FUNDAÇÃO EDUCACIONAL DO MUNICÍPIO DE ASSIS – FEMA

INSTITUTO MUNICIPAL DE ENSINO SUPERIOR DE ASSIS - IMESA

Greenfoot: Uma Ferramenta para o ensino de programação orientada a objetos

Área de Pesquisa: Ciências Exatas e da Terra

Assis – Novembro/2012

FUNDAÇÃO EDUCACIONAL DO MUNICÍPIO DE ASSIS – FEMA

INSTITUTO MUNICIPAL DE ENSINO SUPERIOR DE ASSIS - IMESA

Relatório final submetido à Comissão
do Programa de Iniciação Científica da
FEMA/IMESA 2012.

Autores:
Ariane de Oliveira Silva - Aluna
Professor Dr. Luiz Carlos Begosso - Orientador

SUMÁRIO

1. Programação Orientada a Objetos

A Programação Orientada a Objetos é um paradigma de programação que foi criado na década de 80 e que, segundo Silva Filho (2010), " A Programação Orientada a Objeto é uma das maiores inovações na área de desenvolvimento de software".

Programação Orientada a Objetos (POO) se apresenta como um paradigma de programação que permite aos programadores raciocinar e solucionar problemas em termos de objetos, os quais estão diretamente associados às entidades, elementos, do mundo real.

A POO descreve objetos físicos do mundo real como, por exemplo, livro e aluno através das entidades denominadas objetos, em uma linguagem orientada a objetos.

A principal diferença entre a linguagem orientada a objeto e as demais é a forma como um problema é tratado, o programa é dividido em objetos, e não mais em funções.

2. Justificativa

O presente projeto de Iniciação Científica se justifica por expressar a preocupação com o ensino e aprendizagem dos alunos ingressantes em cursos de informática, na disciplina de algoritmos e programação.

3. Objetivo

Este trabalho tem como objetivo ensinar os principais conceitos da programação orientada a objeto a alunos das 1° séries dos cursos da área de informática. O software Greenfoot, foi utilizado para atender o objetivo estabelecido.

4. Greenfoot

Uma das curiosidades, e o objetivo do software Greenfoot começa pelo seu nome e o símbolo que o representa, pois porque dar o nome, e utilizar como símbolo de um software, um pé verde? Segundo um de seus criadores Michael Kolling o termo Greenfoot é derivado de uma antiga fábula dos aborígenes australianos onde existia um personagem cujo nome era Greenfoot, responsável por iluminar com conhecimento as pessoas dessa tribo.

É este o motivo pelo qual o software Greenfoot possui este nome e o símbolo do pé verde, com o objetivo de iluminar as pessoas no que diz respeito a programação orientada a objetos.

O *Greenfoot* é um software que foi desenvolvido em JAVA por pesquisadores de duas universidades: a Universidade de Kent, na Inglaterra, e a Universidade de Deakin, na Austrália, no ano de 2006. Foi desenvolvido para fins educacionais, ou seja, ensinar programação orientada a objetos a partir da construção de cenários utilizando como linguagem de programação JAVA, permitindo que alunos de cursos da área da computação possam obter experiência em programação orientada à objetos a partir do Greenfoot.

O software Greenfoot permite o desenvolvimento de aplicações gráficas como criação de jogos em ambiente 2D. Através da visualização e interação de objetos, é possível criar mini-mundos (ambiente de uma situação), e representar graficamente seus objetos.

Greenfoot possui elementos típicos de um ambiente de desenvolvimento, como o editor de código-fonte, compilação, etc., o que diferencia o de outros ambientes é a interação direta. Por exemplo: ao instanciar um objeto, ele pode ser colocado em qualquer lugar do “mundo”, e quando é invocado um método que muda a representação gráfica do objeto, esta ação pode ser visualizada imediatamente.

Por este motivo é considerado um Software de fácil manuseio, por possuir muitos detalhes visuais, proporcionando assim uma interação direta com o usuário que pode observar, no cenário construído, as mudanças realizadas no código fonte que afetaram o comportamento do objeto.

No site do Greenfoot (http://www.greenfoot.org) pode-se realizar o download do software e dos cenários utilizados para a programação.

