

ИСПОЛЬЗОВАНИЕ ПРОБЛЕМНО-ЭВРИСТИЧЕСКОГО ПОДХОДА ПРИ ОБУЧЕНИИ ПРОГРАММИРОВАНИЮ

РЯБИНКИН Г.М., БОЛТАК С.В.

*Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники», Республика Беларусь*

Аннотация: Методики проблемного и эвристического обучения являются достаточно эффективными инструментами преподавания, но, несмотря на свою богатую историю, до сих пор не получили широкого распространения. Кроме того, ощущается и недостаток публикаций на тему применения данных подходов к преподаванию технических дисциплин и, в частности, программирования. В данном докладе обобщается опыт обучения программированию с применением вышеназванных методик, рассматриваются случаи из практики и даются общие рекомендации.

Ключевые слова: эвристика, эвристическое обучение, проблемное обучение, программирование, методика преподавания.

USING THE PROBLEM-BASED HEURISTIC APPROACH IN TEACHING OF PROGRAMMING

RABINKIN H.M., BOLTAK S.U.

*Belarusian State University of Informatics and Radioelectronics,
Republic of Belarus*

Abstract: Heuristic and problem-based learning methods are quite effective teaching tools, but despite their rich history, they have not yet become common and widespread. In addition, there is still an insufficient number of publications on topic of implementation of named methods to engineering disciplines as well as computer programming. This report summarizes experience of teaching of programming using mentioned methods, provides case studies, and gives general recommendations.

Keywords: heuristics, heuristic learning, problem-based learning, programming, teaching methods.

Программирование является одной из ведущих дисциплин, необходимых для подготовки специалиста, готового использовать передовые инструменты в своей практике и самостоятельно воплощать в жизнь

новейшие технические решения. При этом как сфера деятельности, оно является само по себе весьма непростым процессом, требующим от субъекта творческий подход к решению задачи, умение нестандартно смотреть на проблему, комбинировать доступные и искать новые средства для достижения результата.

В этой связи, при обучении программированию важным становится не только передача слушателям необходимой информации и помощь в ее усвоении, но и развитие у них умения нешаблонно мыслить и готовности выходить за рамки известных им подходов. Выбранная преподавателем методика может в значительной степени сказаться на результатах изучения слушателями учебной дисциплины и программирование здесь не исключение. В современной педагогике одной из наиболее перспективных и эффективных методик преподавания считается проблемно-эвристическое обучение [1], при этом особенности подготовки и проведения занятий в соответствии с этим подходом в настоящее время в недостаточной степени обобщены и институционализированы, особенно в части, касающейся проведения занятий по техническим дисциплинам.

Но вместе с этим, именно проблемно-эвристический подход позволяет решить задачи качественной подготовки специалистов, воспринимающих программирование не как рудиментарный или недоступный инструмент, а способных в полной мере воспользоваться его преимуществами для автоматизации рутинных процессов и решения широкого спектра инженерных задач.

Говоря о проблемном и эвристическом подходах к обучению, хотелось бы отметить некоторую неопределенность вокруг этих понятий. Так, в ряде источников встречается, как и их использование в качестве взаимозаменяемых, так и противопоставление [2, 3]. Для внесения ясности давайте остановимся на следующем содержании данных категорий: при проблемном обучении предполагается передача опыта преподавателя обучаемым путем постановки задачи (проблемы), которая имеет конкретное решение или хотя бы направление решения. В свою очередь, эвристический подход подразумевает выработку обучаемыми собственного опыта в ходе решения открытой задачи, которая не имеет заранее определенного правильного решения [4].

При таком понимании данных определений проблемное обучение может выступать в качестве подготовительного этапа для эвристического. Так, ознакомление с некоторой темой и новыми для слушателей категориями можно организовать путем проведения проблемного занятия, в то время как открытые эвристические занятия помогут создать знание более высокого порядка, отгалкивающееся от изученных ранее категорий.

