MEDII VIZUALE DE DEZVOLTARE BAZATE PE PROGRAMAREA ORIENTATĂ PE OBIECTE

Adrian Runceanu, *Universitatea Constantin Brancusi, Targu Jiu, ROMANIA*
Mihaela Runceanu, *Colegiul National Ecaterina Teodoroiu, Targu Jiu, ROMANIA*

REZUMAT: In aceasta lucrare va prezentam variante alternative pentru dezvoltarea aplicatiilor avand ca baza limbajul Java. Doua dintre cele mai populare instrumente Java sunt Alice 3 si Greenfoot. Ambele aplicatii ofera medii de programare inovatoare și sunt utilizate pe scară largă în universități și licee din întreaga lume.
Alice 3 si Greenfoot ajuta utilizatorii sa invete programarea in Java si pot fi folosite cu succes si fara o dificultate deosebita in invatarea programarii orientate pe obiecte. Recomandam profesorilor sa foloseasca aceste instrumente Java pentru a preda programarea orientate pe obiecte.

CUVINTE CHEIE: Java, programare, clasa, Alice, Greenfoot, animatie

1. INTRODUCERE

Mediul de programare inovativ Alice 3 învață studentii să programeze folosind Java intr-un mod mai simplu, prin crearea de animații 3D, povești și jocuri video. Alice 3 include personaje, miscari ale corpului și alte elemente din Sims ™ - unul dintre cele mai bine vandute jocuri video pentru PC din toate timpurile.

VISUAL DEVELOPMENT ENVIRONMENTS BASED ON OBJECT-ORIENTED PROGRAMMING

Adrian Runceanu, *Constantin Brancusi University, Targu Jiu, ROMANIA*
Mihaela Runceanu, *The National College Ecaterina Teodoroiu, Targu Jiu, ROMANIA*

ABSTRACT: In this paper we present alternatives for developing applications based on Java, using two of the most popular Java tools Alice 3 and Greenfoot. Both applications provide innovative programming environment and are widely used in universities and high schools around the world.
Alice 3 and Greenfoot help users to learn programming in Java and can be used successfully and without excessive difficulty in learning object oriented programming. We strongly recommend teachers to use these two Java tools for teaching object-oriented programming.

KEY WORDS: Java, programming, class, Alice, Greenfoot, animation.

1. INTRODUCTION

The innovative programming environment teaches students to program with Alice and Java software as they have fun creating 3D animations, stories and video games.
Alice 3 incorporates characters, anatomical motions and other art assets from the Sims™ — one of the bestselling PC video games of all time.

Elementele Sims ™ transforma personajele și animațiile mai rudimentare din Alice 2.0 într-un conținut mai sofisticat pe care fiecare jucător joc il poate recunoaște.
Alice este orientat pe obiecte, un sistem open source dezvoltat în ultimul deceniu și a fost oferit gratuit profesorilor și studentilor de către Carnegie Mellon University. Alice dispune de o interfata drag-and-drop, care permite studentilor să creeze medii 3D și sa le populeze cu o mare varietate de obiecte și caractere ușor de programat.
Greenfoot este un mediu interactiv de dezvoltare Java, proiectat în principal pentru scopuri educaționale la nivel de liceu si universitar. Acesta permite dezvoltarea usoara a aplicatiilor grafice bidimensionale, cum ar fi simulări și jocuri interactive.
Greenfoot este dezvoltat de catre Universitatea din Kent și La Trobe University, cu sprijin din partea Oracle. Acesta este un software gratuit, lansat sub licenta GPL. Greenfoot este disponibil pentru Microsoft Windows, Mac OS X, Linux, Sun Solaris, precum și orice recente JVM

MEDII DE DEZVOLTARE VIZUALA – GREENFOOT AND ALICE 3

Java este un limbaj de programare si prima platforma de calcul lansata de Sun Microsystems în 1995. Aceasta este tehnologia de baza pentru progamele stat-of-the-art, incluzand utilități, jocuri și aplicații de afaceri. Java ruleaza pe mai mult de 850 de milioane de computere personale la nivel mondial, precum și cu miliarde de dispozitive mobile din întreaga lume, inclusiv și dispozitive TV.