No referido site é possível ter acesso à galeria do Greenfoot, assim como os projetos desenvolvidos por outras pessoas ao redor do mundo. É possível também postar dúvidas, e realizar perguntas, que serão respondidas por outros usuários do próprio Greenfoot, podendo até mesmo postar os cenários que desenvolveram para serem analisados por esses usuários, que podem dar notas, e fazer comentários favoráveis ou não quanto ao cenário postado.

4.1. Quem Utiliza

O Software Greenfoot é utilizado por estudantes que não possuem muito conhecimento com a programação orientada a objetos e que querem adquirir tal conhecimento de uma maneira diferente. A partir da criação de cenários, e de jogos, utilizando os principais conceitos da programação orientada a objetos, é possível que os usuários adquiram conhecimento sobre os conceitos fundamentais desta programação de uma maneira interativa, utilizando uma ferramenta com interações gráficas bidimensionais.

O software, permite que esses alunos aprendam conceitos importantes de tal programação de uma forma diferente com desenvolvimento de jogos, fazendo com que os alunos se dediquem mais aos cenários que estão desenvolvendo.

4.2. Trabalhos Correlatos

Al-Bow et al (2008) relatam ter utilizado o Greenfoot com 17 estudantes que não possuíam nenhuma experiência em programação orientada a objetos.

Após um período de utilização do software, os autores constataram significativo aprendizado de aspectos relacionados aos princípios de orientação a objetos.

Numa outra experiência de utilização do Greenfoot, relatada por Gallant e Mahmoud (2008), um grupo de pesquisadores reuniu alguns estudantes para realizarem o projeto *Going To The Moon* um dos cenários Greenfoot, para através deste projeto manter os alunos interessados e envolvidos no processo de aprendizagem de programação. No final deste projeto os pesquisadores esperam formar jovens realmente interessados e com habilidades em programação, para assim ajudarem a dar continuidade ao Greenfoot com a criação de novos cenários.

Vahldick e Mattos (2008), relatam que pesquisadores utilizavam caixas de sapato confeccionadas com abas externas para fixação de tiras de papel para anotações das características dos objetos. Na tampa da caixa foram anotados o nome do objeto e o tipo do objeto, na lateral foi relatado o que se pode fazer com o objeto, ou seja, a lista de métodos, e dentro da caixa os atributos desse objeto. Notou – se que essa abordagem foi muito útil para os alunos, pois mais adiante foi observado que eles não costumavam mais utilizar as caixas para executar os exercícios de orientação a objeto.

5. Estudo de Caso

Entre os meses de outubro e dezembro foi ministrado um mini-curso, com duração de 10 horas, para alunos do primeiro ano do curso de informática da Fundação Educacional do Município de Assis (FEMA). O conteúdo ministrado abordou os principais conceitos da programação orientada a objetos utilizando como ferramenta de ensino cenários do software Greenfoot.

O curso foi realizado aos sábados no Laboratório de informática da FEMA com duração de duas horas a cada encontro. Participaram do curso cinco alunos onde quatro são estudantes do primeiro ano do curso de Análise e Desenvolvimento de Sistemas da FEMA, com faixa etária entre 18 e 32 anos, destes cinco integrantes três eram do sexo feminino e dois do sexo masculino.

O quinto integrante do curso é uma adolescente de 12 anos. Mesmo tendo pouca idade e nenhum contato anterior com programação, foi impressionante o modo como ela lidou com o Greenfoot e conseguiu acompanhar os demais alunos.

5.1 Avaliação

No último dia do curso os alunos realizaram uma avaliação que abordava todo o conteúdo ministrado no mini-curso, onde através desta foi possível analisar os seus graus de aprendizagem. A avaliação está no Anexo 1.

Esta avaliação continha dez questões, algumas de múltipla escolha e outras dissertativas, que abordavam os principais conceitos da programação orientada a objetos como: analisar as classes e ver qual a relação entre elas, como podemos montar a assinatura de um método, chamada de método, tipos de parâmetros. Foi cobrado também alguns métodos que eles viram durante o curso como qual ação que eles executam quando utilizados e em quais situações devemos usá-los.

Os alunos obtiveram uma boa margem de acertos na avaliação que se justifica pelo conteúdo estudado.

