

Protótipo de Editor Gráfico de
Fluxogramas para
Representação de Comandos
da Linguagem Portugol

Aluno

Luís Carlos Seifert de Souza

Orientador

Wilson Pedro Carli

Roteiro da Apresentação

- Introdução
 - Motivação
 - Área
 - Objetivo
- Conceitos Básicos
 - Portugal
 - Diagrama de Fluxo

Roteiro da Apresentação

- Contexto Atual do Tema
 - Ferramenta Case Designer/2000
 - Ambiente Visual para a Edição de Circuitos Lógicos
 - Ambiente Delphi
- Especificação
 - Orientação a Objetos
 - Simbologia

Roteiro da Apresentação

- Especificação
 - Diagrama de Relacionamento de Objetos
 - Objetos Utilizados
- Implementação do Protótipo
- Conclusão
- Sugestões

Introdução

O aprendizado de lógica de programação é geralmente obtido através da resolução de problemas utilizando-se da linguagem Portugol ou através da linguagem gráfica Diagrama de Fluxo para expressar o raciocínio.

Desta forma, este trabalho tem por finalidade o entendimento da sintaxe e comandos da linguagem Portugol e Diagrama de fluxo, além do desenvolvimento de um protótipo que permite editar um fluxograma e gerar a partir deste a representação do código através da linguagem Portugol.

Motivação

Possibilidade de auxiliar alunos no
aprendizado de lógica de programação

Objetivo

Especificação e Implementação de um Editor Gráfico de Fluxogramas para a representação na Linguagem Portugol, criando um facilitador para o ensino de disciplinas que ministram a introdução a lógica de programação.

Conceitos Básicos

Portugol

Segundo Guimarães/Lages, Portugol é “uma pseudolinguagem de programação (simbiose de Português com ALGOL e PASCAL).”

Diagrama de Fluxo

Segundo [GUS77] é Linguagem Gráfica que recebeu o nome de Diagrama de Fluxo ou Fluxograma, utilizada para representar graficamente as diversas ações que o computador executará na resolução de um problema.

Contexto Atual do Tema

Ferramenta CASE Designer/2000

Conceito de Geração de Código Fonte

Ambiente Visual para a Edição de Circuitos Lógicos

- Layout do Editor
- Funcionamento

Ambiente Delphi 4

Desenvolvimento do Protótipo

Especificação

Orientação a Objetos

Segundo Scott W. Ambler o conceito de orientação a objetos deve-se a definição dos seguintes conceitos:

- Objetos, Classes, Atributos, Métodos, Abstração, Encapsulamento, Ocultamento de Informações, Herança, Persistência, Relacionamento entre Informações, Agregação, Colaboração, Acoplamento, Coesão, Polimorfismo

Orientação a Objetos

- **Objetos** - Qualquer indivíduo, lugar, coisa, evento, tela, relatório ou conceito que seja aplicável ao projeto do sistema. Os objetos possuem dados e funcionalidade que definem seus comportamentos.
- **Classe** - Uma categoria de objetos semelhantes. Uma Classe é na verdade uma planta baixa a partir do qual objetos são Criados

Orientação a Objetos

- **Instância** - Outra palavra para objeto. Um objeto é uma instância de uma classe.
- **Atributos** - Algo que um objeto ou uma classe sabem. Um atributo é basicamente um único dado ou informação.
- **Métodos** - Algo que um objeto ou uma classe faz. Um método é semelhante a uma função ou procedimento em uma programação estruturada.

Orientação a Objetos

- **Interface** - O grupo de mensagens que o objeto ou classe vai responder.
- **Abstração** - A definição da interface de uma classe (o que ela sabe e faz).
- **Encapsulamento** - O ocultamento da implementação do que uma classe/objeto sabe ou faz.
- **Ocultamento de Informações** - A restrição ao acesso dos atributos.

Orientação a Objetos

- **Herança** - Permite tirar vantagem das similaridades entre as classes.
- **Persistência** - Diz respeito ao armazenamento permanente dos objetos em disco.
- **Relacionamentos entre Instâncias** - Relacionamentos ou associações entre objetos.

Orientação a Objetos

- **Cardinalidade** - Indica quantos objetos estão envolvidos em um relacionamento.
- **Agregação** - Representa um relacionamento do tipo “faz parte de”.
- **Colaboração** - Classes/ objetos trabalham em conjunto (colaboram para desempenhar suas funções).
- **Mensagem** - Requisição de informações ou um pedido para execução de uma ação.

Orientação a Objetos

- **Acoplamento** - Medida de quanto duas classes se encontram conectadas.
- **Coesão** - Medida de quanto um método ou uma classe fazem sentido.
- **Polimorfismo** - Diz respeito que um objeto pode tomar qualquer de várias formas e que outros objetos podem interagir com este objeto sem que saibam qual forma específica ele toma.

