

METODOLOGIA E FERRAMENTA PARA ENSINO DA PROGRAMAÇÃO ORIENTADA A OBJETOS

Mariane Fogaça Galhardo – marianefg@yahoo.com.br

Faculdade de Engenharia de Sorocaba
Rodovia Senador José Ermírio de Moraes, km 1,5
18001-970 – Sorocaba – SP

Luciana A. M. Zaina – luciana@facens.br

Resumo: *O desenvolvimento de metodologias e ferramentas auxiliares ao ensino da programação tem crescido, mostrando a preocupação existente em transpor conceitos estudados para uma linguagem de programação de forma prática, sem abandonar a importância do entendimento conceitual.*

O objetivo deste artigo é apresentar uma metodologia para acompanhar o aluno durante o ensino dos conceitos fundamentais do paradigma de programação orientado a objetos. Para isso, será apresentado um protótipo de uma ferramenta Web, que permitirá ao docente acompanhar o aluno durante a resolução de exercícios.

Nessa metodologia o docente proporá exercícios onde o aluno poderá construir passo a passo um programa no paradigma citado, sem se preocupar com a estrutura da linguagem adotada.

Assim que o aluno submete o exercício o professor poderá verificá-lo indicando se os objetivos propostos foram atingidos, conduzindo o aluno a reforçar seus estudos em conceitos não assimilados ou propondo novos exercícios que abordem outros conceitos.

Palavras-chave: *Tecnologia para ensino, Programação orientada a objetos, Metodologia para acompanhamento do aprendizado do aluno.*

1. INTRODUÇÃO

A importância do ensino da programação está não só no conteúdo abordado, mas também na capacidade de transpor os conceitos estudados para uma linguagem de programação de forma prática.

O desenvolvimento de ferramentas que atuem como apoio ao ensino da programação tem sido uma alternativa amplamente adotada dentro do ensino de linguagens de programação. Tais ferramentas têm por objetivo enriquecer o processo de ensino aprendizagem e motivar os alunos apresentando-lhes diferentes formas de visualizar os conceitos estudados.

SCHIMIGUEL (2003), por exemplo, apresenta como alternativa para ensino da programação uma ferramenta que utiliza a técnica de desenvolvimento do fluxograma para construção da lógica de programação do aluno. Outros como PIERSON (1998) e STERN (1999) apresentam algoritmos animados utilizados para o ensino de estruturas de dados.

As dificuldades no ensino das linguagens de programação podem ser de diversas naturezas, desde problemas com a estrutura da linguagem escolhida até as diferentes formas de aprender que um aluno pode ter, segundo JENKINS (2002).

O objetivo deste trabalho é apresentar uma metodologia para acompanhamento do aluno durante a resolução de exercícios no paradigma orientado a objetos. Nessa metodologia o docente proporá exercícios onde o aluno poderá construir passo a passo um programa no paradigma citado, sem se preocupar com a estrutura da linguagem adotada. Assim que o aluno

submete o exercício o professor poderá verificá-lo indicando se os objetivos propostos foram atingidos, conduzindo o aluno a reforçar seus estudos em conceitos não assimilados ou propondo novos exercícios que abordem outros conceitos. O foco principal é que através de exercícios pré-cadastrados, sob a supervisão do docente, a ferramenta direcione o aluno através de exercícios, auxiliando na sedimentação de sua aprendizagem no paradigma de programação em questão. Adicionalmente, será armazenado um histórico dos exercícios realizados pelo aluno, permitindo ao docente realizar uma análise de sua evolução.

O desejo é que o aluno iniciante neste novo paradigma de programação se concentre na aplicação dos conceitos que envolvem a construção do programa. A metodologia encontra-se em fase de implementação, porém o núcleo da ferramenta já foi desenvolvido e será apresentado neste trabalho. A ferramenta permite que o aluno através de interações vá construindo seu programa, sem se preocupar com os aspectos estruturais da linguagem se concentrando em transpor os conceitos aprendidos para a modelagem do problema. Ao final é gerado um código na linguagem Java como resultado das interações realizadas pelo aluno.

