Быстрицкая Анна Юрьевна
канд. техн. наук, доцент
ФГБОУ ВО «Московский государственный
технический университет им. Н.Э. Баумана»
г. Москва

ОБУЧЕНИЕ СТУДЕНТОВ ИСПОЛЬЗОВАНИЮ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ГЕНЕРАЦИИ ПРОГРАММНОГО КОДА

Аннотация: *в статье описывается опыт использования систем искусственного интеллекта при обучении студентов по направлению «Программная инженерия». Представлена разработанная автором методика проведения комплексной лабораторной работы для студентов первого курса, объединяющая компетенции в области программирования на языке Си, функционального моделирования IDEF0 и тестирования программного обеспечения. Рассмотрены вопросы корректного взаимодействия с системами искусственного интеллекта при разработке программного обеспечения и верификации полученных результатов.*

Ключевые слова: *искусственный интеллект, большая языковая модель, генерация программного кода, программная инженерия, функциональное моделирование, тестирование программного обеспечения, верификация программного кода, обучение программированию.*

Современный процесс разработки программного обеспечения неразрывно связан с использованием систем искусственного интеллекта. Будущие специалисты в области программной инженерии должны свободно владеть навыками корректного использования ИИ-инструментов. К их числу относятся:

– составление запроса, позволяющего получить программный код для решения поставленной задачи;

– критическое отношение к сгенерированному результату;

– тестирование и верификация полученного программного кода.

Content is licensed under the Creative Commons Attribution 4.0 license (CC-BY 4.0)

В данной работе описывается методика проведения комплексной лабораторной работы, интегрирующая ИИ-инструменты в образовательный процесс и смещающая акцент с простого написания кода на его проектирование и верификацию. Она позволяет обучающимся познакомиться с особенностями генерации программного кода с помощью больших языковых моделей и получить опыт практического использования передовых решений в области искусственного интеллекта. Работа предназначена для студентов первого курса и предполагает совместное применение знаний и навыков, полученных в рамках курсов «Программирование на Си» и «Основы программной инженерии», а также проектно-технологической практики.

В основе лабораторной работы лежит решение достаточно простой задачи по программированию на языке Си. Ее условие специально составлено в общем виде, чтобы студент мог самостоятельно сформулировать и обосновать допущения, которые он сделает при составлении программного кода. Задания индивидуальные. Они охватывают следующие темы: работа с двумерными массивами, алгоритмы сортировки, обработка текста, состоящего из нескольких строк.

Первым этапом лабораторной работы является формулирование допущений. Студент должен предложить и обосновать ограничения, которые он накладывает на входные данные, описать формат ввода и форму представления полученных результатов.

На основе сделанных допущений требуется разработать тесты «черного ящика», выделив необходимые классы эквивалентности [1]. При этом студент должен написать, что именно проверяет каждый тест, привести значения конкретных исходных данных и указать ожидаемый результат работы программы.

На втором этапе лабораторной работы необходимо выполнить декомпозицию рассматриваемой задачи с использованием методологии функционального моделирования IDEF0 [2]. Количество уровней декомпозиции студент определяет самостоятельно таким образом, чтобы оно приводило к однозначному решению задачи. Для построения IDEF0-диаграмм используется программное приложение Ramus Educational [3].

Содержимое доступно по лицензии Creative Commons Attribution 4.0 license (CC-BY 4.0)

Первые два этапа работы позволяют закрепить навыки, полученные при освоении тем «Тестирование программного обеспечения» и «Методология функционального моделирования IDEF0» в рамках курса «Основы программной инженерии».

Третьим шагом становится взаимодействие с системой искусственного интеллекта. На основе сформулированных допущений и разработанной декомпозиции задачи студенту необходимо подготовить запрос (промпт) для большой языковой модели, который позволит получить автоматически сгенерированный программный код для решения исходной задачи. Базовым ИИ-инструментом на данном этапе является Научно-образовательная платформа «Королёв ИИ», разработанная в МГТУ имени Н.Э. Баумана [4]. Допускается использование других систем искусственного интеллекта, но в этом случае студент должен обосновать свой выбор. Если сгенерированная программа содержит ошибки и не проходит компиляцию, то обучающийся должен изменить промпт и повторить генерацию. Вся история запросов приводится в отчете, при этом для неудачных вариантов необходимо перечислить их недостатки или указать причину замены.

