e adicionado no fim da fila, no caso onde os anúncios são carregados de uma fila para outra, o anuncio é adicionado no fim da fila 1 e removido da fila 2.

Explicação Código

O código completo das filas está como anexo, bem como no github (<https://github.com/lucasciello/Projeto-final>).

A classe nodo é utilizada para armazenar os anúncios e um ponteiro que aponta para o próximo nodo, a classe tem seu construtor que é utilizado para armazenar os valores nas variáveis, bem como as funções getVal e getNext que são utilizados para apresentar o anuncio do nodo em questão e o próximo nodo respectivamente, as funções setVal e setNext são utilizadas para definir qual será o anuncio no nodo e qual será o próximo nodo.

A classe fila recebe a classe Node, tendo acesso aos seus atributos e métodos, possui seu construtor e destrutor, bem como as funções, adicionaritem que é responsável por adicionar um novo anuncio, a função mostrar que retorna o anuncio, a função removeranuncio que remove o anuncio da fila e a função listar, que lista todos os anúncios adicionados na fila, essa função não será utilizada no código, pois foge dos conceitos de fila, ela foi utilizada para testes no programa.

Na main é apresentado um menu para o operador poder adicionar, carregar, remover anúncios. Para cada função e funcionalidade do sistema será implementada uma outra função que possibilita o código ser utilizado na plataforma Raspberry PI, utilizando o conceito de polimorfismo para alterar a plataforma a ser utilizada. Os anúncios serão apresentados em um display no Raspberry PI ou na própria tela. O código apresentado é uma versão inicial e será alterado para atender os requisitos do sistema desejado.

Funcionalidade adicional do sistema

Sempre que uma compra for realizada o cliente poderá receber um canudo de cortesia, existira um botão do canudo que ficara disponível sempre que uma compra for realizada, quando acionado a maquina liberara um canudo para o cliente. Caso o cliente não queira o canudo a maquina não liberara canudo. Caso a maquina fique sem canudos um led ascendera indicando a situação.

Para implementação dessa funcionalidade será utilizado o conceito de pilha, os canudos serão acionados na maquina e serão removidos em ordem.

Ferramentas utilizadas

Para implementação do projeto serão utilizados os conceitos vistos em aula, herança, friends, template, funções virtuais, sobrecarga de operador, polimorfismo, para fazer com que as filas possam ser utilizadas em paralelo, uma apresentando os anúncios e outra adicionando novos anúncios, será utilizado o conceito de threads, dentre outros métodos que serão necessários para o funcionamento do projeto e a sua implementação na plataforma Raspberry PI.