

Novemax

Biblioteca OO para abstração de componentes HTML, desenvolvida em PHP

Isac Velezo de C. Aguiar
isacaguiar@gmail.com
Jailson Oliveira da Boa Morte
jailsonboamorte@gmail.com

Instituto de Matemática
Departamento de Ciência da Computação
Universidade Federal da Bahia

***Resumo.** O trabalho proposto tem como objetivo desenvolver na linguagem PHP uma biblioteca para componentes HTML, denominada Novemax. O motivo para o desenvolvimento do trabalho decorre dos fatores relacionados a organização do código HTML tendo em vista que os componentes HTML tornam o código de certa forma complexo, devido a geração de uma grande quantidade de linhas de código para implementar os componentes. Com o desenvolvimento desta biblioteca será possível construir aplicações de forma homogênea entre as linguagens abordadas.*

1. Introdução

A linguagem PHP (PHP: Hypertext Preprocessor) é popular e bastante utilizada no mundo www (World Wide Web), afirma o PHP Group. Sendo uma linguagem open-source (de código aberto) que tem como principal objetivo gerar conteúdo dinâmico para páginas da internet. Ela está em constante crescimento, o que está possibilitando a criação de bibliotecas que visam auxiliar os programadores no processo de desenvolvimento.

Ela é uma linguagem interpretada, devido a isto não há necessidade de criar um programa para gerar e imprimir HTML (HyperText Markup Language), pode-se escrever HTML com código PHP embutido com o intuito de gerar conteúdo dinâmico (Melo, Alexandre de Altair e Nascimento, Mauricio G. F.). Entretanto o mau uso na integração destas tecnologias pode tornar o código fonte confuso e de difícil entendimento, o que não seria considerado uma boa prática de programação.

Levando-se em consideração que os recursos da OO (Orientação à Objeto), disponibilizado a partir da versão 4.0 da linguagem PHP, podem solucionar estes problemas e facilitar a integração destas tecnologias. Porém para isto é necessário

fazer uso de recursos que possibilitem a construção de componentes HTML de forma simples, rápida, eficiente, evite o retrabalho, e a perda de tempo na construção destes componentes, fator este que pode impactar diretamente no projeto, pois o programador poderá perder tempo saindo do foco do negócio.

Neste artigo será apresentada uma biblioteca de classe denominada Novemax, que tem como objetivo principal fornecer aos desenvolvedores de sistema recursos que tendam a atender as necessidades especificadas acima.

A seguir serão apresentados conceitos sobre as linguagens PHP e HTML, será feita uma abordagem o orientação à objeto do PHP, como também sobre a biblioteca proposta neste caso a Novemax e pra finalizar será feita a conclusão obtida durante o processo de desenvolvimento do trabalho.

2. Revisão bibliográfica

Nesta seção serão abordados conceitos sobre as linguagens PHP e HTML, princípios da Orientação a Objeto e a correlação da linguagem PHP e os recursos da OO.

2.1 Linguagem PHP

A linguagem PHP está em constante crescimento, uma das principais características que favorecem este crescimento é o fato dela ser open-source. Isto pode ser confirmado em pesquisas que são realizadas por órgãos competentes, com o Netcraft (pode ser acessado no seguinte endereço da Internet: <http://www.securityspace.com/sspace>).

Como também pode ser percebido no histórico da linguagem que apresenta um grande crescimento desde sua origem em 1994 e com apenas 4 anos de existência (sendo mais preciso no final de 1998) já estava presente em cerca de 10% dos domínios da Internet , segundo o PHP Group. Abaixo pode ser visualizada a imagem que demonstra em forma de gráfico este crescimento.