A 1ª. questão da avaliação era sobre assinatura de métodos, possuía cinco alternativas das quais apenas uma era verdadeira a porcentagem de acerto nesta questão foi de 100%. O sucesso nesta questão ocorreu porque os alunos foram instigados a criarem novos métodos para isso eles precisavam definir as suas assinaturas.

A 2ª. questão apresentava um diagrama de classes onde os alunos deveriam analisar do que este diagrama era composto, neste caso de classes, e qual a relação que estas possuíam, no caso herança. Nesta segunda questão a porcentagem de acerto foi menor, onde apenas um aluno a acertou por completo e os outros alunos erraram apenas a relação que estas possuíam, acertando porém que o diagrama era composto por classes.

A 3ª. questão 75% dos alunos reponderam de forma correta. Esta questão continha duas colunas, uma coluna composta por métodos e a outra continha

explicações destes métodos, onde os alunos deveriam relacionar as duas de forma correta. O elevado grau de acerto desta questão se justifica pelo motivo de que eles haviam utilizados tais métodos para criarem cenários no Greenfoot.

Sobre a 4ª. questão 50% dos alunos a acertaram por completo. Esta questão abordava o método construtor, muito importante na programação orientada a objetos, onde os alunos deveriam avaliar quais questões eram verdadeiras.

A 5ª. questão 50% dos alunos acertaram. Esta questão era de múltipla escolha, abordava sobre o método addObject, utilizado para posicionar objetos no mundo, os alunos deveriam avaliar a assinatura do método e ver qual seu tipo de retorno, nome, e os tipos de parâmetros que ele possui.

Em relação a 6ª. questão, 100% dos alunos a responderam de forma correta. Nesta questão, os alunos deveriam observar as alternativas e completar as frases que tratavam sobre o tipo de retorno dos métodos. Quando void o que ele representa e quando o tipo de retorno é diferente de void o que ele representa. O elevado grau de acerto desta questão se justifica pelo fato de que este tema foi muitas vezes enfatizado durante o curso visto que este faz parte das assinaturas dos métodos e para poder elaborar um método e consequentemente sua assinatura esta ideia de tipo de retorno dos métodos deveriam estar bem clara para eles.

A 7ª. questão 25% dos alunos a acertaram por completo, era uma questão de alternativa onde todas elas eram verdadeiras, abordando vários conceitos como subclasses, métodos, chamadas de métodos, códigos fontes.

A 8ª. questão 100% dos alunos a acertaram. Era uma questão de múltipla escolha onde os estudantes deveriam analisar o parâmetro, e escolher apenas uma alternativa correta, visto que parâmetro é um conceito muito importante, foi muitas vezes enfatizado durante o curso, pois todos os métodos possuem parâmetros ou apenas uma lista de parâmetros vazia.

A 9ª. questão 100% dos alunos a acertaram, também era uma questão de múltipla escolha, onde os alunos deveriam observar uma determinada situação, em particular, localizar um objeto no cenário Greenfoot. Visto que tal método

também trabalha com parâmetros, pois a posição, e resolução do objeto, são passados através de parâmetros, justificando assim o alto nível de acerto nesta questão.

Finalmente, a 10ª. questão 100% dos alunos responderam de forma correta. Nesta questão haviam duas colunas: uma com nome dos métodos e a outra coluna com as explicações destes métodos. Os alunos deveriam analisar as alternativas e colocar o número na explicação que correspondia corretamente com o método.

As figuras 1 e 2 representam o grau de aproveitamento dos alunos na avaliação. A partir destes gráficos podemos analisar o rendimento dos alunos referente a avaliação onde o elevado grau de rendimento se justifica pelo fato de que os alunos não tiveram apenas aulas teóricas mas também aulas práticas. A partir do momento que os alunos aprenderam novos conceitos, eles já os testavam na prática diante da confecção de cenários no Greenfoot, enfatizando assim o conteúdo estudado.

