



**РАЗРАБОТКА СИСТЕМЫ ЛАБОРАТОРНЫХ РАБОТ
ДЛЯ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ**

Нежинский Виталий Владимирович

*магистр Российского государственного
педагогического университета им. А.И. Герцена
191186, РФ, г. Санкт-Петербург, набережная реки Мойки, д. 48
E-mail: SeekerOfConfessor@gmail.com*

Шумара Елена Викторовна

*канд. пед. наук, доцент Российского государственного
педагогического университета им. А.И. Герцена
191186, РФ, г. Санкт-Петербург, набережная реки Мойки, д. 48
E-mail: math_rgpu@mail.ru*

**DEVELOPMENT OF THE SYSTEM
OF LABORATORY WORKS ON PROGRAMMING**

Vitalij Nezhinskij

*Master degree student of Herzen State Pedagogical University,
191186, Russia, Saint-Petersburg, Moika Embankment, 48*

Elena Shumara

*Candidate of Pedagogic Sciences,
associate professor of Herzen State Pedagogical University,
191186, Russia, Saint-Petersburg, Moika Embankment, 48*

АННОТАЦИЯ

В статье представляется и анализируется опыт создания системы лабораторных работ для обучения студентов высших учебных заведений современному программированию. Обосновывается выбор парадигмы программирования для создания дидактических материалов, а именно объектно-ориентированной парадигмы, языка программирования,

реализующего данную парадигму (Java), а также, на основе анализа федеральных государственных образовательных стандартов, выбор специальности «Прикладная математика и информатика» в качестве адресной. Анализируется традиционный лекционно-практический метод обучения программированию, выделяются его основные недостатки и обосновывается решение создать новый методический продукт в виде системы лабораторных работ. Предлагаются принципы, в соответствии с которыми строятся лабораторные работы, в частности, принципы практической ориентированности, непосредственной обратной связи, последовательности изложения материала, внутренних связей между различными блоками содержания. Описывается структура разработанной системы лабораторных работ для тематического планирования на 32 аудиторных часа: девять лабораторных работ (приводится краткое содержание), три проверочные работы (четыре варианта в каждой работе) и проекты четырех уровней сложности (приводится пример для каждого уровня сложности). Указывается, за счет каких средств достигаются промежуточный, а также итоговый контроль. Делается важный вывод о том, что разработанная система лабораторных работ (с учетом измененного содержания) может быть эффективной в рамках изучения любого языка программирования, базирующегося на объектно-ориентированной парадигме.

ABSTRACT

In this article, we introduce experience of creating system laboratory works for Computer Science education to students. The paradigm of programming is selected for creating teaching materials. We select objective-oriented paradigm, language of programming (Java) which realize this paradigm. In addition, we choose the specialization Applied mathematics and Computer Science according to analyzing federal state educational standards. We consider disadvantages of the lecture-practical methods and solve to create new methodic product, which based on laboratory works. We identify the following principles: principle of practical oriented, principle of feedback, sequence of presentation and principal of internal

communications between different blocks of content. The structure of this systems of laboratory work consist of nine laboratory works, three verification work with four variants and projects with different types of difficulties. In addition, we present summary of laboratory works. Thematic planning is 32 hours. Intermediate and final controls are realized by tasks after laboratory works, verification works and project. Conclusion of the article is that system of laboratory works with some changes will be efficient on studying any language of any programming language, which based on objective-oriented paradigm.

Ключевые слова: система лабораторных работ, программирование, Java, объектно-ориентированная парадигма

Keywords: system of laboratory works, Computer Science, Java, object-oriented paradigm

Бурное развитие информационного общества, а также регулярное обновление средств информационных технологий диктует в наши дни новые требования к обучению по соответствующим направлениям и специальностям. Свежее программное обеспечение, в частности, версии языков и сред программирования появляется практически каждый год. Книги и учебные пособия, которые были актуальны еще не так давно, сейчас практически теряют свой смысл и не находят применения у конечных пользователей. В этих условиях приходится искать новые подходы к обучению программированию, нацеленные не на изучение конкретных языков, а на овладение технологиями создания программ в рамках различных парадигм. Мы предлагаем такой подход, который позволит студентам овладеть основами императивного и объектно-ориентированного программирования, научиться грамотно применять базовые структуры данных и алгоритмы для решения возникающих в их деятельности задач (на примере языка программирования Java). В будущем, когда студенты столкнутся с необходимостью изучения другого языка, им придется освоить лишь синтаксис и конкретные технологии среды.

Итак, прежде всего нам необходимо было выбрать парадигму программирования, в рамках которой наиболее целесообразно разрабатывать соответствующие дидактические материалы. Выбор пал на объектно-ориентированную парадигму, и это не случайно. Благодаря четырем принципам объектно-ориентированного программирования (абстракции, наследованию, полиморфизму и инкапсуляции) программа становится более структурированной, защищенной и простой в дальнейшей поддержке. Именно поэтому указанная парадигма является наиболее популярной у разработчиков программного обеспечения.

