

«Низький старт» в навчанні програмування студентів комп'ютерних спеціальностей

Навчання основ програмування, введення основних понять (алгоритм, змінна, типи даних і т.д.) і концепцій – важливі проблеми навчання студентів перших курсів комп'ютерних спеціальностей. Для їх розв'язання на спеціальності «Інформатика» Кримського інженерно-педагогічного університету крім власне дисципліни «Програмування» була введена дисципліна «Програмування для початківців». Доповнюючи і спираючись одна на одну, ці дві дисципліни утворюють платформу для формування системи компетентностей, необхідних програмісту, що дозволяє студентам просуватися від рівня до рівня, від простих завдань до складних проектів.

Курс «Програмування для початківців» включає вивчення середовищ програмування Alice і Scratch, розроблених спеціально для вирішення зазначених проблем. Дослідження основних особливостей мов програмування у цих середовищах і вплив їх вивчення на першому курсі комп'ютерних дисциплін у зв'язку з навчанням складних мов програмування (наприклад C++) є значущою педагогічною проблемою.

Alice – це безкоштовна версія програмного продукту об'єктно-орієнтованого напрямку з інтегрованим середовищем розробки. Являє собою середовище програмування для створення комп'ютерних анімацій з використанням 3D-моделей. Вперше дане програмне забезпечення було розроблено на початку 90-х років минулого століття в університеті Вірджинії, а потім доопрацьоване дослідницькою групою університету Карнегі-Меллона. Перші версії Alice були реалізовані за допомогою мови Python, а більш пізні версії реалізовані за допомогою мови Java. Як зазначено в [1], Alice була розроблена для розв'язання наступних проблем:

- подальшого навчання програмування зі складною семантикою, зокрема, C++. Користувач створює віртуальний світ, розміщує в ньому об'єкти, розробляє програмний код, використовуючи вбудовані конструкції;
- підтримки об'єктно-орієнтованого підходу до програмування, управління моделями даних;
- підготовки користувачів до подальшого професійного програмування, а також використання в навчальному процесі багатьох коледжів і університетів світу.

На даний момент існують наступні версії Alice:

- Alice 2.0, Alice 2.2
- Storytelling Alice
- Alice 3.0

Версії Alice 2.0 і 2.2 призначені для навчання інформатики в середній школі та коледжах. Storytelling Alice спрямована на використання в навчальному процесі молодшої школи. Storytelling Alice була створена Кейтлін Келлехер, в рамках докторської дисертації в галузі комп'ютерних наук в Університеті Карнегі-Меллона. Відмінною особливістю даної версії від попередньої, є впровадження та використання додаткових функцій, зокрема, об'єктам бібліотеки Storytelling Alice можна надавати більш широкий спектр «поведінки» і «емоцій». Також у цій версії проект можна подати в кількох сценах, а в Alice 2.0-2.2 в рамках роботи можна створити віртуальний світ і тільки одну сцену. Alice 3.0 в даний час розробляється командою Alice та Electronic Arts – американською корпорацією, розробником і дистриб'ютором комп'ютерних відеоігор. На даний момент Alice 3.0 представлена тільки в бета-версії. У даній версії студенти працюватимуть у середовищі типу The Sims 2. Середовище типу The Sims 2 – це відеогра, в якій користувач управляє одним або кількома віртуальними об'єктами. The Sims була розроблена компанією Maxis і видана Electronic Arts. Об'єктами галереї Alice 3.0 можна буде управляти за принципом The Sims 2, тобто вибирати об'єкт в режимі віртуального світу і створювати подобу реального життя в ігровому сюжеті.