Există o mulțime de aplicații și site-uri care nu vor funcționa dacă nu aveți instalat Java, și mai sunt create altele în fiecare zi. Java este rapid, sigur și fiabil. De la laptop-uri la centre de date, console de jocuri la supercalculatoare stiintifice, telefoane mobile la Internet, Java este peste tot!

The Sims™ assets transform the more rudimentary characters and animations of Alice 2.0 into sophisticated content that every game player can recognize.
Alice is an object-oriented, open source system developed over the last decade and provided free to educators and students by Carnegie Mellon University. It features a drag-and-drop interface that allows students to create 3D environments and populate them with a wide variety of easy-to-program objects and characters.
Greenfoot is an interactive Java development environment designed primarily for educational purposes at the high school and undergraduate level. It allows easy development of two-dimensional graphical applications, such as simulations and interactive games.
Greenfoot is being developed and maintained at the University of Kent and La Trobe University, with support from Oracle. It is free software, released under the GPL license. Greenfoot is available for Microsoft Windows, Mac OS X, Linux, Sun Solaris, and any recent JVM

VISUAL DEVELOPMENT ENVIRONMENTS – GREENFOOT AND ALICE 3

Java is a programming language and computing platform first released by Sun Microsystems in 1995. It is the underlying technology that powers state-of-the-art programs including utilities, games, and business applications. Java runs on more than 850 million personal computers worldwide, and on billions of devices worldwide, including mobile and TV devices.

There are lots of applications and websites that won't work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!

Sintaxa limbajului derivă mare parte din sintaxa C şi C + +, dar are mai puține facilități de nivel scăzut decât oricare dintre ele. Aplicațiile Java sunt, de obicei compilate în cod octet (fișier de clasă), care poate rula pe orice Java Virtual Machine (JVM), indiferent de arhitectura calculatorului. Java este cu scop general, concurent, pe bază de clase, limbaj orientat-obiect, care este special conceput pentru a avea putine dependențe de punere în aplicare. Acesta este destinat pentru a permite dezvoltatorilor de aplicații " write once, run anywhere " (WORA), ceea ce înseamnă că un cod care ruleaza pe o platforma nu trebuie să fie recompilat pentru a rula pe o alta platforma. Java este din 2012 unul dintre cele mai populare limbaje de programare utilizat, în special pentru aplicații web client-server, cu 10 milioane de utilizatori.

Va vom prezenta două dintre cele mai populare instrumente Java: Alice 3 și Greenfoot. Ambele dintre ele sunt folosite pentru a crea animație. Alice 3 este folosit pentru a crea animații 3D cu obiecte care se mișcă și acționează. Greenfoot este utilizat pentru crearea de jocuri interactive, care implică interacțiuni umane, decizii și acțiuni, dar cu obiecte 2D. Atât Alice si Greenfoot sunt instrumente foarte utile pentru a învăța Java.

De ce sa studiem Alice 3?

Alice 3 ofera o buna introducere pentru invatarea elementelor de programare pentru mai multe motive:

1. Foloseste cuvinte din limba engleza mai usor de folosit decat o sintaxa de programare obscura.
2. Programatorul trage si plaseaza obiectele pe ecran, apoi apasa "Run" pentru a fi executata animatia.
3. Atunci cand programatorul face o greseala in Alice 3, de obicei este foarte simplu sa o resolve. În limbajele de programare, este de multe ori dificil de a interpreta mesaje de eroare.
4. Alice 3 permite învațare de concepte fundamentale de programare, în contextul crearii de filme animate și jocuri video simple.

The language derives much of its syntax from C and C++, but has fewer low-level facilities than either of them. Java applications are typically compiled to byte code (class file) that can run on any Java Virtual Machine (JVM) regardless of computer architecture. Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java is as of 2012 one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users.

