

Введение в ООП: использование специальных программных сред

С.М. Сейдаметова, Ф.В. Шкарбан

Крымский инженерно-педагогический университет, г. Симферополь, Крым

Paper received 01.06.15; Accepted for publication 16.09.15.

Аннотация. В статье отражена необходимость использования на практике специальных программных сред для успешного освоения программирования в целом и ООП в частности.

Ключевые слова: программирование, объектно-ориентированное программирование программное обеспечение, инженер-программист, языки программирования

Введение. В последнее время высшее образование России подвергается изменениям, которые способствуют повышению качества подготовки студентов, адаптации образования к быстро меняющимся требованиям ИТ-рынка труда. В связи с этим, специалистами обсуждаются вопросы, касающиеся содержания образовательной программы по предмету «Информатика». Это привело к созданию и внедрению образовательных стандартов, отражающих понимание предмета «Информатика» и соответствующих ему знаний, умений и навыков.

Краткий обзор публикаций по теме. Среди мировых проектов, предлагаемых по преподаванию информатики, можно выделить проект Computing Curricula. Данный стандарт отражает современные подходы по преподаванию информатики, разработанные в рамках совместного проекта профессиональных обществ – программистов ACM и IEEE Computer Society [1]. Рассматриваемый проект содержит ряд рекомендаций для университетских программ в области информатики, где для каждой области определен минимальный уровень обязательных знаний, имеется и факультативный материал.

Язык программирования является для программиста основой его деятельности. Однако, знаний одного языка недостаточно, специалист должен понимать различные парадигмы программирования. В СС2008 область «языки программирования» содержит одиннадцать разделов, среди которых шесть являются фундаментальными. Для изучения обязательного раздела «Объектно-ориентированное программирование» (ООП) выделяется половина времени отведенного на изучение всей области «Языки программирования» [2, с. 65].

Цель статьи. Показать, что применение на практике специальных программных сред необходимо для успешного освоения программирования в целом и ООП в частности.

Материалы и методы. Основная задача обучения программированию - развитие логического мышления и воображения, алгоритмизация задач. Программирование – это процесс создания компьютерных программ с помощью определенных языков программирования (разработка программного обеспечения). Под понятием «программирование» следует понимать сочетание элементов искусства, науки, математики и инженерии [3]. Искусство программирования включает в себя знания о программных инструментах и языках программирования, навыков по решению проблем в области программирования, а также эффективных стратегиях для разработки и реализации компьютерных программ.

Программирование является одним из основных навыков, которые должны приобрести в процессе обу-

чения инженеры-программисты. Традиционное преподавание программирования рассматривает, как правило, сложные задачи и вопросы, что приводит к дальнейшему непониманию данного направления студентами. Методология и инструменты, используемые традиционными методами обучения программированию, основанные на лекциях и конкретном синтаксисе языка программирования, не представляют интереса у современных студентов или являются непонятным в освоении. Языки программирования при обучении программированию на практике обычно используются из класса профессионального характера, в частности это С, С++, С# и Java. Перечисленные языки программирования имеют довольно объемный синтаксис записи, что делает обучение трудным для начинающего «студента-программиста». Студентам на практике необходимо представить и помочь увидеть:

- процесс проектирования и решения проблемы;
- разделение программы на более простой код;
- представление гипотетической ситуации – ошибки;
- проведение тестирования и выявления ошибок;
- выявление проблем даже в самых основных понятиях (например, переменные, типы данных или адреса памяти и т.д.).

Поэтому для преодоления трудностей, связанных с преподаванием программирования на практике зарубежные педагоги-практики предложили и начали использовать специальные программы, которые позволяют студентам «наблюдать» весь свой процесс обучения. Один из самых успешных программных пакетов программного обеспечения для моделирования и визуализации основных процессов, используемых в процессе обучения программированию, был проект Karel the robot [4]. Данный проект представляет собой программу, основанную на принципах LOGO-языка (объект-робот выполняет последовательность команд, заданных пользователем). Вскоре «компьютерный мир» перешел на С программирование, а затем остановился на использовании объектно-ориентированных языков программирования (ООП). ООП широко используется в сфере образования и промышленности, и почти каждый университет ведет обучение студентов компьютерных специальностей в рамках объектно-ориентированного подхода.

Знакомство с объектно-ориентированным программированием (используя конкретное программное обеспечение) является мотивирующим фактором при изучении дальнейшего программирования. Это осуществляется в ходе работы, в которой студенты, используя определенный программный продукт, непосредственно на практике, формируют представление о ООП, оценивают его преимущества.

