

АКТИВНЫЕ МЕТОДЫ ОБУЧЕНИЯ ФУНКЦИОНАЛЬНОМУ ПРОГРАММИРОВАНИЮ

И. Н. Блинов¹⁾, Т. А. Лядинский²⁾

*Белорусский государственный университет, Республика Беларусь, Минск,
blinov@gmail.com*

*Белорусский государственный университет, Республика Беларусь, Минск,
mmf.lyadinskTA@bsu.by*

Активные методы обучения, применяемые в преподавании функционального программирования, хорошо развивают практические навыки. Основное внимание уделяется проектному обучению, код-ревью, решению сценарных задач и визуализации, которые способствуют глубокому пониманию ключевых концепций. Применение данных методов позволяет повысить вовлеченность, развить критическое мышление и практические навыки студентов.

Ключевые слова: активные методы обучения; функциональное программирование; проектное обучение; код-ревью; визуализация.

ACTIVE METHODS OF TEACHING FUNCTIONAL PROGRAMMING

I. N. Blinou¹⁾, T. A. Lyadinsky²⁾

¹⁾*Belarusian State University, Belarus, Minsk, blinov@gmail.com*

²⁾*Belarusian State University, Belarus, Minsk, mmf.lyadinskTA@bsu.by*

Active learning methods used in teaching functional programming are good at developing practical skills. The main attention is paid to project-based learning, code-review, scenario problem solving and visualization, which promote deep understanding of key concepts. These methods can increase engagement, develop critical thinking and practical skills of students.

Keywords: active learning methods; functional programming; project-based learning; code-review; visualization.

Введение

Активные методы обучения представляют собой педагогические стратегии, направленные на повышение вовлеченности студентов в образовательный процесс. В отличие от традиционных методов, где основное внимание уделяется передаче информации от преподавателя к учащимся,

активные методы предполагают непосредственное участие студентов в обучении. Это может быть решение задач, групповые обсуждения, проектная деятельность, моделирование ситуаций и другие формы интерактивного взаимодействия [1; 2].

Основная цель активных методов обучения — развитие у студентов практических навыков, критического мышления, способности к анализу и самостоятельному решению проблем. В контексте обучения функциональному программированию активные методы особенно ценны, так как позволяют студентам глубже понять концепции, такие как чистые функции, иммутабельность данных, ленивые вычисления и рекурсия.

Виды активных методов обучения, применимых к функциональному программированию:

- Проектное обучение. Студенты разрабатывают программные проекты на функциональных языках (Haskell, F#, Scala, Kotlin, Python), решая практические задачи;
- Обучение через программирование (learning by coding). Практическое кодирование и участие в учебных хакатонах [2];
- Обучение на основе решения задач (problem-based learning). Разбор сложных задач, требующих функционального подхода к решению [3];
- Геймификация. Использование игровых механик для вовлечения студентов в процесс обучения;
- Код-ревью и парное программирование. Совместная работа над кодом помогает глубже понять принципы функционального программирования;
- Сценарные задачи. Они помогают студентам увидеть практическое применение функционального программирования в задачах, с которыми сталкиваются различные проекты [3].

Теоретические основы

Проектное обучение является одним из наиболее удачных подходов к освоению функционального программирования. В рамках этого метода студенты работают над созданием реальных приложений, что позволяет им применять теоретические знания на практике и развивать важные профессиональные навыки.

Преимущества проектного обучения:

- Глубокое понимание концепций. Разработка функциональных приложений помогает студентам осознать важность принципов функционального программирования, таких как чистые функции, иммутабельность и ленивые вычисления;

- Развитие практических навыков. Студенты получают опыт написания кода на функциональных языках программирования, таких как Haskell, Scala, F#, Kotlin, Python, что делает их более подготовленными к работе в реальных проектах разных направлений информационных технологий, начиная от клиент-серверных приложений, заканчивая машинным обучением и аналитикой данных;

- Командная работа. Проектное обучение часто предполагает групповую деятельность, что способствует развитию навыков взаимодействия в команде и обмена знаниями. Это также развивает социальные навыки, что является важным фактором в развитии компетенций;

- Повышение мотивации. Создание полноценного продукта делает процесс обучения более увлекательным и наглядным.