We will present you two of the most popular Java tools: Alice 3 and Greenfoot. Both of them are used to create animation. Alice 3 is used to create animations with 3D objects that move and act. Greenfoot is used for creating interactive games that involve human interactions, decisions, and actions. but with 2D objects. Both Alice and Greenfoot are very useful tools for learning Java.

Why Learn Alice 3?

Alice 3 offers a good introduction to learning how to program for many reasons:

1. It uses natural English language words like "move forward" or "turn left" rather than obscure programming syntax.
2. The programmer drag and drop objects on the screen and press "Run" to run your animation rather than using the keyboard to type.
3. When the programmer makes a mistake in Alice 3, it is usually obvious how to fix the mistake. In programming languages it is often difficult to interpret error messages.
4. Alice 3 allows you to learn fundamental programming concepts in the context of creating animated movies and simple video games.

5. Prin manipularea obiectelor intr-o lume virtuala, puteti castiga experienta cu multe dintre elementele de programare de obicei predate intr-un curs intensiv de programare.

De ce sa studiem Greenfoot?

1. Ne invata elementele de baza ale sintaxei Java si orientarea pe obiecte ceea ce face dezvoltarea de aplicații desktop Java mai ușoara decât pornind de la zero.
2. Interfata sa este un mediu de dezvoltare interactive (IDE) ceea ce permite editarea codului sursa, compilarea si depanarea exact ca in orice alt mediu interactiv Java.

Alice 3 and Greenfoot ajuta utilizatorii sa invete programarea in Java. Cu toate acestea, in scopul de a utiliza Alice 3 si Greenfoot, este nevoie sa dezvoltam anumite competente pentru crea animatii si jocuri. Iata o imagine de ansamblu la nivel înalt cu privire la pasii implicati în crearea unei animații sau unui joc:

1. Definirea unui scenariu
   – Care este scenariul (povestea)?
   – Ce obiecte sunt necesare?
   – Ce actiuni vor avea loc?
2. Proiectarea unui storyboard pentru scenariu – Vizual sau textual (sau ambele)
3. Crearea animatiei sau jocului (in Alice sau Greenfoot)
4. Testarea

Un **scenariu** contine trei parti:

• **Povestea:** Povestea care trebuie spusa, ori jocul care trebuie jucat.

• **Obiectele**: Obiectele pe care vor fi folosite in poveste.

• **Actiunile**: Toate actiunile pe care le vor efectua obiectele.

Procesul de dezvoltare a animatiei

5. By manipulating the objects in a virtual world, you can gain experience with many of the programming constructs typically taught in an introductory programming course.

Why Learn Greenfoot?

1. It teaches the basics of Java syntax and object orientation which makes developing desktop Java applications easier than starting from scratch.
2. Its interface is an interactive development environment (IDE) that allows you to edit source code, compile, and debug, just like in other Java IDE's.

Alice 3 and Greenfoot will help users learn to program in Java. In order to use Alice 3 and Greenfoot, however, we need to learn certain skills to create animations and games. Here is a high level overview of the steps involved in creating an animation or game:

1. Define a scenario
   – What story is to be told?
   – What objects are needed?
   – What actions will take place?
2. Design the storyboard for the scenario – Visual or textual (or both)
3. Create the animation or game (in Alice or Greenfoot)
4. Test

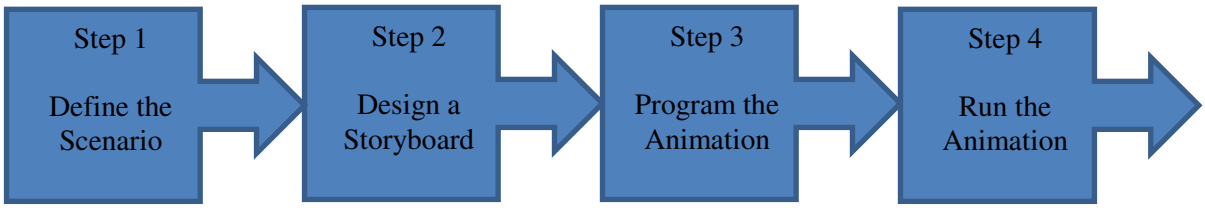
A **scenario** contains three parts:

• **Story:** The story to tell, or game to play. For example, a flying frog will catch flies and eat them.

