

Spyke - Ferramenta de Apoio ao Ensino de Pilhas e Filas

Priscyla Waleska Targino de Azevedo Simões¹, Luciano Dagostin Baldessar², Paulo João Martins³, Ana Cláudia Garcia Barbosa⁴

^{1,2,3,4} Curso de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC – Brasil

¹ Grupo de Pesquisa em Inteligência Computacional Aplicada, Grupo de Pesquisa em Informática Médica e Telemedicina, Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC – Brasil

¹ Curso de Medicina – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma – SC – Brasil

pri@unesc.net, lbaldessar@hotmail.com, {pjm, agb}@unesc.net

Abstract. *This paper describes the development to it of the SPYKE – Shell come back to education of stacks and queues, that it aims at to assist the process of teach-learning of these structures, in them you discipline of Algorithms and Data Structures, the academics of the courses of graduation of the area of Computer science. Its objective is to demonstrate, of interactive and graphical form, the functioning of the primitive operations of stacks and queues, by means of the static and dynamic memory allocation in the programming languages C++, Pascal and Java, being possible also, to analyze these algorithms in these programming languages, and to compare the code source of these algorithms. For the development of the SPYKE the environment Borland Delphi 6,0 for the development of the graphical interface, and the shell Help Studio 2000 for the development of the aid module was used.*

Resumo. *O presente artigo refere-se ao desenvolvimento do SPYKE – Ferramenta voltada ao ensino de pilhas e filas, que visa auxiliar o processo de ensino-aprendizagem dessas estruturas, nas disciplinas de Algoritmos e Estrutura de Dados, a acadêmicos dos cursos de graduação da área de Informática. Seu objetivo é demonstrar, de forma interativa e gráfica, o funcionamento das operações primitivas de pilhas e filas, por meio da alocação estática e dinâmica de memória, nas linguagens de programação C++, Pascal e Java, sendo possível também, analisar esses algoritmos nessas linguagens de programação, e comparar o código fonte desses algoritmos. Para o desenvolvimento do SPYKE foi utilizado o ambiente Borland Delphi 6.0 para o desenvolvimento da interface gráfica, e a ferramenta Help Studio 2000 para o desenvolvimento do módulo de ajuda.*

1. Introdução

O computador processa essencialmente algoritmos escritos em uma linguagem de programação. Para a criação de programas eficientes, não basta dominar os comandos de uma linguagem de programação, é necessário conhecer um pouco mais sobre a construção das estruturas de dados.

Considerando o processo de ensino-aprendizagem, é pressuposto que não se pode aprender a programar apenas lendo um livro, sendo a programação uma habilidade que deve ser desenvolvida com muita prática e observação, pois até mesmo os melhores programadores estudam outros trabalhos, e acabam incorporando essas observações ao seu dia-a-dia [Tajra 1998].

Nesse aspecto, o propósito deste artigo é descrever o desenvolvimento de um tutorial voltado ao ensino de pilhas e filas, o qual, via interface com o usuário, demonstra para essas estruturas, as formas de alocação de memória estática e dinâmica, e evolução gráfica durante a execução das funções primitivas, nas linguagens de programação Pascal, C++ e Java.

A maior parte dos cursos de graduação na área de computação aborda o processo de ensino-aprendizagem de pilhas e filas, pelo foco do Tipo Abstrato de Dados (TAD), que pode ser definido como a maneira pela qual as partes de um todo estão dispostas entre si, evidenciando-se o que faz e ignorando como faz. [Moraes 2001].

Para o estudo de pilhas e filas, o aluno deve possuir uma boa experiência em uma linguagem de programação, sendo difícil o entendimento dos TADs, devido à dificuldade em imaginar essas estruturas e entender seu funcionamento, pela complexidade e características exclusivas a cada TAD, e também, às variadas formas de se inserir ou remover dados nessas estruturas, como por exemplo: nas pilhas as inserções e remoções são realizadas na mesma extremidade, já nas filas as inserções são realizadas no final e as remoções no seu início.

Essas operações acontecem também, sem que se possa verificar e visualizar sua evolução, e por isso, é de grande contribuição acadêmica o desenvolvimento de um ambiente gráfico que possibilite ao usuário a demonstração de como são realizadas as operações básicas de cada TAD (inserções, remoções, consultas, entre outras) a fim de que o estudante possa visualizá-las na linguagem de programação utilizada para o seu estudo [Cormen 2003].

2. Metodologia

A construção do Spyke procurou possibilitar ao usuário um ambiente rico em informações sobre os diferentes métodos de se implementar pilhas e filas, possuir uma interface de fácil navegabilidade, possibilitando ao usuário a construção do conhecimento nessa área de estudo.

A base metodológica utilizada para o desenvolvimento dessa pesquisa é composta pelo estudo de Pilhas e Filas, da modelagem do sistema, projeto de Interface, e elaboração do Módulo de Ajuda..

2.1. Fluxo de Informações do Sistema

O Spyke possibilita ao usuário final (estudantes de Estrutura de Dados de um curso de Ciência da Computação), um ambiente rico em informações a respeito de pilhas e filas, de fácil navegabilidade e com o mínimo de obstáculos que possam impedir a construção do conhecimento em seu processo de aprendizagem.