В качестве иллюстрации возможности использования данного подхода при обучении программированию давайте рассмотрим ряд практических примеров. Так, в ходе изучения первого языка программирования очень

важно, чтобы при знакомстве с новыми возможностями языка обучаемые уже успели столкнуться с проблемами, которые с помощью этих возможностей решаются.

В частности, при знакомстве с возможностью создавать подпрограммы (процедуры, функции, методы) слушатели уже должны иметь некоторый опыт решения простых задач без помощи этого инструмента. При этом важно позволить им самим задаться вопросом, возможно ли как-то вызвать один и тот же код повторно для сокращения объема исходного текста программы, упрощения внесения изменений в часто используемую логику, повышения читаемости кода и т.д. Причем необходимо, чтобы обучаемые на практике столкнулись со всеми проблемами, которые будут решены с помощью использования подпрограмм.

Именно опыт переживания данных проблем должен побудить их к поиску некоторого решения. Конечно, не стоит надеяться, что они сами изобретут процедурное программирование и пройдут тот путь, который проходили лучшие умы мира в первые годы существования программирования как практической дисциплины. Однако в рамках направляемого преподавателем совместного обсуждения обучаемые вполне способны выдвинуть гипотезу о возможности существования инструмента для решения названных проблем и продумать его обобщенный способ реализации.

Таким образом, мы не просто «выдаем» обучаемым инструмент для решения проблем, о которых они еще не подозревают, способом, который не понимают, а фактически позволяем им самостоятельно пройти путь осмысления необходимости конкретной возможности языка, что благоприятным образом сказывается на понимании контекста ее дальнейшего использования.

Данный принцип проблемного обучения применим не только для случаев изучения базовых возможностей языков программирования, но и при знакомстве с более высокоуровневыми концепциями – парадигмами программирования или, например, такой частной особенностью объектно-ориентированного дизайна, как паттерны проектирования. Так, довольно просто объяснить человеку, имеющему даже базовые знания и незначительный опыт программирования, содержание и суть паттернов «Singleton» или «Abstract Factory», но намного тяжелее добиться понимания, какие проблемы эти шаблоны решают, а также когда и в каком контексте их необходимо применять.

Проблемный подход позволяет довольно просто и эффективно передать этот опыт от преподавателя обучаемым, если правильно поставить учебную проблему. Проблема не обязательно должна формулироваться явно, она может возникнуть перед обучаемым с его точки зрения естественным образом. Так, при попытке обучаемого написать приложение даже с минимальным функционалом в рамках парадигмы объектно-

ориентированного программирования (например, адресную книгу с консольным интерфейсом), он непременно столкнется с необходимостью организовать взаимодействие компонентов приложения. Не будучи знакомым с такими категориями как принцип тесной связанности, инверсия зависимостей и др., применяемыми при проектировании объектно-ориентированных систем, обучаемый наверняка их нарушит.

При публичном разборе одного из практических заданий не надо указывать на эти ошибки, так как с точки зрения слушателей это не ошибки, ведь нарушенные ими принципы созданы для решения проблем, о которых они пока не знают. Однако, если далее симитировать распространенную в реальной жизни ситуацию: дополнить техническое задание новыми требованиями, расширить функционал приложения (например, потребовать возможность вывода списка контактов не только в стандартный поток вывода, но и сохранения в виде файлов разных форматов), то слушатели сами увидят узкие места в своем приложении и попробуют решить эту задачу. В ходе совместного обсуждения преподаватель может пояснять, в каких условиях то или иное решение может привести к дополнительным сложностям и постепенно вывести обучаемых к самостоятельной выработке подлежащих изучению порождающих шаблонов проектирования – «Singleton» и «Abstract Factory».

Таким образом, кроме того, что слушатели получили практику в решении задач объектно-ориентированного дизайна, они имеют совершенно четкое представление о контексте применения изучаемого инструмента и проблемах, которые он решает.