• **Objects**: The objects you will use in your story. For example, a frog and flies.

• **Actions**: All the actions the objects will take.

Animation Development Process

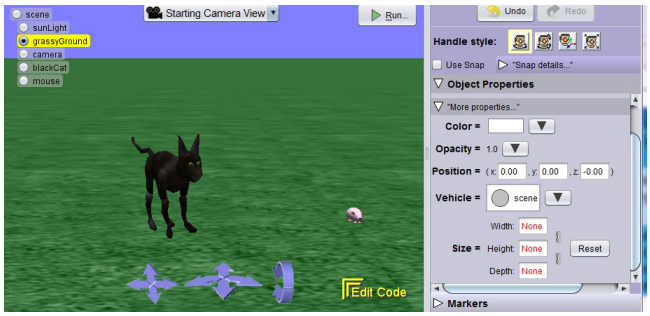


Cand lucram cu Alice 3, invatam cum sa programam fara a invata efectiv un limbaj de programare! Elementele pe care le trage si le plasam intr-un scenario reprezinta structuri logice, iar obiectele si actiunile in Alice 3 corespund limbajului de programare Java.

As we work with Alice 3, we learn how to program without learning an actual programming language! The tiles that we drag and drop represent logical structures and also the objects and actions in Alice 3 correspond to the Java programming language.

Alice versus Java

Alice 3	Java
Mediu de programare 3D **3D programming environment**	**Limbaj de programare; pot fi efectuate editari folosind un mediu de dezvoltare integrat (IDE).** **Programming language; can be edited using integrated development environment (IDE).**
Poate fi folosit pentru a crea animatii, jocuri sau filme interactive **Can be used to create animations, interactive games or videos**	**Poate fi folosit pentru crearea aplicatiilor care ruleaza pe orice platform, inclusiv pe web.** **Can be used to create applications that run on any platform, including the web.**
Operatiile de tip "drag and drop" reduc erorile de sintaxa **Drag and drop coding reduces syntax errors**	**Limbajul orientat pe obiecte reduce complexitatea deoarece modeleaza obiecte din lumea reala, este permisa reutilizarea lor si obiectele sunt mai usor de intretinut.** **Object oriented language reduces complexity because objects model real world objects, allow for re-use and easier maintenance.**



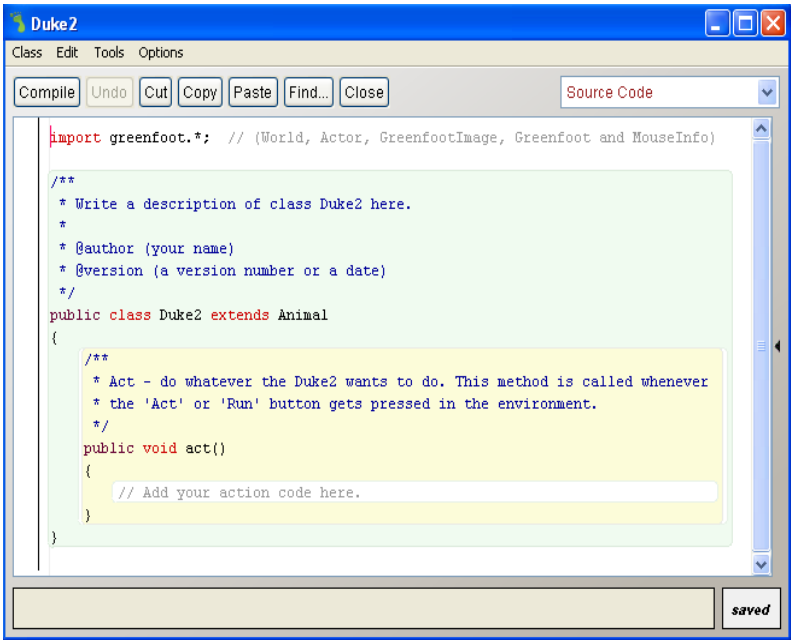

```
public class HelloWorld
{
        public static void main()
        {
        System.out.println( "Hello World!" );
        }
}
```