2. ENSINO DA PROGRAMAÇÃO ORIENTADA A OBJETOS

Visualizar a construção de programas em diversos paradigmas de linguagens de programação não é transparente ao aluno. As dificuldades encontradas durante o processo de aprendizagem são reflexo de diversos problemas como lógica de programação, a não familiaridade com um determinado paradigma, entre outros. Estas barreiras podem se tornar ainda maiores quando a abordagem está relacionada a conceitos da programação orientada a objetos, JONES (2003), JENKINS (2002) e THRAMBOULIDIS (2003).

Normalmente, o percurso a ser percorrido pelo aluno durante o ensino da programação, adotado pelos diversos cursos de graduação, se inicia com a aprendizagem da programação imperativa, pois a mesma possibilita que a lógica de programação possa ser explorada nos alunos com maior facilidade. Quando o aluno começa a ter contato com uma linguagem orientada a objetos ele sente dificuldades em transpor as barreiras que diferenciam os estilos de programação imperativa e orientada a objetos.

Algumas experiências têm sido realizadas mundialmente tentando modificar o percurso de ensino de linguagens de programação, começando pelo ensino orientado a objetos. Porém, os resultados encontrados ainda não motivam a mudança. Segundo BURTON (2003), o paradigma orientado a objetos utiliza estruturas existentes no paradigma imperativo. Caso seja invertida a ordem no processo de aprendizagem destes paradigmas, seria necessário explorar os aspectos estruturais imperativos primeiro para depois introduzir os conceitos da orientação a objetos propriamente dito.

É perceptível que o raciocínio humano possui uma modelagem mais próxima das técnicas de programação imperativas. Embora o processo de abstração de objetos utilizado na programação orientada a objetos aproxime o aluno do mundo real durante o a concepção do problema a ser resolvido, conforme SEBESTA (2000), as dificuldades na concretização da constituição geral do problema tornam-se maiores no paradigma orientado a objeto, segundo BURTON (2003).

Atualmente muitas são as estratégias adotadas por docentes para ensino da programação orientada a objetos, como pode ser observado em THRAMBOULIDIS (2003), ROLSTAD (2003), JONES (2003) e MARTINEZ-UNANUE (2002) que adotam ferramentas que estimulem o aluno durante o ensino, procurando resgatar conhecimentos já adquiridos pelos mesmos. Em resumo, o objetivo de todas as estratégias é tornar mais simples e transparente o processo de aprendizagem da técnica orientada a objetos.

MARTINEZ-UNANUE (2002) aborda que embora nos dias atuais as ferramentas de desenvolvimento facilitem a construção de programas, as mesmas ainda não satisfazem todas as necessidades do ponto de vista educacional. É necessário que haja um enfoque mais didático e pedagógico para obter maior eficiência no ensino.

Portanto, pode-se concluir que é importante haver um complemento entre o processo pedagógico do ensino dos conceitos e a tradução dos mesmos para o paradigma de programação que está sendo adotado. Desta forma, apresenta-se as possibilidades ao aluno para que ele possa sair do processo de abstração de um problema para a concretização de fato do mesmo.

3. METODOLOGIA E FERRAMENTA PARA ACOMPANHAMENTO DO ALUNO

A partir das dificuldades encontradas na transição do ensino da programação imperativa para o ensino orientado a objetos decidiu-se desenvolver uma metodologia que apoiasse os alunos iniciantes neste novo paradigma. Esta tem como foco principal o apoio aos conceitos estudados durante as aulas presenciais de um curso de Engenharia da Computação.

O objetivo é que a ferramenta de apoio desenvolvida a partir da metodologia possa ser acessada em qualquer momento e local pelo aluno. Para satisfazer esta necessidade decidiu-se pela adoção da tecnologia de Servlets, devido à mesma ser multiplataforma e ser uma tecnologia para Web.

Na primeira fase do projeto, foi concebida a metodologia para acompanhamento e desenvolvido o núcleo da ferramenta onde é possível ao aluno construir passo a passo a classe resultado da solução de um problema apresentado.

3.1. Metodologia

O acompanhamento da evolução do aluno durante a resolução de exercícios não é uma tarefa muito fácil. Realizar um histórico desta tarefa é de extrema importância e deve auxiliar docentes no processo de ensino-aprendizagem.