Simbologia

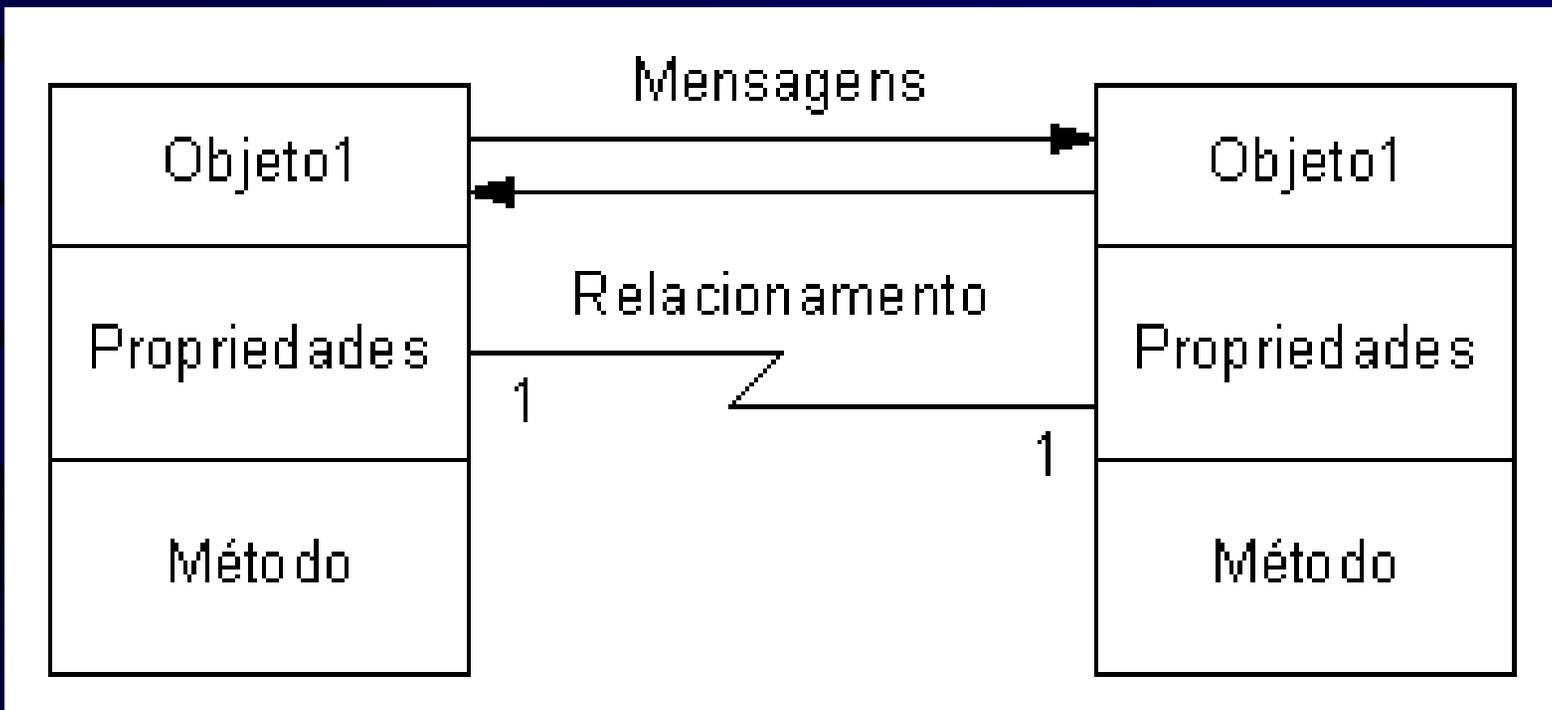


Diagrama de Relacionamento dos Objetos



Objetos Utilizados

Objeto Diagrama

Objeto base da implementação, pois faz a união de todos os componentes utilizados no protótipo.

Diagrama
Criar, Abrir, Salvar, Gerar Código Portugol, Verificar Diagrama.

Objeto Lista

Responsável por armazenar os componentes do diagrama, sendo utilizado principalmente para percorrer todos os componentes do diagrama.



Objeto Componente

Objeto que representa as ações que os símbolos do diagrama representam.

Componente

Nome,
Topo Início,
Esquerda Início,
Topo Fim,
Esquerda Fim,
Tipo,
Ação,
Movimento,
Ligação Entrada 1,
Ligação Entrada 2,
Ligação Saída 1,
Ligação Saída 2

Desenhar, Remover
Criar

Objeto Conexões

Objeto que faz a conexão entre os componentes do diagrama, dando sentido lógico e seqüencial as ações que os componentes representam.

Conexões
Ponto Inicial, Ponto Final
Limpar, Desenhar, Remover

Implementação do Protótipo

Conclusão

O tema abordado neste trabalho é muito interessante, sendo que no desenvolvimento deste, tive alguma dificuldade em encontrar assuntos específicos e atualizados sobre o tema Diagrama de Fluxo e Portugol, apesar disso, a maior dificuldade encontrada está na implementação, pois a pouca experiência em programação dificultou em muito a realização deste protótipo.

Sugestões

- Mostrar o fluxo da execução do fluxograma, através da mudança de cor dos componentes editados, além de um melhoramento na edição do fluxograma;
- Gerar o Fluxograma a partir da edição de um algoritmo (Linguagem Portugal).
- Gerar a impressão do código português e do fluxograma editado.

FIM