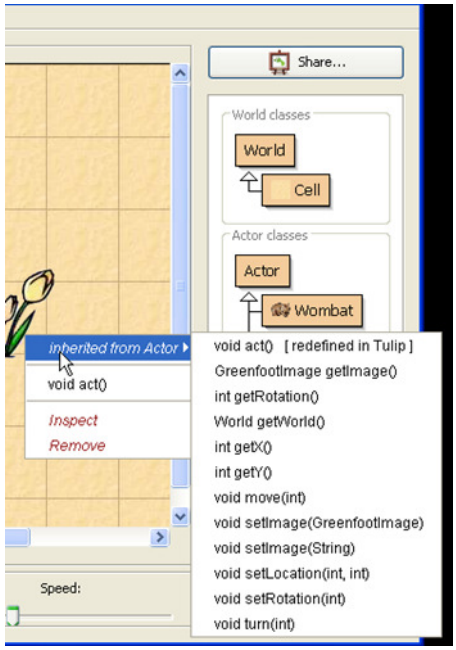
Greenfoot versus Java

Spre deosebire de Alice, care presupune foarte multa programare vizuala, Greenfoot inseamna mai mult scrierea de cod Java. Principalul instrument de programare in Greenfoot este editorul de cod (code editor). Editorul de cod afiseaza codul sursa pentru clasa. Codul sursa al clasei specifica toate proprietatile si caracteristicile clasei si ale obiectelor sale. Programatorul poate cere obiectelor din scenario sa efectueze actiuni sau sa raspunda unor cerinte prin scrierea de cod sursa in limbajul de programare Java. Cand selectam "Open Editor" din meniul clasei putem vedea fereastra de editare care contine codul sursa al clasei. Codul sursa afiseaza ce actiuni pot realiza obiectele clasei)

Greenfoot versus Java

Unlike Alice that means very much visual programming, Greenfoot means mostly writing code in Java. The main tool for programming in Greenfoot is the code editor. The code editor displays the source code for the class. The source code of a class is the code that specifies all of the properties and characteristics of that class and its objects. The programmer can command the objects in his scenario to perform tasks or answer questions by writing source code, or syntax, in the Java programming language. When selecting the Open Editor from the class's menu to see the editor window that contains the class's source code. The source code displayed defines what the objects of the class can do.





In editorul de cod, programatorul poate sa realizeze urmatoarele:

- Sa scrie cod sursa pentru a preciza cum vor actiona instantele clasei
- Sa revizuiasca metodele si proprietatile mostenite de clasa, sa inteleaga ce actiuni pot fi efectuate de catre instante
- Sa revizuiasca metodele create special pentru clasa de catre programatorul care a scris codul sursa
- Sa modifice codul sursa existent pentru a modifica comportamentul unei instante

In the code editor, the programmer can:

- Write source code to tell instances of the class how to act
- Review a class's inherited methods and properties, to understand what actions the instances are capable of taking
- Review methods created specifically for the class by the programmer who wrote the source code
- Modify existing source code to change an instance's behavior

Modelul de programare Greenfoot este format din clasa *World* (reprezentata printr-o zona de ecran dreptunghiulara) impreuna cu un numar de obiecte *actor* care sunt prezente in aceasta *lume* (*world*) si care pot fi programate sa actioneze independent. Lumea si actorii sunt reprezentati de obiectele Java si sunt definite de clasele Java. Greenfoot ofera metode de a programa usor acesti actori, incluzand metode pentru miscare, rotatie, schimbarea aspectului, detectarea coliziunii etc.

