

ВЗАИМОСВЯЗАННОЕ ОБУЧЕНИЕ КАК ОДИН ИЗ ПЕРСПЕКТИВНЫХ ПОДХОДОВ К ОБУЧЕНИЮ СТУДЕНТОВ ТЕХНОЛОГИЯМ ПРОГРАММИРОВАНИЯ

*Статья рекомендована к публикации научным руководителем,
заведующим кафедрой прикладной математики и информатики,
кандидатом педагогических наук, доцентом Зенько С.И.*

Аннотация. Проанализированы подходы к обучению студентов технологиям программирования в высших учебных заведениях. Обосновывается целесообразность реализации обучения студентов на основе взаимосвязанного обучения, приводится перечень дидактических материалов, разработанных с учетом особенностей данного подхода и внедренных в учебный процесс. Описаны преимущества взаимосвязанного обучения будущих учителей информатики технологиям программирования.

Ключевые слова: взаимосвязанное обучение, обучение студентов технологиям программирования, взаимосвязанное обучение технологиям программирования, последовательное обучение, параллельное обучение.

В настоящее время в литературе наиболее часто встречаются два подхода обучения студентов технологиям программирования – подход последовательного обучения и подход параллельного обучения.

Последовательно обучение технологиям программирования предполагает учет принципа историчности и изучение технологий в соответствии с эволюцией языков программирования и развитием парадигм программирования. Согласование этих идей с опорой на компетенции студентов, полученные ими в процессе обучения в школе, позволяет преподавателям структурировать содержание процесса обучения технологиям программирования в виде последовательности, состоящей из двух относительно завершенных и целенаправленно невзаимосвязанных блоков обучения. Первый блок предполагает изучение идей только структурного программирования (включая изучение методов алгоритмизации). Второй блок – идеи парадигмы объектно-ориентированного программирования. Последовательное обучение вышеуказанным технологиям программирования предполагает, что студенты сначала учатся создавать функции и процедуры в рамках структурного программирования, а затем определяют классы объектов и

устанавливают отношения между ними (Н.А. Мещерякова, Ю.А. Петрова и др.). Это приводит к тому, что студенты рассматривают класс объектов не как абстрактный тип данных, а как некий набор функций и процедур. Однако технология объектно-ориентированного программирования не является расширением структурного программирования, а представляет самостоятельную технологию со своей уникальной идеологией, которая диктует свои приемы программирования. Поэтому необходимо дополнительно рассматривать специальную подготовку студентов по этим вопросам.

При последовательном обучении технологиям программирования также возникает потребность в дополнительных временных затратах на «адаптационный период», в течение которого происходит приобретение теоретических знаний и соответствующих компетенций, которые являются отличительными для технологии объектно-ориентированного программирования. Как отмечает Б. Страуструп, у среднего программиста либо студента, изучающего объектно-ориентированное программирование на основе структурного программирования, на данный адаптационный период может потребоваться от 6 до 13 месяцев [3]. Это в свою очередь приводит к противоречию между необходимостью в изучении различных технологий программирования, что связано с существенным увеличением временных затрат, с одной стороны, и стремительным сокращением жизненного цикла приобретаемого запаса знаний для практики при обучении в вузе, с другой

А.И. Газейкина, С.А. Егорова, В.В. Пенкрат придерживаются идеи параллельного изучения нескольких языков программирования в процессе обучения программированию студентов. А.И. Газейкина отдает предпочтение только объектно-ориентированным языкам программирования, и на первом курсе ею предлагается изучение языков Object Pascal и Java, на втором дополнительно – C, C++ [1]. В результате проведенного анализа работ автора можно отметить недостаточное уделение внимания такому принципу обучения как преемственность. Существует множество работ в сфере методики преподавания, которые подчеркивают важность и острую необходимость соблюдения принципа преемственности именно на этапе школа–вуз. А.П. Сманцер, Т.А. Ханалыев указывают на необходимость целостности процесса обучения, которая требует на каждой ступени наличие системы преемственно взаимосвязанных между собой целей-задач, которые должны стать целями обучающихся [2]. В.В. Пенкрат реализует идею параллельного преподавания алгоритмического языка ИнтАл и Pascal, С.А. Егорова предлагает параллельное изучение языков C++ и Pascal. Параллельное изучение языков

программирования позволяет использовать разный инструментарий при решении одних и тех же задач, однако параллельность не предполагает одновременное использование вышеуказанного инструментария (это связано с особенностями построения процесса обучения разным языкам программирования).