A partir da metodologia de acompanhamento do aprendizado desenvolvido em ZAINA (2002) é possível cadastrar uma série de exercícios e controlar o caminho seguido pelo aluno dependendo da resposta dada por ele durante a resolução da série. Tal metodologia determina que o aluno é direcionado a exercícios conforme seu desempenho na resolução de exercícios anteriores. Desta maneira, o docente determina o grau de importância dos conceitos e constrói a seqüência mais adequada a ser resolvida.

O professor cadastra os exercícios dividindo os mesmos por assunto e definindo a descrição de cada um. Durante o cadastro dos exercícios, o professor define também o próximo exercício que o aluno deverá resolver caso o exercício, resolvido pelo aluno, esteja correto.

O objetivo principal da ferramenta desenvolvida a partir da metodologia será acompanhar o aluno durante a resolução de exercícios. A seqüência funcional do acompanhamento pode ser observada na Figura 1. A seqüência seguida pelo aluno será: o aluno escolhe o exercício, resolve o mesmo e salva. Em seguida, o docente é avisado que o exercício já foi resolvido e começa o processo de correção do mesmo. A ferramenta envia automaticamente um e-mail ao aluno avisando que o exercício foi corrigido, assim que o docente finaliza este procedimento. Se o exercício esteja correto, o aluno passa para o próximo da seqüência, caso contrário o aluno deverá refazer o exercício para procurar sanar suas falhas de aprendizagem. Neste ponto ele recebe orientação do docente que irá realizar observações a respeito da correção realizada.

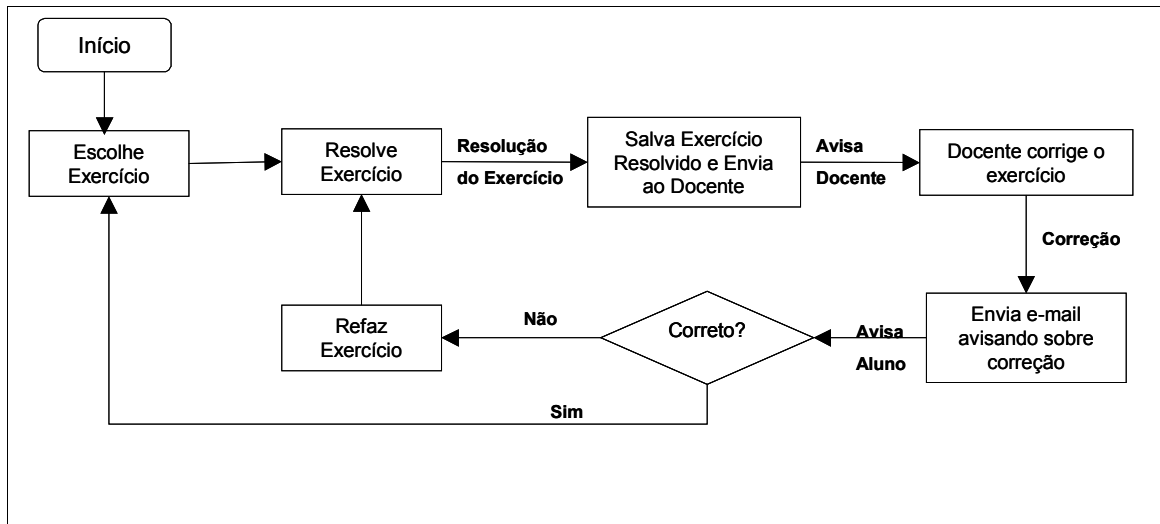


Figura 1 – Sequência funcional da ferramenta

O exercício resolvido é armazenado no sistema com data e hora da resolução final. Quando o professor acessa o sistema, tem acesso aos exercícios resolvidos pelos alunos. Este histórico é importante para que o docente possa acompanhar de fato a evolução do aluno, comparando as diversas informações colhidas pelo sistema.

A ferramenta possibilita ao aluno saber se o exercício a ser resolvido é o primeiro da série. Permite também o que ele verifique a correção de seu exercício, podendo assim analisar seus erros. A Figura 2 apresenta o funcionamento da resolução dos exercícios. Quando o aluno seleciona um exercício para resolução a ferramenta verifica se é o primeiro exercício de uma série. Caso não seja, é necessário checar se o exercício anterior da série está correto, para que o aluno esteja apto de fato a resolver o próximo exercício. Desta forma, o docente consegue controlar quando um aluno está apto ou não a prosseguir dentro de um conjunto de conceitos.