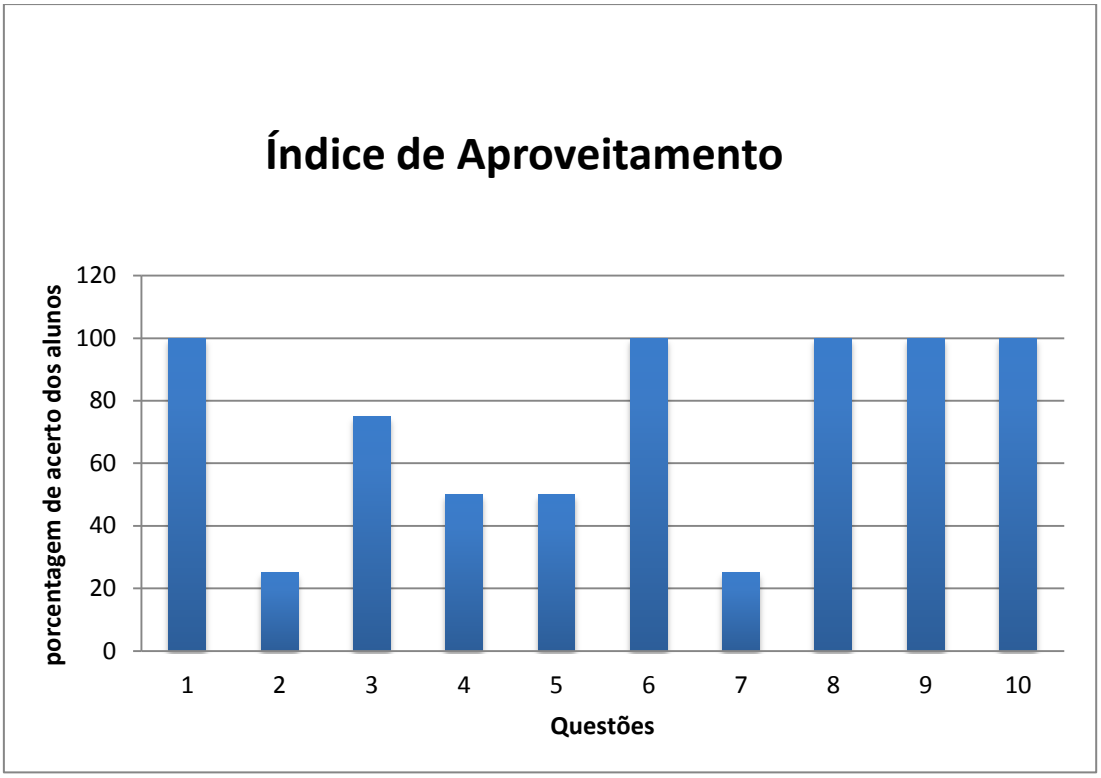


Figura 1 - Acerto dos alunos por questão

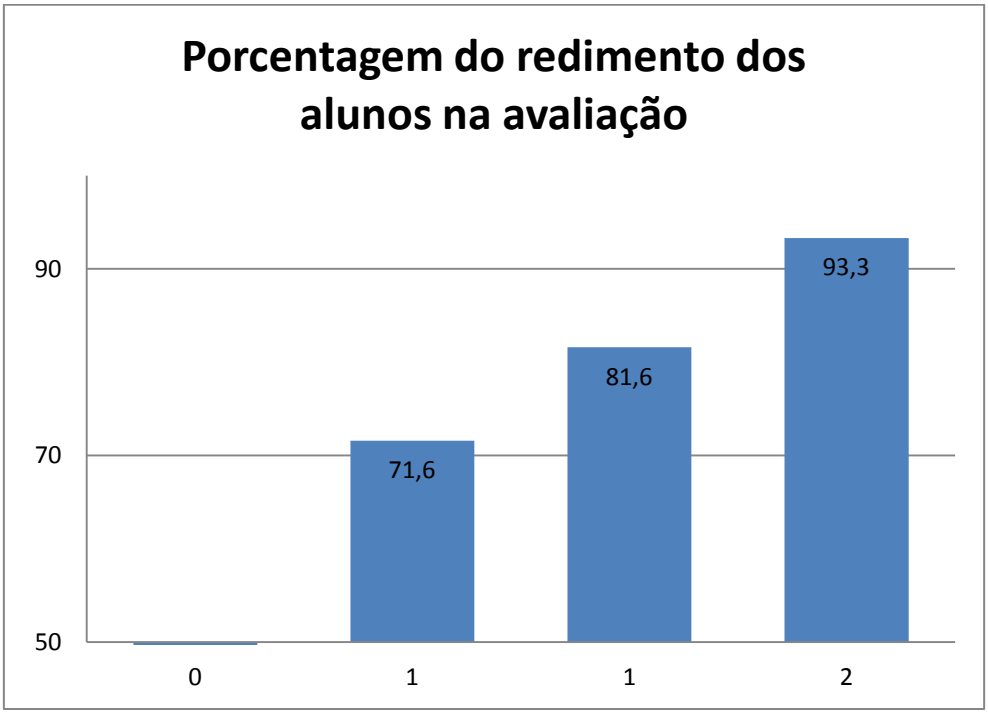


Figura 2 – Desempenho geral dos alunos na avaliação

Uma visão geral sobre o rendimento dos alunos na avaliação sobre programação orientada a objetos, eles deram conta de todo material trabalhado, tiveram bastante atenção nas aulas e faziam perguntas sobre o conteúdo trabalhado.

O fato de poderem colocar em prática tudo que viram na teoria, durante o mini curso, foi muito importante, pois os ajudou a entender melhor o conteúdo trabalhado. O Greenfoot promove rápido feedback, pois caso os alunos, ao editarem o código fonte do cenário realizassem algum erro, o cenário não compilava fazendo com que os alunos analisassem novamente o código da

classe trabalhada para localizarem o erro enfatizando desta forma seus aprendizados.

Podemos afirmar que, a partir da avaliação aplicada e o desempenho obtido dos alunos durante o curso, o objetivo desta pesquisa foi atingido, visto que com a utilização do software Greenfoot os alunos puderam aprender os conceitos básicos da programação orientada a objetos. No último dia do curso, os alunos foram incentivados a analisarem novamente os cenários sobre os quais trabalharam durante o curso, e promover algo a mais que achassem interessante e, se desejarem, deveriam postar tais cenários na galeria do Greenfoot para que sejam comentados e avaliados por outros usuários, favorecendo o aprendizado de novos conceitos que ainda não foram trabalhados.

6. Referências

AL-BOW et al. Using Greenfoot and Games to Teach Rising 9th and 10th Grade Novice Programmers. University of Denver.2008.