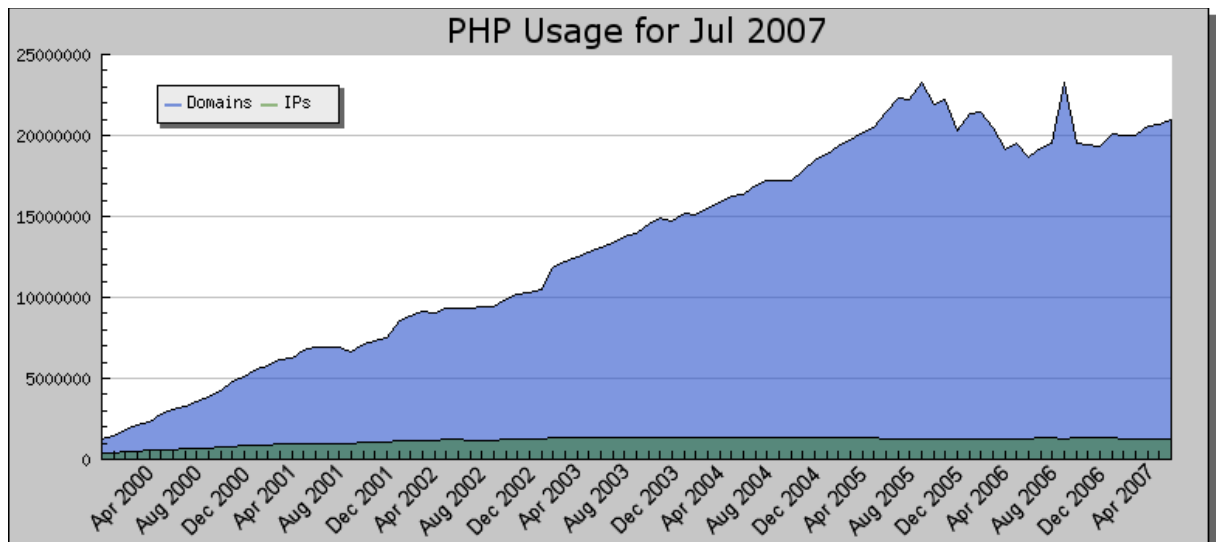


Figura 1 - Números do PHP no mundo.

Fonte Netcraft – Abril/2007.

Como foi descrito o principal objetivo desta linguagem é gerar conteúdo dinâmico para as páginas da Internet. Para isto ela possui uma característica muito importante que seria o fato dela permitir que seja escrito código HTML juntamente com código PHP. Porém nem sempre esta integração é realizada da forma mais adequada, ou seja, não é efetuado utilizando boas práticas de programação. Podemos citar como exemplo a possibilidade de carregar um componente HTML select dinamicamente, poderíamos fazer somente em php, escrevendo código HTML, ou em HTML embutindo código PHP quando necessários e aplicando as sugestões de diversas formas. Imagine um projeto em que você pretende carregar diversos select's em páginas distintas, isto poderia trazer problemas, que iriam exigir mudanças em diversas partes do código fonte do projeto.

2.2 Linguagem Html

HTML é um acrônimo para a expressão inglesa Hypertext Markup Language, que significa Linguagem de Marcação de Hipertexto, isto é, designa uma linguagem de descrição de documentos standard da World Wide Web. É uma aplicação da SGML que utiliza tags para definir os diferentes elementos, tais como texto, elementos multimídia, formulários, hiperligações, etc.

2.3 Orientação a Objetos

A orientação a objetos, também conhecida como Programação Orientada a Objetos (POO) ou ainda em inglês *Object-Oriented Programming (OOP)* é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamadas de objetos. O funcionamento deste sistema se dá através do relacionamento e troca de mensagens entre estes objetos. A base deste paradigma é a implementação de um conjunto de classes que definem os objetos presentes no sistema de software. Cada classe determina o comportamento (definidos nos métodos) e estados possíveis (atributos) de seus objetos, assim como o relacionamento com outros objetos. Através desse mecanismo, classes derivadas compartilham propriedades comuns de uma classe base, enquanto mantêm seus comportamentos específicos (Stroustrup, 1997). A seguir uma breve descrição dos principais conceitos de OO:

Classe – escreve um grupo de objetos com propriedades (atributos), comportamento (operações) similares, relacionamentos comuns com outros objetos e uma semântica comum.