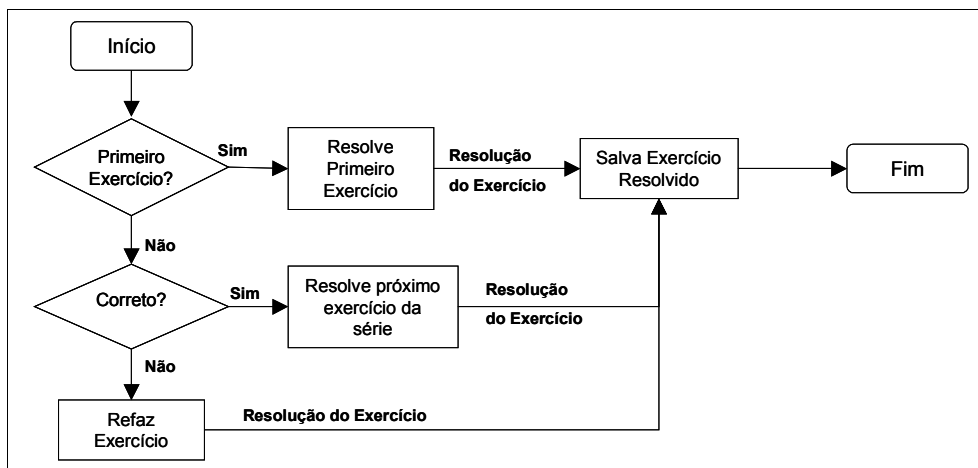


Figura 2 - Resolução de exercícios através da visão do aluno

Para corrigir os exercícios, o professor deverá selecionar o assunto e a descrição do exercício a ser corrigido. Dessa forma, a ferramenta exibirá os exercícios que foram resolvidos em uma tabela. Após terminar a correção, será possível ao professor acrescentar observações sobre a resolução do aluno e definir se ele foi aprovado na resolução ou não. A Figura 3 apresenta todo o processo de cadastro e correção de exercícios na visão do professor.

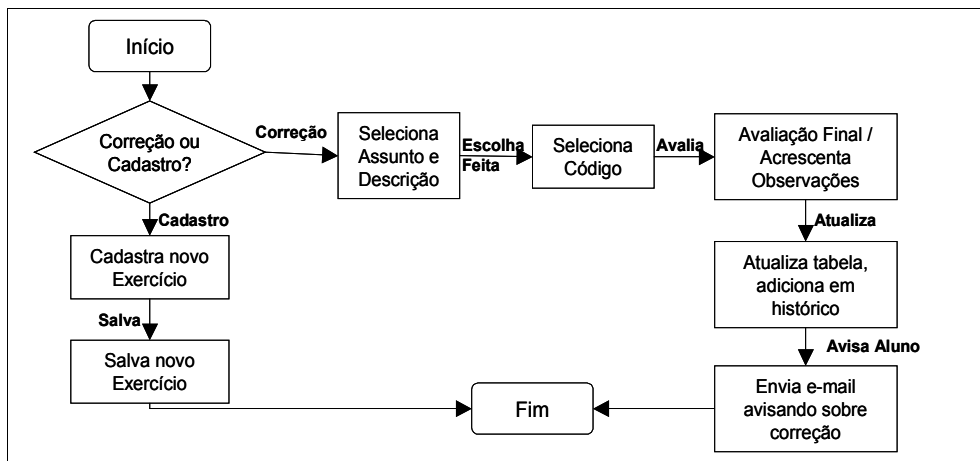


Figura 3 - Fluxograma: atividades do docente

3.2. Núcleo da ferramenta

O protótipo desenvolvido até o momento possibilita que o docente cadastre enunciados de exercícios propostos - Figura 4 - e o aluno tem a possibilidade de escolher o enunciado do exercício que deseja resolver, conforme observado na Figura 5.