GALLANT Rand J. and MAHMOUD Qusa H. Using Greenfoot and a Moon Scenario to Teach Java Programming in CS1. University of Guelph-Humber, Toronto, ON, Canadá M9W 5L7. 2008.

KÖLLING Michael and HENRIKSEN Poul. Game Programming in Introductory Courses With Direct State Manipulation.2005.

SILVA FILHO, ANTONIO MENDES DA . Introdução à Programação Orientada a Objetos com C++, RJ.Editora: Elsevier, 2010.

VAHLDICK, A, MATTOS, M. M. Relato de uma Experiência no Ensino de Algoritmos e Programação Utilizando um Framework Lúdico. In: **XIX Simpósio Brasileiro de Informática na Educação**, 2008, Fortaleza. XIX Simpósio Brasileiro de Informática na Educação, 2008.

7. Anexo1- Modelo da Avaliação de Programação Orientada a Objetos

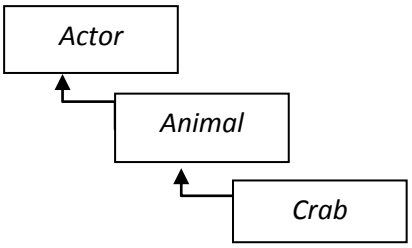
Curso Greenfoot - Introdução à Orientação a Objetos
Avaliação – 01.dez.2012

Nome: _____

1)Observe a instrução: ***void setDirection (int direction)***. Identifique, na ordem: a *assinatura* do método, o *tipo* de retorno, o *tipo* do parâmetro e o seu *nome,* o *nome* do método:

( )	setDirection (int direction)	int	void setDirection	Direction
( )	(int direction)	float	float Greenfoot	set
( )	Void	bool	bool void	void
( )	void setDirection (int direction)	void	int direction	setDirection
( )	Greenfoot	char	void move()	direction

2)Quando abrimos um projeto no software *Greenfoot*, um Diagrama de Classes sempre nos é apresentado. Veja o Diagrama abaixo e responda às questões em seguida:



O que é *Actor*?_____
O que é *Animal*?_____
O que é *Crab*?_____
Essa hierarquia denota um tipo de relacionamento. Como chamamos esse relacionamento em Orientação à Objetos? _____.