Затем мы рассмотрели федеральные государственные образовательные стандарты (ФГОС) [4] и выбрали в качестве адресной специальности «Прикладная математика и информатика». Студенты, обучающиеся по этой специальности, имеют хорошую математическую подготовку и интерес к программированию. Содержание их обучения предоставляет возможности для глубокого изучения программирования, и их будущая профессиональная деятельность напрямую связана с программированием.

На следующем этапе мы проанализировали традиционный лекционно-практический метод обучения программированию, выделили его достоинства и недостатки. К наиболее существенным недостаткам мы отнесли следующие: многие теоретические моменты остаются неясными для студентов из-за отсутствия непосредственной обратной связи с преподавателем; теория и практика не всегда соответствует друг другу; нередко прочитанный материал не находит должного применения в практической работе; иногда теоретические моменты, которые нужны для решения задач, сообщаются на лекциях слишком поздно, и в результате на практических занятиях приходится тратить время на теорию; мотивация студентов нередко оказывается недостаточной. Опираясь на сделанные выводы, мы решили создать методический продукт (систему лабораторных работ), который решал бы задачу обучения программированию более успешно, реализовывал бы требования ФГОС и был направлен

на взаимодействие преподавателя и студента, в котором обратной связи было бы больше.

Лабораторные работы мы решили строить в соответствии со следующими принципами:

- теоретический материал, как правило, не выделяется в качестве отдельного блока, а органичным образом сочетается с практической деятельностью;

- задачи и проблемы, которые формулируются в начале каждой темы, мотивируют студентов к изучению теоретического материала;

- деятельность студентов в рамках лабораторных работ практико-ориентирована;

- в процессе учебной деятельности реализуется обратная связь между преподавателем и студентами; преподаватель ставит перед студентами проблему, мотивирует их на решение этих проблем и в дальнейшем корректирует их деятельность;

- в процессе выполнения каждой лабораторной работы студентам предлагаются упражнения и задачи разной сложности; они могут служить для наилучшего осознания теоретического материала, отработки и закрепления учебных навыков, самостоятельной работы под руководством преподавателя, контроля;

- каждая лабораторная работа является логическим продолжением предыдущей;

- материал, как теоретический, так и практический, подается от простого к сложному;

- в начале обучения формируется представление о конечной цели – создании проекта; студенты понимают, что им предстоит последовательно, шагом за шагом продвигаться к его реализации.

Важной задачей стал выбор языка программирования, для которого целесообразно было бы реализовать конкретный экземпляр такой системы. Требования к языку были выдвинуты следующие: он должен быть легко

изучаемым с точки зрения синтаксиса, поскольку на базовом этапе важно не столько изучение непосредственно языка, сколько понимание структур данных, базовых алгоритмических конструкций, основных принципов объектно-ориентированного программирования; он должен быть популярным и решать широкий класс задач. В результате исследований, в частности, благодаря анализу ресурсов GitHub [10], Jobs Tractor [11] и TIOBE Index [12] был выбран язык программирования Java.

Созданная нами система состоит из компонентов трех типов.

I. Непосредственно лабораторные работы, которые охватывают следующее содержание.

1. Введение в программирование на языке Java. Примитивные типы данных. Операции над ними. Объявление переменных. Операция ввода-вывода.

2. Явное и неявное преобразование типов. Классы обертки.

3. Массивы и строки. Ссылочные типы.

4. Условные операторы: if – else и switch – case.

5. Циклы с предусловием, циклы с постусловием, циклы со счетчиком.

6. Объектно-ориентированное программирование. Основные принципы и концепции. Классы, объекты. Модификаторы доступа. Конструктор (создание объекта), деструктор (удаление объекта).

7. Наследование. Интерфейсы, абстрактные классы. Переопределение методов. Принцип замещения Барбары Лисков.

8. Исключение. Иерархия исключений.

9. Потоки ввода-вывода. Запись в файл и чтение из файла.

Лабораторные работы включают в себя изложение теоретического материала, вспомогательные упражнения и задачи (мотивирующие задачи; задачи, подводящие под понятие; задачи для самостоятельного решения). После каждой лабораторной работы предлагается задание для самостоятельного ознакомления с теорией, практические задания для самостоятельного выполнения разных уровней сложности (от простого к сложному) и методические указания, способствующие реализации проектов.

II. Проверочные работы, в каждой из которых предлагается четыре варианта.

Первая проверочная работа посвящена структурному программированию (работе с типами данных, операторами условного перехода, циклами).

Вторая проверочная работа включает себя задачи на объектно-ориентированное программирование.

Третья проверочная работа направлена на проверку усвоения тем, посвященных исключениям и работе с файлами.

III. Проекты разных уровней сложности.

1) Базовый уровень – проект «Телефонная книга».