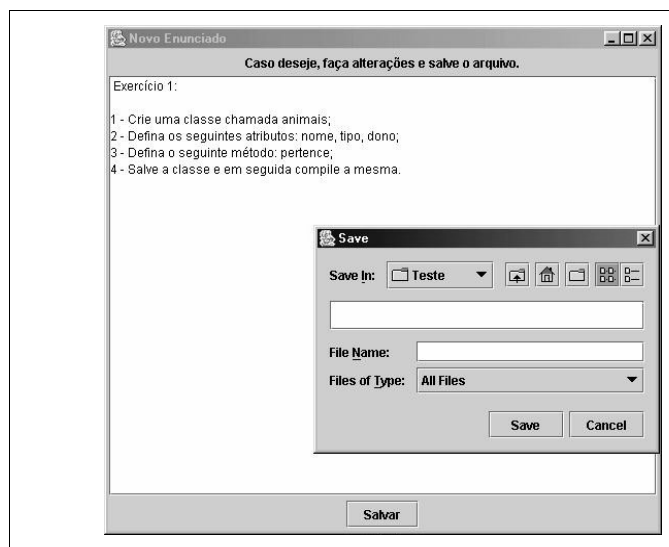


Figura 4 - Cadastro dos enunciados de exercícios propostos aos alunos

Após escolher o exercício e ler o respectivo enunciado o aluno pode iniciar o processo de resolução. Este módulo do protótipo permite que o aluno interaja com a ferramenta para construção da solução do exercício passo a passo. É importante destacar que todo este processo não depende do exercício que o aluno está resolvendo. A ferramenta irá construir a solução de acordo com os dados informados pelo aluno.

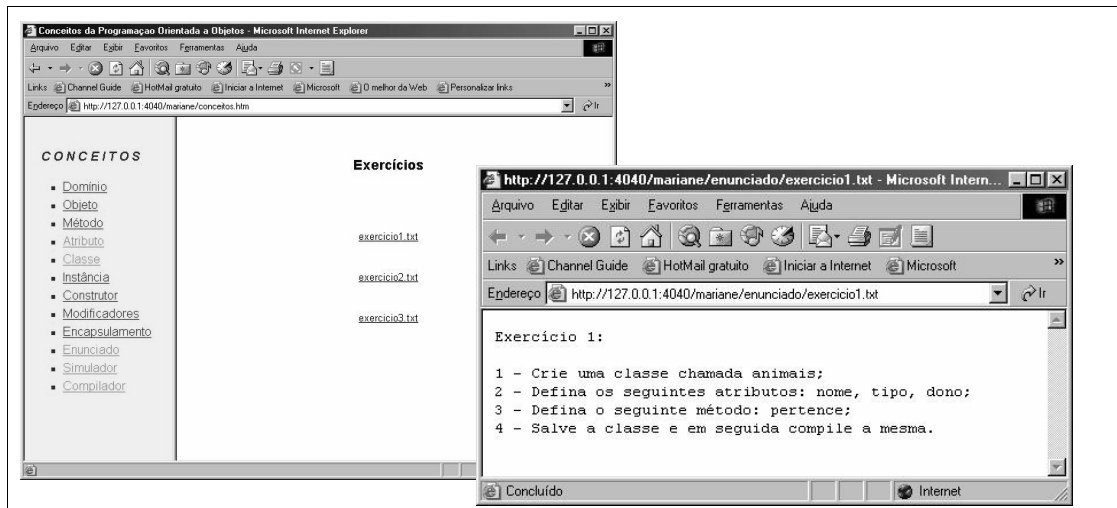


Figura 5 - Enunciados dos exercícios disponíveis aos alunos

Durante o processo de resolução do exercício o aluno irá, gradativamente, fornecer todos os dados necessários para construção do código-solução, que partem das características gerais da classe, passam pela definição dos atributos e métodos e terminam com a definição dos valores iniciais dos atributos para serem utilizados pelo construtor da classe.

O percurso percorrido pela ferramenta é apresentado na Figura 6.