Programarea in Greenfoot are ca baza doua component, clasele *World* and *Actor* si se dezvolta subclase ale acestora. O instanta a subclasei *World* reprezinta mediul in care are loc actiunea. Subclasele *Actor* sunt obiecte care pot exista si actiona in acea lume. O instanta a subclasei *world* is create automat de catre mediul de programare.

Executarea in Greenfoot consta dintr-o bucla principala care apeleaza in mod repetat fiecare metoda de actiune a actorului. Programarea unui scenariu, prin urmare, constă în principal din punerea în aplicare a metodelor de actiune pentru actorii scenariului. Punerea în aplicare se face în standardul Java. Greenfoot oferă metode API pentru o gamă de sarcini comune, cum ar fi animatie, sunet, randomizare, precum și manipularea imaginii. Toate bibliotecile standard Java pot fi folosite la fel de bine, și pot fi atinse funcționalitati complexe.

Concepte Java importante

In Alice3

- Animatia obiectelor 3D are loc intr-o lume virtuala
- Galeriile contin obiecte 3D
- Obiectele se pot misca in 6 directii
- Un obiect are o orientare
- Centrul unui obiect este stabilit de creatorul obiectului.
- Un scenario descrie o animatie de ansamblu
- Un storyboard poate fi vizual sau textual

The Greenfoot programming model consists of a *World* class (represented by a rectangular screen area) and any number of *actor* objects that are present in the world and can be programmed to act independently. The world and actors are represented by Java objects and defined by Java classes. Greenfoot offers methods to easily program these actors, including method for movement, rotation, changes of appearance, collision detection, etc.

Programming in Greenfoot at its most basic consists of subclassing two built-in classes, *World* and *Actor*. An instance of the world subclass represents the world in which Greenfoot execution will occur. Actor subclasses are objects that can exist and act in the world. An instance of the world subclass is automatically created by the environment.

Execution in Greenfoot consists of a built-in main loop that repeatedly invokes each actor's *act* method. Programming a scenario, therefore, consists mainly of implementing *act* methods for the scenario's actors. Implementation is done in standard Java. Greenfoot offers API methods for a range of common tasks, such as animation, sound, randomisation, and image manipulation. All standard Java libraries can be used as well, and sophisticated functionality can be achieved.

Important Java Concepts

In Alice3

- Animation of 3D objects takes place in a virtual world
- Galleries contain 3D objects
- Objects can move in 6 directions
- An Object has an orientation
- An Object's center is set by the object creator
- A scenario describes an overall animation
- A storyboard can be visual or textual
- A program consists of lines of code
- Program code is structured in blocks (Do together)

- Un program este format din linii de cod
- Codul programului este structurat in blocuri
- Functiile furnizeaza proprietatile unui obiect
- Functile pot fi folosite pentru a calcula o valoare

In Greenfoot

- Scenariile sunt formate dintr-o multime de clase
- Obiectele sunt create dintr-o clasa
- Obiectele plasate intr-o lume sunt cunoscute sub denumirea de actori
- Obiectele au metode
- Metoda returneaza un tip care specifica ce va returna aceasta
- Void inseamna ca nu returneaza nimic
- Parametrii sunt folositi pentru a transmite date unei metode
- Parametrii au tipuri (int, boolean)
- Signatura este specificatia metodei
- O subclasa este o specializare a unei clase
- Fiecare clasa este definite de codul sursa
- Codul sursa necesita compilare
- Notatia cu punct este folosita pentru a apela o metoda static in alta clasa
- Metodele care apartin claselor folosesc cuvantul cheie static in signature lor.
- Metodele pot fi definite pentru o actiune noua.