Objeto – definido neste modelo como um conceito, abstração ou algo com limites e significados bem definidos para a aplicação em questão.

Atributo – um valor de dado assumido pelos objetos de uma classe.

Método – implementação de uma operação para uma classe, que podem ser aplicadas a ou por objetos em uma classe.

Herança – é uma poderosa abstração para compartilhar similaridades entre classes e ao mesmo tempo preservar suas diferenças. Cada classe derivada não apenas herda as todas as características de todos seus ancestrais como também pode acrescentar seus atributos operações específicos.

Polimorfismo – permite que referências de tipos de classes mais abstratas representem o comportamento das classes concretas que referenciam. Assim, um mesmo método pode apresentar várias formas, de acordo com seu contexto. O polimorfismo é importante pois permite que a semântica de uma interface seja efetivamente separada da implementação que a representa.

2.4 PHP orientado a Objeto

Levando-se em consideração o amplo crescimento, evolução constante da linguagem e principalmente a característica da OO adotada em sua versão 4.0, que permite a implementação de um conjunto de classes formado por código reutilizável, organizado e disponibilizado para simplificar o trabalho do programador, e que este código encontra-se geralmente em bibliotecas dinâmicas que a aplicação utiliza durante a sua execução. Surgiu então a idéia de criar uma biblioteca de classes para oferecer maior facilidade e menor complexidade de construir componentes HTML, podendo este possibilitar a utilização de recursos avançados para serem utilizados por parte dos desenvolvedores nos sistemas desenvolvidos em PHP.

Em junho de 2004 foi lançada a versão 5 do PHP, introduzindo um novo modelo de orientação a objeto, como por exemplo:

- Reformulação dos Construtores e adição de Destrutores
- Visibilidade de acesso
- Abstração de objeto
- Interfaces de objetos

O tratamento de objetos do PHP foi completamente reescrito, permitindo um desempenho melhor e mais vantagens. Enquanto na versão anterior era preciso muito esforço para atender à orientação a objetos e aos padrões de projetos (alguns não eram possíveis), o PHP 5 veio para sanar essa deficiência.

2.5 Novemax - Implementação e Resultados

O Novemax é uma biblioteca de classes desenvolvida em php utilizando fundamentos da Orientação a Objeto. Neste capítulo serão apresentados diagramas de classe utilizando representação UML (Unified Modeling Language) das classes que compõe a biblioteca, destacando-se os conceitos da OO e os recursos da linguagem PHP.

A biblioteca tem como principal objetivo permitir a construção de componentes HTML básicos e avançados de forma simples. Ela possui a seguinte estrutura de pacotes abaixo.