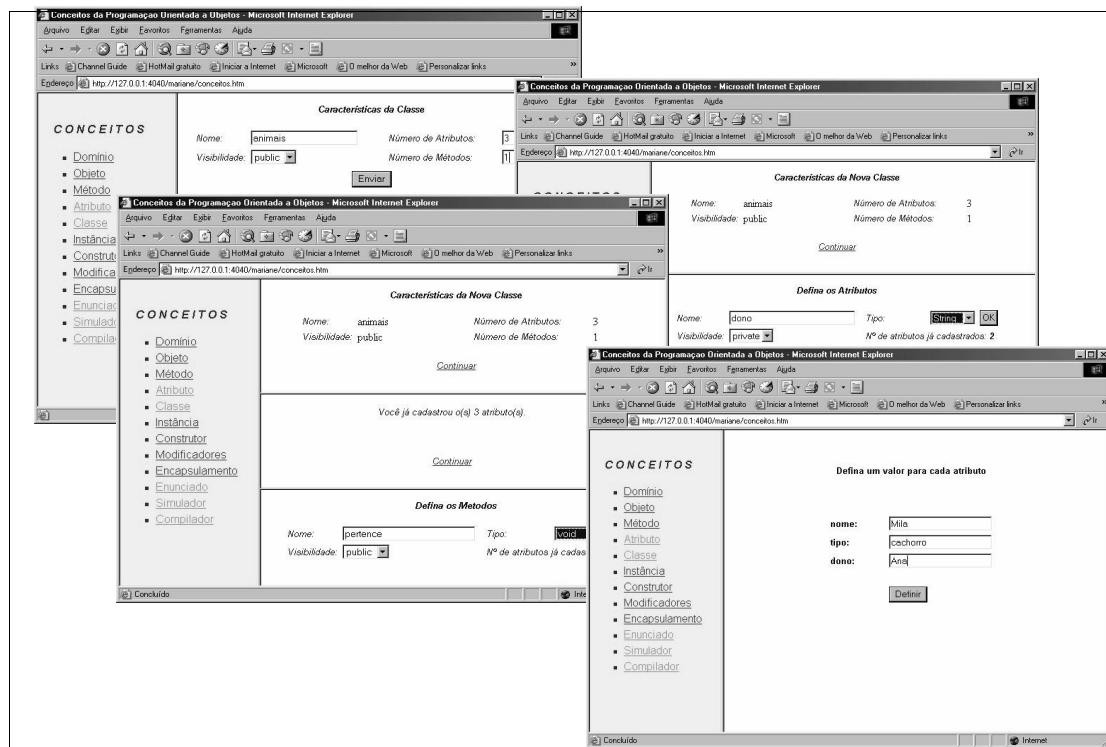


Figura 6 - Construção da solução do exercício passo a passo

Ao finalizar a resolução do exercício a ferramenta gera para o aluno o código correspondente aos dados que ele informou em Java. Durante o processo de interação com a ferramenta, o aluno não necessita se preocupar com a sintaxe da linguagem. Ele concentra-se

na busca da solução do exercício, apoiado nas definições conceituais de um código orientado a objetos. Porém, como resultado deste processo de interação ele verá o código gerado em uma linguagem pertencente a este paradigma.

Além de visualizar o código o aluno também pode alterar o mesmo caso seja necessário. É disponibilizado, também o recurso que permite ao aluno salvar o código gerado em qualquer pasta que desejar, conforme observado na Figura 7.

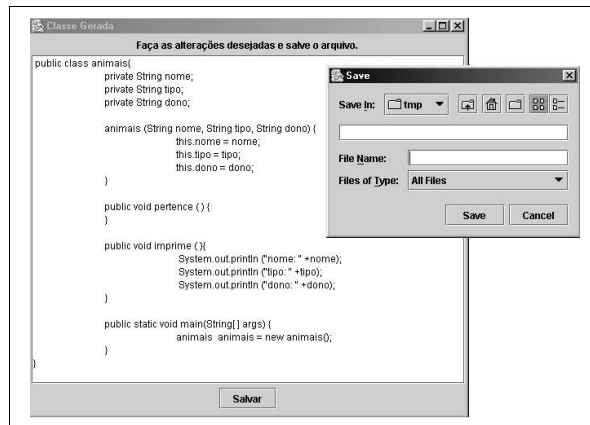


Figura 7 - Código gerado pela ferramenta que pode ser armazenado pelo aluno

Após ter construído a solução para o exercício, o aluno terá a possibilidade de analisar o código gerado e comparar o mesmo com as definições e exemplos dos conceitos da orientação a objetos disponibilizados na ferramenta. Desta forma, ele pode realizar um paralelo do abstrato (conceitos) para o concreto (código gerado).