Etapele de testare a software-ului sunt:

1. Analiza cerintelor – procesul de determinare a asteptarilor utilizatorului (listarea tipurilor de teste pentru a fi executate, stabilirea prioritatilor testelor, fezabilitate automata)
2. Planificarea testului – Scrierea unei strategii de planificare a testului; selectarea uneltelor de test; estimarea timpului necesar pentru testare; instruirea
3. Dezvoltarea situatiei de test – poate fi definit ca o multime de conditii sau variabile cu care un inginer de test va determina daca programul functioneaza corect.
4. Configurarea mediului – Configurarea mediului hardware si software

- Functions provide properties of an object
- Functions can be used to compute a value
- Boolean functions return true or false

In Greenfoot

- Scenarios consist of a set of classes
- Objects are created from a class
- Objects placed in a world are known as actors
- Objects have methods
- Method return types specify what a method will return
- Void return types return nothing
- Parameters are used to pass data to a method
- Parameters have types (int, boolean)
- Signatures are a specification of a method
- A subclass is a specialization of a class
- Every class is defined by source code
- Source code needs to be compiled (translated tomachine code)
- Dot notation is used to call a static method in another class
- Methods belonging to classes use the static keyword in their signature
- Methods can be defined for a new action

Software testing activities or stages include:

1. Requirement analysis – the process for determining the user expectations (listing the type of tests to perform, prioritizing and focusing tests, automation feasibility)
2. Test planning - Writing test plan strategy; selecting test tools; estimating time required for testing; training
3. Test case development - can be defined as a set of conditions or variables with

which a test engineer will determine if a software program is working correctly

4. **Environment setup** - Set up hardware and software environment

4. **Configurarea mediului** – Configurarea mediului hardware si software

5. Executarea testului – se defineste ca executarea aceluiasi test pe mai multe parti de program

6. **Testare cyclului de inchidere closure** - este un set de puncte de start și puncte pentru verificarea calității programului.

Utilizatorii pot gasi descrierea tuturor metodelor Java in on-line Java API. Intelegerea navigarii in aceasta vasta biblioteca de metode si clase standard va ajuta in scrierea programelor Java sis a reutilizati multe blocuri de cod care au fost deja create de altii. Aici puteti gasi Editia Standard pentru Java 6: **http://docs.oracle.com/javase/6/docs/api/**

CONCLUZIE

Utilizand un mediu de programare 3D, inovator, care face mai ușoara crearea de animații sau jocuri, proiectul Alice urmărește să ofere unelte și materiale pentru un nucleu conceptual al gândirii computationale, rezolvarea de probleme și programarea calculatorului.

Suita de instrumente educaționale Alice este conceputa pentru a sprijini predarea și învățarea la diferite nivele de varsta si de specializare, atat la nivel preuniversitar cat si la nivel de universitate.

Greenfoot își propune să motiveze persoanele care învață rapid prin asigurarea unui acces facil la grafică animata, sunet și interacțiune. Mediul este extrem de interactiv și încurajează explorarea și experimentarea. Pedagogic, proiectarea se bazează pe abordări constructiviste.

În al doilea rând, mediul este conceput pentru a ilustra și sublinia abstractizarile importante și conceptele de programare orientata pe obiecte. Concepte cum ar fi relația clasa / obiect, metodele, parametrii, și interacțiunea obiectelor sunt transmise prin intermediul vizualizării și a interacțiunilor ghidate. Scopul este de a construi și sprijini un model mental care reprezintă intr-un mod corect sistemele moderne de programare orientate pe obiect.

5. Test execution - is defined as the execution of the same test against many parts of a software program (interface, business logic, web layer, etc.)

6. **Test cycle closure** - is a set of defined start and stop points in a quality assurance program.

The users can find a description of all Java methods in the on-line Java API. Understanding how to navigate this vast library of standard methods and classes will aid you in writing Java programs and reusing many code blocks that have already been created by others. Using a browser search engine search for the keywords "Java API." You will find your way to several editions. Here is the Standard Edition for Java 6: **http://docs.oracle.com/javase/6/docs/api/**

CONCLUSION

Using an innovative 3D programming environment that makes it easy to create animations or games, the Alice Project seeks to provide tools and materials for a conceptual core of computational thinking, problem solving, and computer programming.