Полученный программный код тестируется на данных, подготовленных на первом этапе лабораторной работы. Для автоматизации этого процесса учащиеся используют скрипты, разработанные в ходе проектно-технологической практики. Если какой-то тест не проходит, студенту необходимо определить причину сбоя и указать ее в отчете. Если программа не прошла более половины тестов, требуется изменить промпт, повторно сгенерировать код и снова провести его тестирование.

На четвертом этапе лабораторной работы студенты проводят замеры времени работы полученной программы и объемов потребляемой памяти. При этом они закрепляют навык применения методики, освоенной в рамках проектно-технологической практики.

Пятый этап посвящен доработке полученного программного кода. Студент должен проанализировать программу, сгенерированную системой искусственного интеллекта, и внести такие исправления, чтобы проходили все тесты,

Content is licensed under the Creative Commons Attribution 4.0 license (CC-BY 4.0)

разработанные на первом этапе. Затем учащимся необходимо найти в программе нерациональные блоки и улучшить их. Последним заданием этого этапа является оформление программного кода в соответствии с требованиями кафедры. Окончательный вариант программы еще раз проходит все тесты и для него замеряются время работы и объем потребляемой памяти.

Отчет о ходе выполнения лабораторной работы оформляется в виде отдельной Wiki-страницы в учебном репозитории студента на платформе GitLab. В отчет включаются:

– условие решаемой задачи;

– сделанные допущения;

– тесты «черного ящика» с указанием классов эквивалентности;

– IDEF0-диаграммы;

– название используемой системы искусственного интеллекта;

– история запросов с указанием недостатков промежуточных вариантов;

– полученный текст программы;

– результаты тестирования сгенерированной программы;

– описание внесенных исправлений;

– результаты замеров времени работы и объемов потребляемой памяти;

– вывод.

Защита выполненной работы проходит в форме беседы с преподавателем, в ходе которой студент обосновывает принятые решения и комментирует особенности программы, сгенерированной системой искусственного интеллекта. Особенное внимание уделяется верификации полученной программы. Преподавателю необходимо формировать у обучающихся критическое отношение к результатам работы систем искусственного интеллекта, для чего следует подробно обсуждать со студентом те исправления, которые он внес в программу, акцентируя внимание на необходимости доработки автоматически сгенерированного кода.

Таким образом, в ходе выполнения описанной лабораторной работы на примере простой задачи студенты знакомятся с основными этапами разработки программного обеспечения. Обучающиеся приобретают навык использования

Содержимое доступно по лицензии Creative Commons Attribution 4.0 license (CC-BY 4.0)

больших языковых моделей для генерации программного кода, а необходимость верифицировать программу и исследовать ее эффективность способствует формированию у них критического отношения к полученным результатам. Предложенная комплексная лабораторная работа способствует укреплению междисциплинарных связей между курсами и формирует у студентов ответственный подход к использованию больших языковых моделей в профессиональной деятельности.

Список литературы

1. Майерс, Г. Искусство тестирования программ / Г. Майерс, Т. Баджетт, К. Сандлер. – Москва : Диалектика, 2020. – 272 с.

2. Методология функционального моделирования IDEF0. – URL: https://advanced-quality-tools.ru/assets/idef0-rus.pdf?ysclid=mwggu6f7m5456836085 (дата обращения: 14.11.2026). – Текст : электронный.

3. Программный пакет «Ramus Educational». – URL: https://ramus-educational.software.informer.com (дата обращения: 14.11.2026). – Текст : электронный.

4. Научно-образовательная платформа Королёв ИИ. – URL: https://korolev-ai.bmstu.ru/ (дата обращения: 13.11.2026). – Текст : электронный.

Content is licensed under the Creative Commons Attribution 4.0 license (CC-BY 4.0)