Outro recurso disponibilizado pela ferramenta é a compilação do código gerado. O aluno pode realizar o processo de compilação da classe, pois caso ele tenha feito alguma modificação na classe, possíveis erros poderão ser detectados. Como resultado a ferramenta irá apresentar um relatório de erros ou uma mensagem de sucesso da compilação, como pode ser observado na Figura 8.

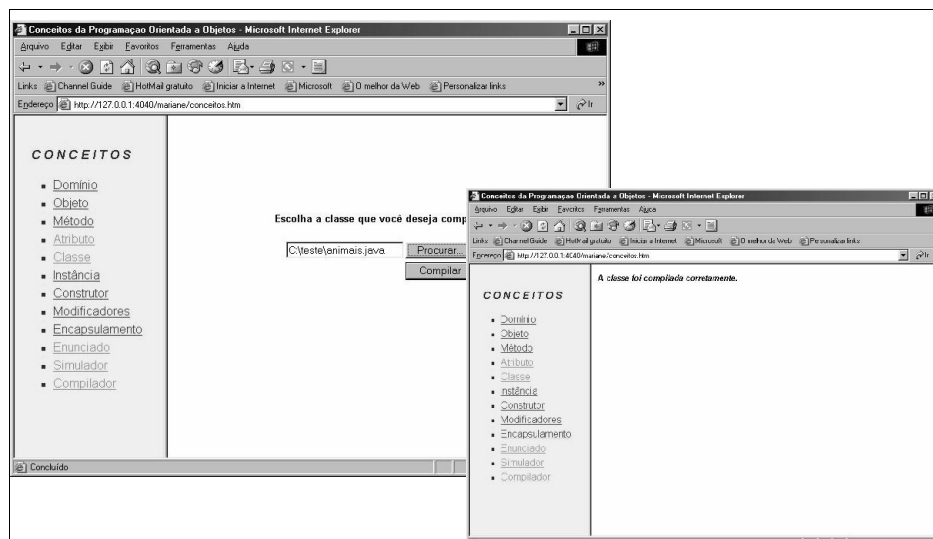


Figura 8 - Compilação do código gerado pela ferramenta

4. CONCLUSÕES

A utilização de metodologias para auxiliar alunos a assimilarem os conceitos do paradigma orientado a objetos tem crescido, devido à preocupação em facilitar o entendimento dos estudantes neste paradigma sendo um complemento de aulas presenciais.

É inerente ao aluno que está iniciando um curso de programação ter uma afinidade maior com o paradigma imperativo, já que o mesmo se aproxima mais dos processos lógicos utilizados na resolução de problemas. É importante, então, apresentar aos alunos alternativas diferentes que auxiliem a aprendizagem de um novo paradigma de programação.

O protótipo apresentado neste trabalho vem dar subsídios aos alunos iniciantes na programação orientada a objetos para que seus esforços sejam concentrados nos conceitos fundamentais durante o estudo. Para tanto, a ferramenta disponibiliza os conceitos iniciais e necessários ao aluno para desenvolver códigos modelados em uma linguagem orientada a objetos.

A interação do aluno durante a resolução do exercício é fundamental para que ele consiga visualizar como acontece a transcrição do conceitual para o real. A classe gerada possibilita ao aluno verificar como os dados por ele informados resultam em um código estruturado no paradigma orientado a objetos.

Deve-se ressaltar a importância do aluno ter bem sedimentado os conceitos da programação orientada a objetos antes de construir seus códigos. O entendimento das conceituações passa a ser fundamental para a geração do resultado final. Além disto, fazer a transcrição de abstrações para a geração concreta de um código é muito valorosa para que o aluno entenda o processo de construção que deve ser utilizado para a resolução de problemas no paradigma de programação abordado.

A ferramenta descrita neste trabalho, além de acompanhar passo a passo o desenvolvimento de uma classe gerada a partir de um problema proposto, possibilita ao aluno ser acompanhado através da Web pelo professor durante a construção do processo de ensino-aprendizagem.

Dessa forma, o aluno tem a possibilidade de ter um retorno mais rápido, podendo inclusive dar continuidade na seqüência de exercícios a ser resolvida. Além disso, a metodologia descrita proporciona um histórico do aluno, permitindo ao docente realizar uma análise gradual de cada estudante.