Об'єкти в Alice існують в тривимірному віртуальному світі, так само, як сучасні відеоігри. Alice, як і багато сучасних об'єктно-орієнтованих мов програмування, зокрема C++, мають характерну для них граматику і синтаксис запису, та побудована таким чином, що студенти мають можливість самостійно створювати методи (шляхом з'єднання кольорових плиток). Це дозволяє студентам зосередитися на вивченні ідей об'єктно-орієнтованого програмування. Віртуальний світ Alice – це той, який студенти можуть побачити. Як і в реальному світі, заселені об'єкти мають властивості, такі як: колір, розмір, місце розташування, напрям руху об'єкта. Тобто, використовуючи Alice на заняттях, студенти можуть втілити свої абстрактні ідеї в програмний код і спостерігати за наслідками своєї роботи.

Розробники програмного продукту Alice стверджують ([1], [2], [3]), що на практиці необхідно використовувати всі версії даного продукту, незважаючи на те, що кожна версія Alice спрямована на конкретну вікову групу. Тому що Alice – це новий і потужний інструмент, за допомогою якого можна

залучити студентів до програмування, познайомити з поняттями алгоритма, типами алгоритмів. Використання Alice на заняттях також допоможе створити сприятливе соціальне середовище в робочій аудиторії.

Щоб розпочати роботу з Alice, не вимагається спеціальної установки. Досить завантажити ZIP-файл, що містить середовище Alice, і далі розархівувати файли з архіву і помістити в локальний каталог. Подальший доступ до Інтернету не потрібен. Бібліотека 3D об'єктів зберігається локально, в галереї. Але з метою економії простору локального диска класи для багатьох інших об'єктів не включені до місцевої бібліотеки, вони можуть бути доступні з веб-версією класу галереї.

Scratch – це візуальна об'єктно-орієнтована мова програмування, в якій підтримуються 50 мов інтерфейсу, у тому числі і російська. Вперше програмне забезпечення Scratch з'явилося в 2003 році, і було представлено бета-версією Scratch 0,1. Сьогодні актуальна і використовується на практиці версія Scratch 1.4. У лютому 2011 була випущена перша онлайнова бета-версія Scratch 2.0. У таблиці 1 наведена коротка характеристика всіх існуючих версій Scratch.

Таблиця 1.

Версія Scratch	Рік створення	Характеристика
Scratch 0,1	2003р.	Перша бета – версія.
Scratch 1.0	8 січня 2007р.	Перша офіційна версія Scratch для відкритого доступу.
Scratch 1.1	середина 2007р.	<ul style="list-style-type: none"> • додана функція «Імпорт спрайту»; • використання конструкції Repeat Until; • підключення звуку (Play Sound); • використання команд «Костюм», «Звук», «Трансляція» без урахування регістру; • додавання кнопки перемикання мови; • кожен вбудований спрайт має конкретний вбудований звуковий супровід.
Scratch 1.2-1.2.1	7 грудня 2007р.	<ul style="list-style-type: none"> • у режимі презентації (запуску роботи) приховані спрайти не блокують поточні; • визначено правильний переклад для ключових імен; • при запуску проекту імена спрайтів і костюмів відображаються поточною мовою; • стиснення графічних зображень; • виправлена підказка-повідомлення при використанні команди «Вилучити»; • остання реалізація для Windows 98 і ME.
Scratch 1.3-1.3.1	2 вересня 2008р.	<ul style="list-style-type: none"> • підтримка 42 національних мов; • можливість приховувати змінні; • додані списки, коментарі; • форматування шрифтів в рамках робочого проекту; • повністю переглянутий блок графіки.
Scratch 1.4	19 липня 2009р.	<ul style="list-style-type: none"> • додано новий блок для опрацювання рядків; • додано Wait блок; • використання веб-камери; • повністю змінено інтерфейс програми; • сумісність з Windows 7.
онлайнова бета-версія Scratch 2.0	лютий 2011р.	<ul style="list-style-type: none"> • планується включення деяких можливостей, таких як створення користувацьких процедур, векторної графіки, клонування спрайтів, можливість групової роботи над проектами.