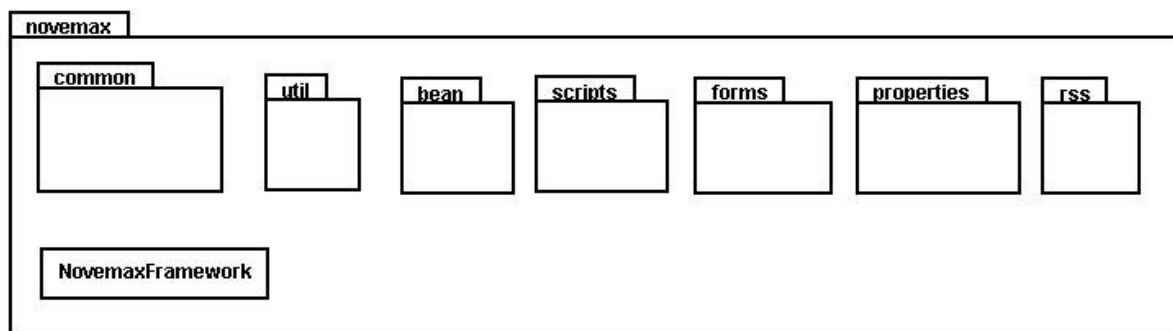


Figura 2 – Estrutura física da biblioteca

Existe um pacote principal, denominado Novemax onde está disponível o código fonte da biblioteca, nele destaca-se o arquivo (do tipo inc.php) na pasta principal, denominado NovemaxFramework (na ilustração sendo representada como uma classe). Ele é o principal arquivo da biblioteca, pois para utilização é necessário fazer a referência apenas para ele, sendo possível desta forma fazer o uso de todas as classes existentes na biblioteca.

E nos pacotes encontram-se as classes que foram definidas utilizando conceitos da Orientação a Objeto, a seguir estes pacotes serão listados e descritos utilizando a diagramação UML.

Common: este pacote é composto de classes que constrói elementos comuns quando em relação a componentes HTML, inicialmente serão disponíveis para construção dos seguintes componentes: links, tabelas, e-mail, imagens.

Util: pacote composto inicialmente de apenas uma classe responsável para criação de paginação de listagem, permitindo a navegação nos registros existentes em outras páginas.

Bean: pacote composto por uma classe que faz referência a uma entidade no caso para composição de elementos que contemplam o Combobox (valor e descrição).

Scripts: pacote que contém classes que constroem e executam quando necessários funções JavaScript.

Forms: pacote composto pelas classes que constroem os componentes HTML mais básico e com recursos avançados. Dentre estes componentes podemos citar – text, hidden, submit, reset dentre outros.

Properties: pacote escolhido para ser responsável pelas classes que são responsáveis por classe de diferentes pacotes que possuem características em comuns.

Com estas definições não se é possível entender o funcionamento da biblioteca, com o intuito de facilitar a compreensão do leitor iremos demonstrar relacionamentos existentes entre as classes do projeto, estas que foram definidas para solucionar problemas que poderiam ser cometidos facilmente em grande parte do código de um sistema desenvolvido em PHP, que utilize algum tipo de componente que vise garantir as boas práticas e faça com que o desenvolvedor tenha foco na regra de negócio.

Assegurando que o projeto foi modelado e implementado utilizando características da OO, apresentaremos o uso de herança, herança de interface e polimorfismo, tecnologias fundamentais que permitem a verdadeira programação orientada a objetos (DEITEL, H. M e DEITEL, P. J.)

A herança é uma forma de reutilização de *software* em que novas classes são criadas a partir de classes existentes. Nesta biblioteca temos um exemplo de herança representado por uma classe fundamental para a construção de componentes HTML, a *Super Classe NovemaxBasic*.

A *NovemaxBasic*, classe pai composta por operações básicas e comuns dos componentes HTML, operações estas que são reutilizadas pelas Sub Classes (classes filhas) *NovemaxInput* e *NovemaxTable*, indicando assim que estas classes herdam operações existentes na classe pai.