REFERÊNCIAS BIBLIOGRÁFICAS

BRIDGEMAN, S et al. An Interactive Tool for Learning and Grading. In: SIGCSE BULLETIN CONFERENCE PROCEEDINGS – THE 31^o SIGCSE TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION, Austin, TX, USA: 2000. p.139-142.

BURTON, P. J. et al. **Teaching Programming in the OOP Era**. Arkansas, v.35, n.3, p.111-114, 2003.

JENKINS, T. On The Difficulty of Learning to Program. In: 3rd ANNUAL LTSN-ICS CONFERENCE, UK: 2002. p. 53-58.

JONES, R et al. ObjectWorld: Helping Novice Programmers to Succeed Through a Grafical Objects-first Approach. In: 4th ANNUAL LTSN-ICS CONFERENCE, UK: 2003. p. 111-114.

KORHONEN, A et al. Algorithm Simulation with Automatic Assessment. In: SIGCSE BULLETIN – THE 5^o ANNUAL SISCSE/SIGCUE CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION, Finlândia: 2000. p. 160-163.

MARTINEZ-UNANUE, R. et al. **Electronic books for programming education: a review and future prospects**. Dinamarca, v.34, n.3, p. 34-38, 2002.

OLIVEIRA, I. C. A. et al. Objetos de Aprendizagem para o Ensino de Programação em Engenharia e Computação. In: 3rd INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION. São Paulo: 2003. p. 1-5

PIERSON, W. C. et al. **WEB-based Animation of Data Structures Using JAWAA**. v.30, n.1, USA, p. 267-271, 1998.

ROLSTAD, C. Using Image Processing and Visualization to Teach Object Oriented Programming in C++. In: INTERNACIONAL CONFERENCE ON ENGINEERING EDUCATION. Spain: 2003. p. 1-5

SCHIMIGUEL, J. et al. Desenvolvimento de Simulações para o Aprendizado em cursos na Web. In: 3rd INTERNATIONAL CONFERENCE ON ENGINEERING AND COMPUTER EDUCATION. São Paulo: 2003. p. 1-5.

SEBESTA, R. W. **Conceitos de Linguagens de Programação**. Bookman, 2000.

SIMONSON, M. et al. **Teaching and Learning at a Distance. Foundations of Distance Education**. United States of America: Prentice Hall, 2000

STERN, L. et al. A Strategy for Managing Content Complexity in Algorithm Animation. In: SIGCSE BULLETIN – THE 4^o ANNUAL SISCSE/SIGCUE CONFERENCE ON INNOVATION AND TECHNOLOGY IN COMPUTER SCIENCE EDUCATION. Austrália: 1999. p.127-130.

THRAMBOULIDIS, K. et al. Teaching Advanced Programming Concepts in Introductory Computing Courses: A Constructivism Based Approach. In: INTERNACIONAL CONFERENCE ON ENGINEERING EDUCATION. Spain: 2003. p. 2-7

ZAINA, L. A. M. **Acompanhamento do Aprendizado do Aluno em Cursos a Distância através da Web: Metodologias e Ferramenta**. 2002. Dissertação(Mestrado em Engenharia Elétrica – ênfase em Sistemas Digitais), Universidade de São Paulo, São Paulo.

METHODOLOGY AND TOOL TO OBJECT-ORIENTED PROGRAMMING EDUCATION

Abstract: *The development of methodologies and tools to support the programming teaching has increased, showing the worry about translating concepts learning in a programming language in practical way, without forgetting the importance of conceptual understanding.*

The goal of this article is to present a methodology to track the student during the teaching of the fundamental concepts of object-oriented programming paradigm. A prototype of a Web tool will be showed, allowing the teacher to track the student evolution during the exercises resolution.

In this methodology the teacher will propose exercises which the student can build step-by-step a computer program in paradigm cited without worrying with the structure of the adopted language.

As soon as the student sends the exercise, the teacher can check it, pointing if the proposed objectives were reached, guiding the student to reinforce his studies in not assimilated concepts or the teacher can propose new exercises that include other concepts.

Key-words: *Technology to education, Object-oriented programming, Methodology to track the student learning.*