В раздел ООП включены следующие темы [2, с. 66]: объектно-ориентированное проектирование, инкапсуляция и сокрытие информации, классы и подклассы, наследование, полиморфизм, иерархия классов, внутреннее представление объектов и таблиц.


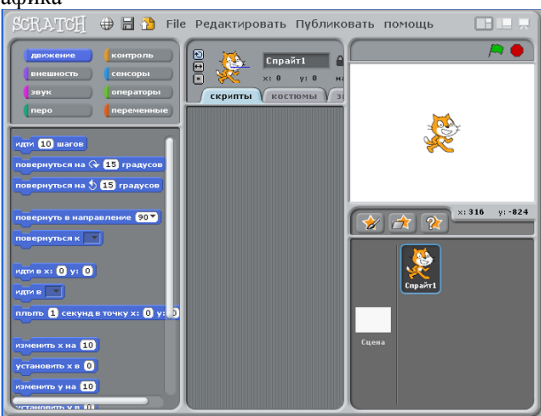

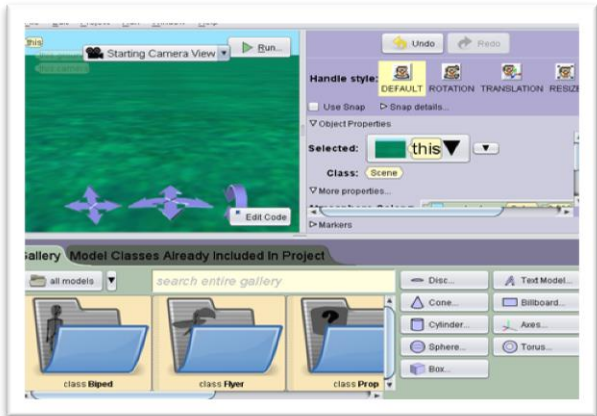
Для изучения ООП используется модель реального мира, где есть какие-либо объекты, обладающие свойствами и методами (поведением). У каждого объекта есть два обязательных метода: конструктор и деструктор (объект «рождается и умирает») и текущее положение. Объекты с одинаковыми свойствами и поведением группируются в классы. Объект, принадлежащий некоторому классу, называется экземпляром этого класса.