The Alice Suite of educational tools is designed to support teaching and learning across a spectrum of ages, grade levels, and classes in K-12 and in college or university courses.

Greenfoot aims to motivate learners quickly by providing easy access to animated graphics, sound and interaction. The environment is highly interactive and encourages exploration and experimentation. Pedagogically, the design is based on constructivist and apprenticeship approaches.

Secondly, the environment is designed to illustrate and emphasize important abstractions and concepts of object-oriented programming. Concepts such as the class/object relationship, methods, parameters, and object interaction are

conveyed through visualizations and guided interactions. The goal is to build and support a mental model that correctly represents modern object-oriented programming systems.

BIBLIOGRAFIE

[1] Ficher, S., Kölling, M., Utting, I., Brown, N., Stevens, P., 2010, Repositories of Teaching Material and Communities Of Use: Nifty Assignments and the Greenroom, Proceedings of the Sixth international workshop on Computing education research (ACM SIGCSE): 107–114.
[2] Gaddis, T., Starting Out with Alice: A Visual Introduction to Programming, 2007, Pearson Addison Wesley
[3] Henriksen, P., Kölling, M., McCall, D., 2010, Motivating Programmers Via An Online Community, Journal of Computing Sciences in Colleges (Association for Computing Machinery) 25 (3): 82–93.
[4] Herbert C. W., An Introduction to Programming Using Alice, 2009
[5] Kölling, M., 2010, The Greenfoot Programming Environment, ACM Transactions on Computing Education (TOCE), Vol. 10, No. 4, Article 14, Pub. date: November 2010.
[6] Kölling, M., 2010, Introduction to Programming with Greenfoot - Object-Oriented Programming in Java with Games and Simulations, Pearson Education.
[7] Shelly, G.B., Cashman T. J., Herbert, C.W., 2009, Alice 2.0: Introductory Concepts and Techniques
[8] Utting, I., Cooper, S., Kölling, M,. Maloney, J., Resnick, M., Alice, 2010, Greenfoot, and Scratch - A Discussion, ACM Transactions on Computing Education (TOCE) Vol. 10, No. 4, Article 17
[9] http://en.wikipedia.org/wiki/Greenfoot
[10] http://en.wikipedia.org/wiki/Alice_%28software%29

REFERENCES

[1] Ficher, S., Kölling, M., Utting, I., Brown, N., Stevens, P., 2010, Repositories of Teaching Material and Communities Of Use: Nifty Assignments and the Greenroom, *Proceedings of the Sixth international workshop on Computing education research* (ACM SIGCSE): 107–114.
[2] Gaddis, T., Starting Out with Alice: A Visual Introduction to Programming, 2007, Pearson Addison Wesley
[3] Henriksen, P., Kölling, M., McCall, D., 2010, Motivating Programmers Via An Online Community, *Journal of Computing Sciences in Colleges* (Association for Computing Machinery) **25** (3): 82–93.
[4] Herbert C. W., An Introduction to Programming Using Alice, 2009
[5] Kölling, M., 2010, The Greenfoot Programming Environment, *ACM Transactions on Computing Education (TOCE),* Vol. 10, No. 4, Article 14, Pub. date: November 2010.
[6] Kölling, M., 2010, Introduction to Programming with Greenfoot - Object-Oriented Programming in Java with Games and Simulations, Pearson Education.
[7] Shelly, G.B., Cashman T. J., Herbert, C.W., 2009, Alice 2.0: Introductory Concepts and Techniques
[8] Utting, I., Cooper, S., Kölling, M,. Maloney, J., Resnick, M., Alice, 2010, Greenfoot, and Scratch - A Discussion, *ACM Transactions on Computing Education (TOCE)* Vol. 10, No. 4, Article 17
[9] http://en.wikipedia.org/wiki/Greenfoot
[10] http://en.wikipedia.org/wiki/Alice_%28software%29