Дане середовище програмування являє собою програмну оболонку і призначене для створення анімаційних та напівавтоматично виконуваних проектів: комп'ютерних ігор, мультиплікаційних роликів. Scratch спрямований на використання в навчальному процесі, починаючи з восьми років, за його допомогою можна ознайомити користувачів з азами програмування, вивчити основні принципи об'єктно-орієнтованого програмування. Навчання програмування в середовищі Scratch присвячені роботи зарубіжних дослідників, таких як М. Резнік, Дж. Мелоні, А. Монрой-Ернандес, К. Бреннан [6], а також дослідження вітчизняних вчених Е.О. Єрьоміна [7], О.М. Петрової [8]. Всі перелічені вище дослідники приходять до єдиної думки, що використання в навчальному процесі Scratch не тільки покаже і пояснить роботу різних операторів, циклів, умов, але й навчить мислити особливим чином,

розуміти призначення команд. Середовище Scratch виступає як інструмент для творчості в процесі програмування. Пов'язуючи з об'єктами певні дії в програмі Scratch, студент не знайомиться, а вивчає основні фундаментальні поняття циклів та умовних конструкцій.

Alice і Scratch позиціонуються як засоби навчання програмування, а тому в них використовуються його основні принципи та конструкції. Незважаючи на те, що кінцевим продуктом є мультиплікація, створюється вона за допомогою послідовності чітких інструкцій, подібних до тих, що визначені в повноцінних мовах програмування, таких як C++.

В обох середовищах підтримуються імперативна і об'єктно-орієнтована парадигми: з одного боку програма являє собою набір команд, які виконуються послідовно одна за одною, з іншого боку програма являє собою опис властивостей і «поведінки» окремих об'єктів, які можуть «взаємодіяти» між собою. Об'єктами є персонажі мультиплікаційних роликів, камера і джерело світла, для маніпулювання якими користувачеві доступні їх властивості та методи.

Для зберігання значень логічних і арифметичних операцій в Alice і Scratch вводиться поняття змінної, яка характеризується типом: числова, булева або рядкова. Після надання імені змінній їй можна називати значення:

set < ім'я змінної > to < значення > (Scratch)

< ім'я змінної > set value to < значення > (Alice)

Більше того, спираючись на досвід студентів з розробки програм в Scratch, можна ввести поняття області видимості змінних в C++. При створенні змінної в цьому середовищі користувач вибирає один з двох варіантів видимості: у поточному спрайту (шару мультиплікації) або для всіх спрайтів.

У таблиці 2 наводиться відповідність конструкцій управління в програмах, описаних мовою C++, чи розроблених в середовищах Alice або Scratch. У таблиці можна побачити, що в обох розглянутих середовищах наявні основні конструкції управління, а їх синтаксис інтуїтивно зрозумілий і в деяких випадках схожий з синтаксисом мови C ++, яка вивчається студентами з першого курсу.

Таблиця 2.

Конструкції управління	C++	Alice	Scratch
Розгалуження	if (<i>expression</i>) <i>instruction 1</i> ; else <i>instruction 2</i> ;	if (<i>expression</i>) <i>instruction 1</i> else <i>instruction 2</i>	if (<i>expression</i>) <i>instruction 1</i> else <i>instruction 2</i>
Перемикання	switch (<i>expression</i>){ case a1: <i>instruction1</i> ; case a2: <i>instruction1</i> ; ... };	відсутнє	відсутнє
Цикл параметром	for (<i>exp.1;exp.2;exp3</i>){ <i>instruction</i> ; }	loop (<i>exp.1;exp.2;exp3</i>) <i>instruction</i>	repeat (<i>counter</i>) <i>instruction</i>
Цикл передумовою	while (<i>expression</i>) { <i>instruction</i> ; }	while (<i>expression</i>) <i>instruction</i>	repeat until (<i>expression</i>) <i>instruction</i> або forever if (<i>expression</i>) <i>instruction</i>
Цикл післяумовою	do{ <i>instruction</i> ; } while(<i>expression</i>);	відсутнє	відсутнє

Одна з ситуацій, на яку слід звернути увагу студентів при вивченні розгалуження – необхідність виконання деяких інструкцій при істинності заданої умови і відсутність будь-яких дій в іншому випадку. У мові C++ в цьому випадку розгалуження буде виглядати наступним чином:

if (*expression*) *instruction*;

У Scratch для таких завдань передбачена аналогічна конструкція. На місці умовного позначення *expression* розміщується логічний вираз, для побудови якого, як і в будь-якій мові програмування, доступні різні логічні операції.