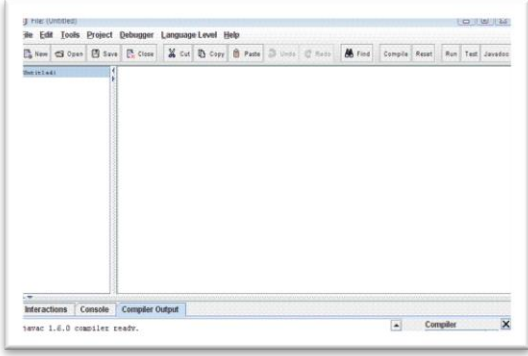

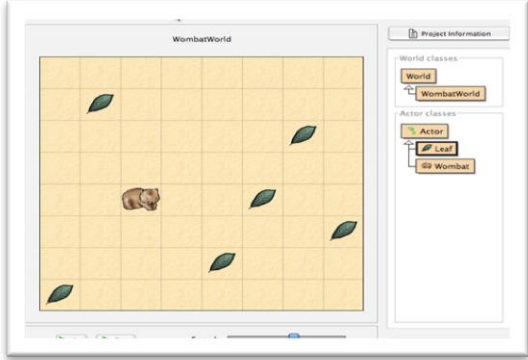

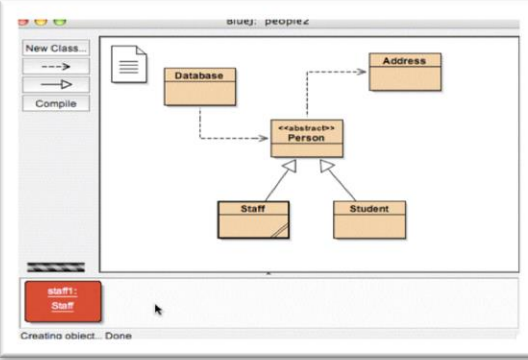

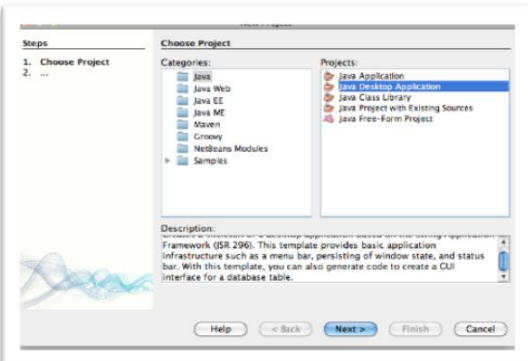
Результаты и обсуждение. Сегодня зарубежные университеты мира используют для обучения программированию программные среды для генерирования конкретных визуальных представлений программы, в частности, Alice [5], BlueJ [4] и т. д. Эти программные средства при обучении программированию используются для усиления мотивационных аспектов курса. Инструменты анимации, визуальные подсказки, звуки и интерактивность поддерживают активность студентов во время занятий. Однако, исследования в области психологии программирования показывают, что программирование требует большой практики, а понимание сути программы студенты получают именно в процессе обучения [6].


На первом курсе будущие инженеры-программисты работают в инструментальной учебной среде «Alice», которая представляет собой 3D интерактивную, анимационную, программную среду для создания виртуальных миров. Студент добавляет 3D объекты, указывая при этом расположение каждого из них в виртуальном мире. Каждый объект имеет свои собственные данные и методы. Студент может выбрать для объекта примитивные методы, которые посылают сообщения объекту, либо может написать свои алгоритмы и функции. При этом студент знакомится с понятиями «объекты» и «инкапсуляция».

Для обучения программированию в области ООП существуют специальные программные среды, являющиеся интегрированной средой разработки с простым интерфейсом. Они позволяют наглядно представлять модель создаваемой программы, выполнять методы классов объектов в процессе их написания, что осуществляется созданием экземпляра класса объектов с последующим вызовом его метода или непосредственным выполнением, тестировать классы объектов, задавая различные параметры для методов по мере их написания.

В таблице 1 представлена характеристика наиболее используемых программных сред, которые способствуют дальнейшему обучению и развитию профессионального программирования.

Инструментарий	Особенности	Концепции обучения
	<p><i>Простейший инструмент программирования</i></p> <ul style="list-style-type: none"> – Перетягивание блоков программного кода – 2D графика 	<ul style="list-style-type: none"> - последовательность итераций; - условная логика; - переменные; - структуры данных (динамические списки); - обработка событий; - параллельное выполнение; - синхронизация; - случайные числа; - булева логика; - динамическое взаимодействие; - пользовательский интерфейс; - публикация проектов.
	<p><i>Простейший инструмент программирования</i></p> <ul style="list-style-type: none"> – Перетягивание блоков программного кода. – 3D графика. 	<ul style="list-style-type: none"> - все концепции Scratch; - процедуры и функции; - передача параметров и возвращаемых значений; - рекурсия; - определение классов объектов; - наследование; - ввод текста.

Инструментарий	Особенности	Концепции обучения
	<p><i>Менее простой инструмент программирования</i></p> <ul style="list-style-type: none"> – Интерактивный ввод кода – Классы Media вычислений 	<ul style="list-style-type: none"> - все предыдущие концепции Alice без учета дизайна пользовательского интерфейса; - манипуляция аудио, видео изображения, видео через классы Media вычислений; - все доступные функции Java.
	<p><i>Менее простой инструмент программирования</i></p> <ul style="list-style-type: none"> – Тип, компиляция, запуск, отладка; – Игровая основа программирования с помощью классов Java. 	<ul style="list-style-type: none"> - все предыдущие концепции Alice ...; - классы Java; - инкапсуляция; - понятие игры и симуляции; - все доступные функции Java.
 <p>BlueJ</p>	<p><i>Менее простой инструмент программирования</i></p> <ul style="list-style-type: none"> – Тип, компиляция, запуск, отладка – Нет стандартного графического окружения 	<p>Java программирование:</p> <ul style="list-style-type: none"> - язык; -особенности программирования
	<p><i>Комплекс инструментов программирования</i></p> <ul style="list-style-type: none"> – Тип, компиляция, запуск, отладка – Нет стандартного графического окружения 	<p>Несколько языков программирования:</p> <ul style="list-style-type: none"> - Java - Ruby - Groovy - Python - PHP - JavaScript)

Инструментарий	Особенности	Концепции обучения
	<p><i>Любой инструмент</i></p> <ul style="list-style-type: none"> - Тип, компиляция, запуск, отладка - D графика - структура - Нет стандартного графического окружения 	<p>Язык программирования общий со средами:</p> <ul style="list-style-type: none"> - Alice - Dr. Java - Greenfoot - BlueJ - NetBeans

Использование рассмотренных выше программных сред для обучения программированию не помешает, поскольку это дает возможность перехода курса программирования на более современный уровень и расширяет возможности будущих профильных курсов, а также устраняет различия между современным состоянием программирования как науки и содержанием преподавания этой дисциплины в высших учебных заведениях. Использование специальных программных сред позволяет изучать общепринятые конструкции программирования в разрезе их дальнейшего использования в объектно-ориентированном программировании. Это качественная практическая поддержка систем задач и упражнений, решение которых позволяет студентам охватить все основные нюансы современных методов разработки программного кода.