Por meio da manipulação de componentes visuais da biblioteca VCL do ambiente de programação Delphi 6, utilizado na construção dessa ferramenta, foi possível agrupar informações, obtendo-se diversas telas com a criação de vários formulários, que possibilitam visualizações gráficas de Pilhas e Filas estáticas e dinâmicas, nas linguagens C++, Pascal e Java, e possibilitam ao usuário interação na criação, inserção, remoção e consulta de elementos, operações essas efetuadas graficamente, que oferecem uma melhor compreensão desses TADs.

O fluxo de informações do SPYKE (figura 1) começa a partir de uma tela principal, dividida em quatro outras telas, que demonstram graficamente os seguintes TADs: Pilha Estática, Pilha Dinâmica, Fila Estática e Fila Dinâmica, sendo cada estrutura composta por mais três opções, relacionadas a demonstração gráfica dos TADs nas linguagens C++, Pascal e Java.

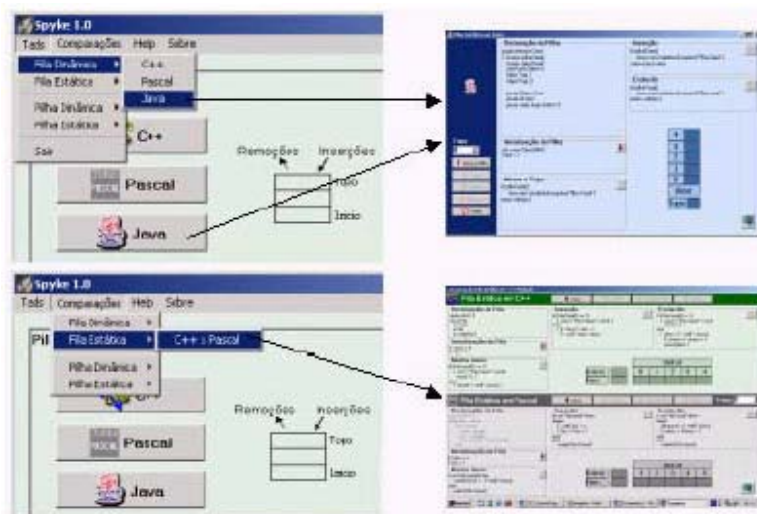


Figura 1. Fluxo de informação do SPYKE.

O SPYKE oferece também a possibilidade da realização de comparações, onde o TAD é comparado entre diferentes linguagens de programação.

O sistema é composto por uma tela principal que é dividida em quatro partes, as quais agregam um resumo de todas as informações do sistema, que são: os quadros pilha estática, pilha dinâmica, fila estática e fila dinâmica. Cada um desses quadros, possui três botões, C++, Pascal e Java, pelos quais o usuário tem as opções de visualizar as estruturas nessas três linguagens de programação.

Por exemplo, na Figura 2, que apresenta a tela pilha estática em C++, são exibidos vários quadros, sendo cada um referente a uma função primitiva da pilha, indicando do lado direito, a sua representação gráfica. Os botões presentes na tela do lado esquerdo referem-se à seqüência de ações que o usuário pode realizar, ou seja, cada uma das operações primitivas – Inicialização da Pilha, Mostrar Topo, Exclusão, e Inserção.

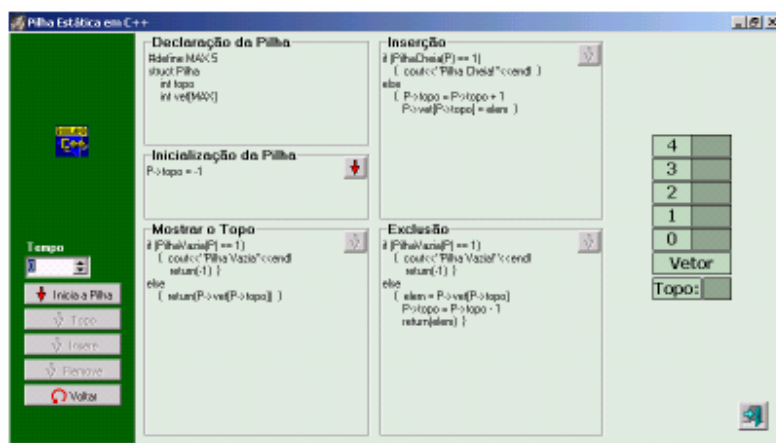


Figura 2. Pilha estática em C++.

3. Conclusão

As instituições de ensino estão mais conscientes do importante papel que o computador tem no aumento gradual do desempenho no aprendizado de seus alunos, portanto, nesse contexto, é de fundamental importância o desenvolvimento de ferramentas alternativas que auxiliem o aluno no processo de ensino-aprendizagem.

O desenvolvimento dessa ferramenta se propõe a proporcionar um recurso de auxílio no processo de ensino-aprendizagem de pilhas e filas nas disciplinas de Algoritmos, Estrutura de Dados, e afins, a acadêmicos dos cursos relacionados à área de informática.

Concluindo, com o desenvolvimento do SPYKE possibilitou-se a demonstração gráfica do funcionamento das funções primitivas de pilhas e filas, nas linguagens C++, Pascal e Java, sendo possível comparar esses algoritmos em diferentes ambientes de programação.

Referências

- Moraes, C. R. (2001), “Estrutura de dados e algoritmos: uma abordagem didática”. São Paulo: Berkeley Brasil.
- Cormen, T. H; SOUZA, V. D. (2002), “Algoritmos: teoria e prática”. Rio de Janeiro: Ed. Campus.
- Tajra, S. F. (1998), “Informática na Educação: Professor na Atualidade”. São Paulo: Érica.