2) Продвинутый уровень (по возрастанию уровня сложности):

- Проект «Строчный калькулятор»;

- Проект «Чат (клиент-сервер)» для обмена сообщений между пользователями с последующей записью сообщений в базу данных.

- Проект «Программа для тестирования», которая может стать полезным рабочим инструментом в работе педагога.

Виды контроля предполагаются следующие: промежуточный контроль, который достигается за счет задач, которые студенты решают самостоятельно, после чего демонстрируют свои решения преподавателю; проверочные работы; проектная деятельность. По окончании курса выставляется зачет на основании выполнения трех перечисленных пунктов.

В качестве основных литературных источников использовались при подготовке и предлагались студентам базовые книги по Java [7; 8], а также лекции Massachusetts Institute of Technology (MIT) [9]. В качестве дополнительных источников – документация Oracle [5] и другие источники [1; 2; 3; 6].

Предложенная система, состоящая из девяти лабораторных работ, трех проверочных работ и одного проекта, стала методической основой для изучения дисциплины «Языки и методы программирования» (32 аудиторных часа) и показала свою эффективность при применении ее для обучения студентов бакалавриата математического факультета РГПУ им. А.И. Герцена

по направлению «Прикладная математика и информатика» современному программированию.

Используя структуру разработанной нами системы, преподаватели могут создать собственные системы лабораторных работ для любого языка программирования, базирующегося на объектно-ориентированной парадигме.

Список литературы:

1. Бутч Г., Энгл М., Янг Б., Коналлен Д., Хьюстон К. Объектно-ориентированный анализ и проектирование с примерами приложений. 3-е изд. – Москва: Вильямс, 2010. – 720 с.
2. Дейт Кр. Введение в системы баз данных. – Москва: Вильямс, 2006. – 1328 с.
3. Лафоре Р. Структуры данных и алгоритмы Java. – СПб: Питер, 2016. – 704 с.
4. Министерство образования и науки. – [СПб] / [Электронный ресурс]. – Режим доступа: URL: минобрнауки.рф/документы/336 (дата обращения: 01.04.2016).
5. Официальная документация компании “Oracle” [Электронный ресурс]. – Режим доступа: URL: www.oracle.com (дата обращения: 30.04.2016).
6. Сьера К., Бейтс Б. Изучаем Java. – Москва: Эксмо, 2012 – 720 с.
7. Шилдт Г., Java. руководство для начинающих. 6-е издание, Москва: Вильямс, 2015.—712 с.
8. Эккель Б., Философия Java.Библиотека программиста. 4-е издание.— Спб: Питер, 2014. – 640 с.
9. E. Jones, O. Koch, P. Cudre-Mauroux, Introduction to programming in Java. Lectures 2009 / [Электронный ресурс]. – Режим доступа: URL: <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/> (дата обращения: 30.04.2016).

10. Github. A small place to discover languages in github / [Электронный ресурс]. – Режим доступа: URL: <http://github.info/> (дата обращения: 01.04.2016).
11. Jobstractor.com It finds developer jobs on Twitter, and displays the ones local to you. / [Электронный ресурс]. – Режим доступа: URL: <http://jobstractor.com/monthly-stats> (дата обращения: 01.04.2016).
12. TIOBE Index for April 2016 / [Электронный ресурс]. – Режим доступа: URL: http://www.tiobe.com/tiobe_index (дата обращения: 01.04.2016).

References:

1. Booch G., M. Engle, B. Young, J. Conallen, K. Houston Object-Oriented Analysis and Design with Applications. Moscow, Williams Publ., 2010. 720 p. (In Russian).
2. C. Date. An Introduction to Database Systems. Moscow, Williams Publ., 2006. 1328 p. (In Russian).
3. R. Lafore. Data Structures and Algorithms in Java. St. Petersburg, Piter Publ., 2016. 704 p. (In Russian).
4. The Ministry of education and science. Available at: минобрнауки.рф/документы/336 (accessed 01.04.2016).
5. Official documentation of company “Oracle”. Available at: www.oracle.com (accessed 30.04.2016).
6. K. Sierra, B. Bates. Head first Java. Moscow, Eksmo Publ., 2012, 720 p. (In Russian).
7. H. Schildt, Java: A Beginner's Guide, 6th edition, Moscow: Williams, 2015.— 712 с. (In Russian).
8. B. Eckel, Thinking in Java. Library of programming. St. Petersburg, Piter Publ., 2014. 640 p. (In Russian).
9. E. Jones, O. Koch, P. Cudre-Mauroux, Introduction to programming in Java. Lectures 2009. Available at: <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/> (accessed 30.04.2016).

10. Githut. A small place to discover languages in github. Available at: <http://githut.info/> (accessed 01.04.2016).
11. Jobstractor.com It finds developer jobs on Twitter, and displays the ones local to you. Available at: <http://jobstractor.com/monthly-stats> (accessed 01.04.2016).
12. TIOBE Index for April 2016. Available at: http://www.tiobe.com/tiobe_index (accessed 01.04.2016).