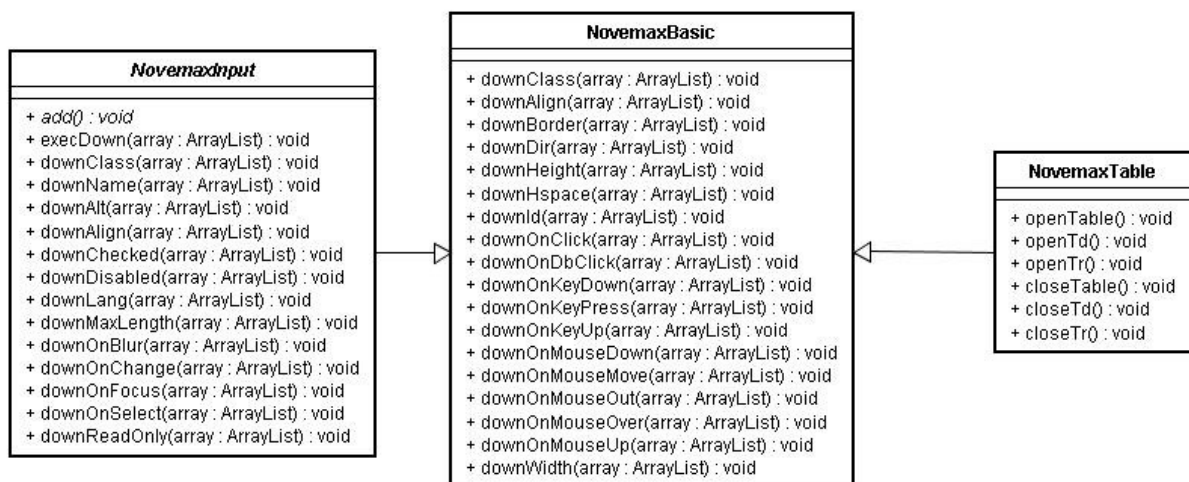


Figura 3 – Ilustração UML da Super-Classe *NovemaxBasic* e suas derivadas

A reutilização de *software* economiza tempo no desenvolvimento dos programas. Ela incentiva a reutilização de *software* de alta qualidade testados e depurado, reduzindo os problemas depois que um sistema se torna operacional (DEITEL, H. M e DEITEL, P. J.). O polimorfismo permite-nos escrever programas de uma forma geral para tratar uma ampla variedade de classes relacionadas existentes e ainda a serem especificadas, tornando fácil adicionar novos recursos a um sistema.

Representando a utilização do polimorfismo, segue abaixo a figura 4.

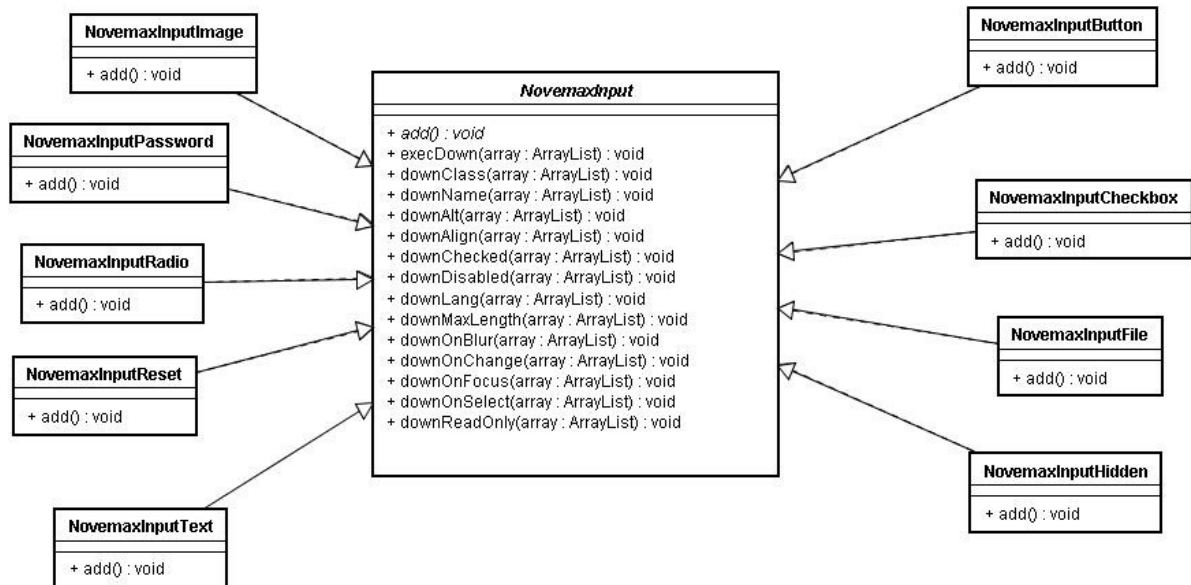


Figura 4 – Super Classe *NovemaxInput*, e suas Sub Classes implementando polimorfismo