Говоря об эвристическом подходе – постановке открытой учебной задачи, не имеющей однозначного правильного решения, тут при обучении программированию прекрасно подходят лабораторные практикумы, подразумевающие возможность создания относительно объемных проектов. При этом, для достижения необходимой сложности проекта и его масштаба при работе над ним приветствуется итерационный подход, предполагающий инкрементное расширение функциональных возможностей от задания к заданию.

Такой формат организации выполнения практических заданий позволяет добиться длительной работы над одним сквозным проектом (например, на протяжении всего семестра или тематического цикла), благодаря чему обучаемый может приобрести опыт разработки системы значительно большего объема, чего было бы невозможно добиться при серии несвязанных между собой заданий, предполагающих работу над самостоятельными законченными приложениями незначительного объема.

Возрастающая сложность приложения становится особенно важна при изучении концепций, связанных с парадигмой объектно-ориентированного программирования. Перед обучаемыми возникают проблемы масштабирования ранее реализованных решений, расширения функционала

приложения, поддержки уже функционирующих частей системы, добавленных в предыдущих итерациях – то есть они сталкиваются с задачами, характерными для разработки программного обеспечения в реальной жизни.

В ходе серии таких работ хорошей практикой является постановка обучаемым открытых задач по реализации функциональных требований. В частности, при заданной доменной области проекта и минимальных функциональных требованиях у слушателя остается полная свобода в определении структуры своего приложения, перечня используемых технологий (если это допускается учебной программой), конкретного определения объектной модели доменной области, путей реализации отдельных требований и т.д.

То есть обучаемый будет сам приобретать опыт в ходе решения проблем, с которыми ранее не сталкивался. Задача преподавателя на данном этапе – не сказать правильно или неправильно выполнено задание, а показать слушателю условия, при которых его решение может быть недостаточно эффективным, позволить обучаемому продолжить поиск более удачной реализации – то есть дальнейшее приобретение собственного опыта, а не принятие опыта преподавателя.

Таким образом, комбинация проблемного и эвристического подходов позволяет осуществлять обучение программированию, в рамках которого обучаемым будет передаваться не только зафиксированное знание, как в случае с репродуктивными методами обучения, но и опыт обучающего, что подразумевает намного более качественное и глубокое усвоение материала. Кроме того, проблемно-эвристический подход прививает обучаемым важные для программиста навыки творческого нешаблонного мышления и умение искать пути решения новых проблем, мотивирует их к самостоятельному обучению и постоянному расширению имеющихся знаний.

Вместе с этим, данная методика накладывает дополнительные требования к преподавателю, такие как более глубокое, чем того требует репродуктивный подход, понимание материала, богатый практический опыт применения изучаемых технологий, а также, что не менее важно, развитые коммуникативные навыки ввиду постоянной необходимости в прямом двустороннем контакте с обучаемыми.

Список литературы.

1. Дронь, М. И. Информационно-инновационные стратегии подготовки обучающихся к созданию и реализации прорывных технологий в деятельности человека / М. И. Дронь // Высшее техническое образование: проблемы и пути развития: матер. междунар. науч.-методич. конф., Минск, 1–2 нояб. 2018 г. / редкол.: В. А. Богуш [и др.]. – Минск: БГУИР, 2018. – С. 142–145.

2. Король, А. Д. Эвристический урок. Результаты, анализ, рефлексии (Как разработать и провести эвристический урок): методическое пособие. 2-е изд. / А. Д. Король. – Минск: Вышэйшая школа, 2018. – 223 с.

3. Махмутов, М. И. Проблемное обучение. Основные вопросы теории / М. И. Махмутов. – М.: Педагогика, 1975. – С. 246–258

4. Улезко, Ю. Д. Методические условия применения эвристических методов обучения на уроках биологии / Ю. Д. Улезко. – Екатеринбург: Уральский государственный педагогический университет, 2018. – С. 9–12.