Таким чином, виконання подібних завдань в цьому середовищі може створити основу для вивчення кодування різних варіантів розгалуження мовою C++. У Alice такий запис відсутній і в будь-якому випадку доведеться записати:

```
if (expression)
    instruction
```

```
else
```

де після службового слова else просто немає ніяких команд (DoNothing).

Організація циклу в Alice здійснюється за допомогою однієї з двох операцій: loop або while (таблиця 2). Семантика першої з них відповідає циклу з параметром в мові C++ і передбачає різні варіанти використання. У найпростішому випадку користувач задає кількість ітерацій:

```
loop 5 times
    instruction
```

В такому разі буде змінюватися значення параметра циклу, починаючи з нуля і збільшуючись на 1 до тих пір, поки операції в тілі циклу не будуть виконані вказану кількість разів. В іншому випадку користувач може задавати довільне початкове значення параметра циклу, кінцеве значення і крок зміни параметра.

Що стосується Scratch, до категорії циклу з передумовою можна віднести дві конструкції мови: forever if (expression) або repeat until (expression). В обох випадках у дужках розташовується логічний вираз (таблиця 1). Команди, про розташовані у тілі циклу, будуть виконуватися до тих пір, поки цей логічний вираз набуде значення *істинно*.

Слід зазначити, що в конструкції forever може бути відсутнім задання умови, що дозволяє побудувати нескінченний цикл так само, як і конструкція while (true) {...} у мові C++.

Важливим моментом з точки зору використання Alice і Scratch для навчання основ програмування є наявність готових шаблонів конструкцій управління використанням програми, які можна додавати до своєї програми у вікно редактора переміщенням. Отже, виключається можливість появи в програмі синтаксичних помилок. Таким чином, на початковому етапі навчання програмування немає необхідності запам'ятовувати синтаксис безлічі команд. Тут увагу зосереджено на розумінні правил і логіки їх виконання, вмінні визначати, в якій ситуації яка з доступних конструкцій є найбільш придатною. Одна зі складностей, помічених нами при навчанні програмування студентів першого курсу, – правильна побудова розгалуження або циклу залежно від поставленого завдання так, щоб код був якомога більш простим з точки зору читання, розуміння та виконання.

Слід відзначити і візуальне подання коду програми у середовищах Alice і Scratch. Наприклад, в Scratch шаблон кожної мовної конструкції візуально схожий на елемент пазла або конструктора «Лего» (рис. 1). Таким чином, готова програма створює враження правильно складених частин одного цілого.

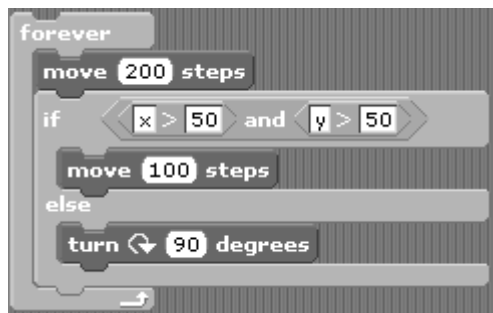


Рис. 1

У статті «Scratch – програмування для всіх» [6] розробники цього середовища посилаються на слова Сеймура Паперта (автора книги про середовище програмування Logo) про те, що мова програмування повинна відповідати трьом критеріям «low floor + high ceiling + wide walls». Це означає, що мова програмування повинна бути досить простою для початківців (low floor), досить потужною для складних проектів (high ceiling) і відповідати різноманіттю інтересів розробників (wide walls). Саме з метою забезпечення ще більш простого старту (low floor), відзначають автори Scratch [6], була запозичена ідея конструктора «Лего», де можна збирати різні фігурки з окремих блоків, з'єднуючи їх між собою. А потім з цих блоків можна зібрати єдиний цікавий проект.