Следующем наиболее важным элементом активных методов является код-ревью. Он хорошо применим совместно с проектной деятельностью. Этот метод предполагает систематический анализ кода, написанного студентами, с целью выявления ошибок, улучшения структуры и повышения читаемости программного кода. Также это позволит студентам погрузиться в основы agile-методик, которые применяются в коммерческой разработке. В целом такие практики позволяют сохранять чистоту кода проекта, погружаться в сам проект и привнести лучшие решения [2]. Также, для улучшения качества код-ревью необходимо иметь банк знаний, на который можно опираться, изучая основные парадигмы и примеры качественного написания кода в рамках парадигмы функционального программирования в сравнении с подходом объектно-ориентированного программирования.

Преимущества код-ревью:

- Обучение на чужих ошибках. Разбор кода одногруппников помогает студентам замечать типичные ошибки и избегать их в будущем;

- Развитие навыков критического анализа. Студенты учатся оценивать код с точки зрения производительности, читаемости и соответствия функциональным принципам;

- Повышение качества кода. Регулярные код-ревью способствуют выработке хорошего стиля программирования и соблюдению лучших практик;

- Формирование культуры командной работы. Код-ревью моделирует профессиональную среду разработки.

Сценарные задачи представляют собой учебные кейсы, основанные на реальных проблемах, требующих применения функционального программирования для их решения. Они помогают студентам увидеть практическое применение теоретических знаний и развить аналитическое мышление.

Преимущества сценарных задач:

- Приближенность к реальности. Студенты работают с задачами, схожими с теми, которые встречаются в индустрии;
- Развитие критического мышления. Требуется анализировать условия задачи и находить оптимальные решения;
- Повышение вовлеченности. Реалистичные сценарии делают обучение более интересным;
- Укрепление понимания функциональных концепций. Помогает студентам закрепить принципы чистых функций, рекурсии, иммутабельности и ленивых вычислений.

Так как функциональное программирование в основе своей широко использует принципы абстракции, то визуализация может также стать хорошим методом для изучения концепций функционального программирования.

Примеры визуализации функционального программирования:

- Диаграммы потока данных. Можно визуализировать передачу данных в чистых функциях и их композиции. Например, можно визуализировать, как `map`, `filter`, `reduce` изменяют данные;
- Графики рекурсии. Визуализация рекурсивных вызовов позволяет понять, как функции вызывают сами себя и какие промежуточные состояния проходят;
- Комбинаторные схемы. Использование блок-схем для объяснения функциональных операторов, таких как `map`, `filter` и `reduce`;
- Интерактивные среды. Использование инструментов вроде Jupyter Notebook для пошагового выполнения кода и отображения изменений в данных.

Регулярная обратная связь и оценка прогресса студентов позволяют выявлять трудности и корректировать подходы к обучению [1].

Преимущества формативного оценивания:

- Индивидуальный подход. Позволяет преподавателю адаптировать обучение под уровень подготовки каждого студента;
- Обратная связь. Студенты получают рекомендации по улучшению своих навыков и устранению ошибок;
- Мотивация к обучению. Регулярная оценка достижений помогает студентам отслеживать свой прогресс;
- Гибкость. Возможность оперативно корректировать программу обучения в зависимости от результатов студентов.

Заключение

Активные методы обучения играют ключевую роль в освоении функционального программирования, обеспечивая глубокое понимание концепций и развитие практических навыков. Использование проектного обучения, парного программирования, код-ревью и интерактивных заданий позволяет студентам не просто запоминать синтаксис и теоретические принципы, но и осваивать фундаментальные идеи, такие как чистые функции, иммутабельность, ленивые вычисления и рекурсия.

С помощью активных методов обучения функциональному программированию, студентов можно погрузить в сценарную среду разработки, что позволит им вникнуть в бизнес-процессы, а использование функциональных подходов в отдельных языках программирования дадут знания и в определенных областях, начиная от алгоритмических основ и написания собственных реализаций отдельных алгоритмов, в том числе машинного обучения, заканчивая разработкой мобильных и веб приложений.

Библиографические ссылки

1. Experiences of early assessment to teach functional programming. Journal of Functional Programming. Cambridge Core [Electronic resource]. URL: <https://www.cambridge.org/core/journals/journal-of-functional-programming/article/experiences-of-early-assessment-to-teach-functional-programming/667F18D3611D4A3A7C7C6A2A2FA4FB3C/> (date of access: 12.03.2025).
2. Learning Styles Impact Students' Perceptions on Active Learning Methodologies [Electronic resource]. URL: <https://www.mdpi.com/2227-7102/14/3/250> (date of access: 12.03.2025).
3. Teach Programming Using Task-Driven Case Studies: Pedagogical Approach, Guidelines, and Implementation [Electronic resource]. URL: <https://www.mdpi.com/2073-431X/13/9/221> (date of access: 12.03.2025).