Outra característica da OO também composta na biblioteca é a hierarquia de interface, isto é, tem sua funcionalidade em uma posição mais baixa na hierarquia – uma super classe especifica um ou mais métodos que devem ser chamados identicamente para cada objeto na hierarquia (possuem as mesmas assinaturas), mas as subclasses individuais fornecem suas próprias implementações dos métodos. Na ilustração abaixo temos a representação UML da interface *NovemaxRss*, e uma subclasse *NovemaxRssUol_*

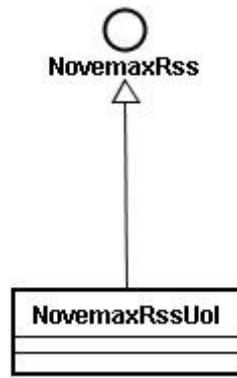


Figura 5 – Interface *NovemaxRSS*

3 Considerações Finais

Visando contemplar todo o aprendizado obtido no decorrer da elaboração da biblioteca, iremos descrever os princípios que nortearam a sua construção, tanto como os benefícios que o seu uso possibilita.

A construção da biblioteca foi projetada para ser implementada (e foi), utilizando a linguagem PHP, uma linguagem bastante utilizada no mundo web, que permite a construção de bibliotecas utilizando conceitos da Orientação a Objeto.

Levando se em consideração que a linguagem PHP é utilizada juntamente com a linguagem HTML, com o intuito de gerar código dinâmico, e que muitas esta utilização pode tornar o código confuso e de difícil entendimento. Surgiu se então a idéia de construir uma biblioteca, para gerar componentes de forma simplificada, fácil e padronizada.

A biblioteca não só melhorou o processo de implementação, como também fez a integração de forma homogênea entre as linguagens envolvidas (HTML e PHP), isso se tornou possível devido à elaboração dos componentes HTML serem desenvolvidos de forma encapsulada pela linguagem PHP.

Contudo podemos constatar que com o uso da biblioteca proposta em questão, o desenvolvimento foi facilitado, permitindo que os programadores focassem nas exigências do software e não em detalhes de baixo nível do sistema, como também possibilitou a construção de aplicações homogêneas entre as linguagens abordadas, com um código mais organizado e unificado, onde não existe complexidade para a geração dos componentes.

4 Referências

ABDEL-AAL, R.E. & AL-GARNI, Z. *Forecasting Monthly Electric Energy Consumption in eastern Saudi Arabia using Univariate Time-Series Analysis*. Energy Vol. 22, n.11, p.1059-1069, 1997.

ABRAHAM, B. & LEDOLTER, J. *Statistical Methods for Forecasting*. New York: John Wiley & Sons, 1983.

DALL’OGLIO, PABLO. **PHP Programando com Orientação a Objeto**. Editora Novatec, 2007.

DEITEL, H. M. & DEITEL, P. J. *Java Como Programar*; trad. Carlos Arthur Lang Lisbôa 4º ed. Porto Alegre Bookman, 2003.

LIM, C. & McALEER, M. *Time Series Forecasts of International Travel Demand for Australia*. Tourism Management, artigo aceito em 2001 para publicação, aguarda impressão.

MAKRIDAKIS, S.; WHEELWRIGHT, S. & HYNDMAN, R.J. *Forecasting Methods and Applications*. 3. ed. New York: John Wiley & Sons, 1998.

MELO, ALEXANDRE ALTAIR DE & NASCIMENTO, MAURICIO G. F. *PHP Profissional – Aprenda a desenvolver sistemas profissionais orientados a objeto com padrões de projeto*. Editora Novatec.

PELLEGRINI, F.R. & FOGLIATTO, F. *Estudo comparativo entre modelos de Winters e de Box-Jenkins para a previsão de demanda sazonal*. Revista Produto & Produção. Vol. 4, número especial, p.72-85, 2000.

STROUSTRUP, B. *The C++ Programming Language*. Massachusetts, 1997.