3)Observe a tabela abaixo. Na coluna da esquerda existem métodos e na coluna da direita as explicações para cada método da esquerda. Obviamente eles não estão em ordem. Relacione as duas colunas:

1	turn (n)	( )	Exemplo de assinatura de método.
2	void move ( )	( )	Método que retorna um valor inteiro.
3	boolean atWorldEdge()	( )	Espera um parâmetro do tipo inteiro.
4	int getLeavesEaten()	( )	Método que não retorna valor.

4)Sobre o método construtor é verdadeiro afirmar (pode ter mais de uma alternativa verdadeira):

( ) Um ***construtor*** não tem o tipo de retorno entre a palavra *public* e o seu nome.

( ) O nome do *construtor* é sempre igual ao nome da classe.
( ) Um *construtor* é um tipo especial de método que é executado sempre, automaticamente, que uma instância da classe for criada.

5)O método *addObject* é utilizado para adicionar um novo ator ao mundo. Ele é um método da classe *World* e sua assinatura é: *void addObject (Actor object, int x, int y)*

Assinale a alternativa que melhor explica a sua assinatura:

( )	void= o método retorna valor bool	*addObject*= nome da classe	*(Actor object, int x, int y)* = o método tem 2 parâmetros: eles são do tipo int.
( )	void= o método retorna valor int	*addObject*= nome da herança que pertence à classe World	*(Actor object, int x, int y)* = o método tem 3 parâmetros: O 1º é do tipo World, o 2º é float e o 3º é bool.
( )	void= o método não retorna valor	*addObject*= nome do método	*(Actor object, int x, int y)* = o método tem 3 parâmetros: O 1º é do tipo Actor. Os dois outros são int.
( )	void= a classe não retorna valor	*addObject*= nome do objeto	*(Actor object, int x, int y)* = o método tem 2 parâmetros: x e y.

6) **Métodos** com tipo de retorno diferente de *void* representam _____, ou seja, eles sempre nos dizem algo sobre o **objeto** (quantas folhas comeu, se pode andar, etc). **Métodos** com tipo de retorno *void* representam _____.
( ) objetos; laços.
( ) construtores; classes.
( ) objetos; instâncias.
( ) estruturas; condicionais.
( ) questões; comandos.

7) Considere as afirmações abaixo. Para cada afirmação coloque **V** (verdadeiro) ou **F** (falso):
( )Uma subclasse é uma classe que representa uma especialização de outra classe. Isto é representado com uma seta no Diagrama de Classes.
( )A especificação do método a qual mostra seu tipo de retorno, nome e parâmetros é chamado de **assinatura**.
( )No Greenfoot, toda a classe é definida por um **código-fonte**. Este código define o que os objetos desta classe podem fazer.
( ) Uma ***chamada de método*** é uma ***instrução*** que diz ao objeto realizar uma ***ação***. A ***ação*** é definida por um ***método*** do ***objeto***.

8) Observe o método setLocation(): public void **setLocation**(int x, int y). Este método atribui uma nova localização ao ator movendo-o para uma posição específica do mundo. Assinale a

alternativa que identifique, na ordem: os tipos dos parâmetros e seus nomes; e o tipo de retorno do método:

( )	int x	int y	void
( )	x	Y	float
( )	x	Y	void
( )	int y	int x	int
( )	int x	int y	public

9) Você precisa dimensionar o mundo para um cenário Greenfoot. O requisito é que o cenário tenha 200 por 300 células e cada uma delas tenha 5 pixels. Assinale a alternativa que expressa tal situação:

( )super(300, 200, 25);
( )super(200, 300, 5, 5);
( )super(300, 200, 5);
( )super(300, 200, 5, 5);
( )super(200, 300, 5);

10) Observe a tabela abaixo. Na coluna da esquerda existem métodos e na coluna da direita as explicações para cada método da esquerda. Obviamente eles não estão em ordem. Relacione as duas colunas:

1	getRandomNumber(n)	( )	Executa um arquivo de som.
2	isKeyDown("xxxx")	( )	Retorna true se o mouse foi clicado sobre um objeto.
3	playSound("xxxx.yyy")	( )	Gera um valor aleatório entre 0 e n-1.
4	mouseCliked(Aaaa.class)	( )	Verifica se uma determinada tecla foi pressionada.