Слід відзначити також автоматичне дотримання відступів рядків коду, що позначають вкладеність мовних конструкцій. Надалі при описуванні програми будь якою мовою програмування грамотне оформлення коду стає звичним і само собою зрозумілим.

Крім навичок побудови алгоритму виконання завдання і його кодування якою-небудь мовою програмування важливим для майбутнього програміста є вміння працювати в команді. З цієї точки зору Scratch є простим і захоплюючим способом навчити цього студентів першого курсу. Окремі проекти, створені за допомогою Scratch, можна імпортувати в один загальний проект, отримавши в результаті більш складну анімацію. Кожен імпортований проект в цьому випадку «лягає» окремим

шаром на загальну сцену і виконується одночасно з іншими. Таким чином, студенти можуть у командному режимі розробляти складні цікаві анімації або ігри, розподіляючи між собою частини одного спільного проекту. Після імпортування скрипт кожного проекту все ще доступний для редагування і в разі необхідності може бути приведений у відповідність з іншими скриптами проекту.

Вивчення Scratch може серйозно допомогти студентам освоїти ази алгоритмізації і програмування, а отримані знання стануть в нагоді для подальшого і більш серйозного вивчення програмування. Студенти першого курсу спеціальності «Інформатика», використовуючи Scratch на заняттях, створюють проекти, мультиплікаційні ролики, комп'ютерні ігри. Scratch допомагає студентам виробити глибокий рівень свободи в інформаційних технологіях, а також виразити себе більш повно і творчо. Програмування за допомогою Scratch допоможе їм розвинути логічне мислення і зрозуміти принципи функціонування і використання нових технологій, які вони зустрічають у повсякденному житті.

Слід зазначити, що інтерес студентів до конкретної галузі розробки проектів сприяє формуванню позитивної мотивації до вивчення програмування, зокрема опису програм мовою C++. При використанні курсу «Програмування для початківців» до організації процесу навчання програмування в цілому досліджуваний матеріал ефективно сприймається і засвоюється студентами, активізується їх пізнавальна діяльність, розвиваються творчі здібності. Все це створює сприятливу обстановку для підвищення якості отриманих знань, умінь і навичок в галузі програмування.

Література

1. Сайт проекта Alice [Електроний ресурс]. – Режим доступу: <http://www.alice.org>
2. Professor Developed Alice Animation Tool to Teach Computer Programming [Електроний ресурс]. – Режим доступу: <http://www.webwire.com/ViewPressRel.asp>
3. Dann W.P. Learning To Program with Alice / W.P. Dann, S. Cooper, R. Pausch.- 2nd edition. – Prentice Hall, 2009. – 384 p.
4. Сайт проекта Scratch [Електроний ресурс]. – Режим доступу: <http://www.scratch.mit.edu>
5. Сайт «Учитесь со Scratch» [Електроний ресурс]. – Режим доступу: <http://setilab.ru/scratch/category/commun/>
6. Resnick M. Scratch: Programming for All / M. Resnick, J. Maloney, A. Monroy-Hernandez, K. Brennan. – Communications of the ACM. – 2009. – [Електроний ресурс]. – Режим доступу: <http://scratchrus.wordpress.com/2011/>
7. Еремін Е.А. Среда Scratch – первое знакомство / Е.А. Еремін // Информатика. – М.: Первое сентября, 2008. – №20 (573). – С.16–28.
8. Петрова О. М. З досвіду використання об'єктно-орієнтованого середовища програмування Scratch / О.М. Петрова // Комп'ютер у школі та сім'ї. – 2009р. – №6. – С.25-27