Проанализировав два подхода к обучению студентов технологиям программирования, можно утверждать, что каждый из них имеет свои достоинства и недостатки. Однако, требования к подготовке будущего учителя информатики в современных условиях вызывают необходимость поиска новых подходов к обучению, позволяющих рационально, сбалансировано и результативно осуществлять учебный процесс в вузе. Поэтому считаем перспективным рассмотрение взаимосвязанного обучения студентов технологиям программирования.

Под взаимосвязанным обучением технологиям программирования будем понимать учебно-преподавательскую деятельность (студента и преподавателя), направленную на одновременное формирование учебных умений студентов в процессе согласованного изучения ими двух или более технологий программирования, при котором каждая из технологий программирования является средством и целью обучения. Сущность технологии программирования как средства обучения реализуется через один из языков программирования, поддерживающий данную технологию.

Учебные умения студентов предполагают приобретение ими опыта осознанного самостоятельного выполнения действий (эффективное управление своей учебной деятельностью, самостоятельное использование оптимальных методов, форм и средств приобретения профессионально-педагогических знаний, умений и навыков, владение рациональными способами решения предметных задач, развитость способности анализировать и оценивать собственную деятельность), прочность которых базируется на сформированности таких компонентов учебной деятельности студентов как мотивация, целеполагание, планирование, реализация, самоконтроль и оценка.

В учебный процесс подготовки будущего учителя информатики по специальности 1-02 05 01 «Математика и информатика» при изучении учебной дисциплины «Технологии программирования и методы алгоритмизации» на физико-математическом факультете учреждения образования «Белорусский государственный педагогический университет имени Максима Танка» внедрены дидактические материалы, разработанные с учетом особенностей

реализации взаимосвязанного обучения технологиям программирования. Они включают:

- банк образцов установленной формы (условие задачи, её формализация, графическое представление решения в виде блок-схемы, псевдокод, реализация алгоритма решения задачи на языках программирования PascalABC.NET и C#);
- учебно-методическое пособие в электронном виде, включающее систему индивидуальных заданий по всем темам, предусмотренным учебной программой учебной дисциплины «Технологии программирования и методы алгоритмизации» для студентов первого курса;
- комплекс интерактивных приложений для сопровождения лекций с помощью интерактивной доски;
- электронную систему диагностики готовности студентов к применению теоретического материала на практике – в процессе выполнения лабораторных работ.

Взаимосвязанное обучение технологиям программирования позволяет:

- расширить систему учебных задач, основанную на взаимосвязанном использовании языков программирования, поддерживающих идеи разных технологий программирования;
- существенно уменьшить временные затраты на изучение различных технологий программирования;
- оптимизировать обучающую деятельность студентов;
- использовать средства (одинаковые или близкие по характеру) различных языков программирования при решении одних и тех же задач за счет одновременного и согласованного изучения содержания технологий программирования;
- проводить качественный сравнительный анализ реализации алгоритмов решения задач с учетом особенностей различных технологий программирования и оценивать эффективность использования средств той или иной технологии программирования;
- развивать на новом уровне алгоритмическое и логическое мышление студентов в условиях преемственной системной скоординированной учебной деятельности студентов;
- приобрести прочные знания о реализации связей между технологиями программирования;
- формировать системные знания для реализации образовательных проектов в профессионально-педагогической деятельности учителя информатики.

Данные выводы подтверждаются результатами экспериментального исследования, проведенного на кафедре прикладной математики и информатики в 2014/2015 учебном году в рамках учебной дисциплины «Технологии программирования и методы алгоритмизации» автором статьи, доцентом С.И. Зенько и старшим преподавателем Н.Н. Нарейко.

Литература

1. Газейкина, А.И. Обучение программированию будущего учителя информатики / А.И. Газейкина // Педагогическое образование в России. – 2012. - №5. – С.45 – 48.
2. Сманцер, А.П. Формирование у студентов ценностного отношения к образованию в процессе обучения / А.П. Сманцер, Т.А. Ханалыев. – Минск: БГУ, 2010. – 303 с.
3. Страуструп, Б. Программирование: принципы и практика использования C++ = Programming: Principles and Practice Using C++ / Б. Страуструп. – М.: Вильямс, 2011. – 1248 с.

РЕПОЗИТОРИЙ БГУ