Итак, можно сделать вывод, что используя при обучении программированию специальное программное обеспечение можно решить ряд задач:

- формирование объектно-ориентированного стиля мышления;
- ознакомление с объектами и построение их иерархии;
- выявление событий, связанных с объектами;
- реализация методов классов объектов;
- анализ работы программы.

Рассмотрим принципы использования среды «Alice» в процессе знакомства с объектно-ориентированным программированием.

Alice – это свободный и открытый объектно-ориентированный язык программирования, направленный для обучения программированию, с интегрированной средой разработки (IDE), что позволяет создавать трехмерную анимацию. Она реализована средствами Java и Python. Alice использует методы drag-and-drop для создания компьютерной анимации с использованием 3D-моделей [7].

Мир программной среды Alice состоит из объектов, а объекты в свою очередь делятся на составные объекта. В процессе работы студент собирает из представленных блоков трехмерный мир, который «засе-

ляют» объекты (животные, люди, растения). Alice состоит из функций, переменных, параметров и рекурсий, которые записываются в единую программу. Это позволяет студентам участвовать и в то же время иметь возможность развивать интуитивное понимание основных концепций программирования. Таким образом, студенты, которые используют «Alice» сразу могут увидеть, как их анимационные программы работают и сделанные ими действия, понять фактическое функционирование различных программ.

Таким образом, если в процессе обучения программированию использовать вспомогательные программные среды, в частности «Alice», то это позволит изучить на практике преимущества объектно-ориентированного программирования, что соответствует современным требованиям подготовки специалистов.

Выводы. В заключении следует отметить, что интерес студентов к использованию конкретного программного обеспечения способствует формированию положительной мотивации к изучению программирования. При использовании сред в процессе обучения программированию, исследуемый материал эффективно воспринимается и усваивается студентами, активизируется их познавательная деятельность, развиваются творческие способности. Все это создает благоприятную ситуацию для повышения качества приобретенных знаний, умений и навыков в области программирования.

Изучаемые дисциплины, предшествующие ООП позволяют получить необходимые знания, на основе которых строится ООП. Объектно-ориентированный подход в программировании содержит ряд плюсов, без которых не обойтись при написании кода программы: разбиение сложной задачи на более мелкие части, расширяемость системы, многократное использование кода и т.д. Данная дисциплина, являясь завершающей в области «Языки программирования» позволяет закрепить полученные знания, совершенствовать умения и навыки в области программирования.

REFERENCES

- [1] Association for Computing Machinery: Advancing Computing as a Science & Profession / URL: <http://www.acm.org/>
- [2] Computer Science Curriculum 2008 (CS 2008): An Interim Revision of CS 2001 – ACM, IEEE-CS, December 2008. - 108 p.
- [3] Stroustrup, B. Programming: Principles and Practices of Use C++, revised edition = Programming: Principles and Practice Using C++ / B. Stroustrup – M.: Williams, 2011. – 1248 p. – ISBN 978-5-8459-1705-8
- [4] Improving teaching and learning of computer programming through the use of the Second Life virtual world / URL: http://home.utad.pt/leonelm/papers/BJET_Micaela.pdf
- [5] Dann, W.P. Learning To Program with Alice / W.P Dann, S. Cooper, R. Pausch.- 2nd edition. - Prentice Hall, 2009. – 384 p.
- [6] Fincher, S. What are We Doing When We Teach Programming? / URL: http://kar.kent.ac.uk/21750/1/what_are_we_doing_when_we_teach_programming_fincher.pdf
- [7] Alice / URL: – <http://www.alice.org>

Introduction into object-oriented programming: use of special programming environments

S.M. Seydametova, F.V. Shkarban

Abstract. The article deals with the importance of using special programming environments in practice for successful mastering of programming in general and object-oriented programming in particular.

Keywords: programming, object-oriented programming, the engineer-programmer, software, programming languages