Использование генеративного искусственного интеллекта для PascalABC.NET

Михалкович С.С.

Южный федеральный университет,
Институт математики, механики и компьютерных наук
E-mail: miks@sfedu.ru

В последние два года генеративный искусственный интеллект (ИИ) переживает стремительный рост, успешно решая разнообразные задачи, связанные с языками программирования. В данной работе рассматриваются возможности использования генеративного ИИ для повышения эффективности и функциональности системы программирования PascalABC.NET, которая активно развивается последние 10 лет [1]-[3] и основные этапы развития которой многократно докладывались на конференции СИТО начиная с 2004 года [4].

Исследование охватывает как создание заданий и учебных материалов, так и улучшение языковых конструкций и продвижение PascalABC.NET в образовательной среде. Анализируются успешные примеры промптов для реализации поставленных задач, а также потенциальные трудности и способы борьбы с ними.

В данной статье описываются исключительно примеры успешных промптов (запросов к моделям генеративного ИИ). Для исследования были использованы следующие модели: ChatGPT, DeepSeek, Qwen, Grok и Claude.

В ходе исследования были рассмотрены следующие типы задач:

1. Генерация кода на PascalABC.NET для решения различных задач с целью сформулировать рекомендации по формулированию эффективных запросов.
2. Создание примеров кода на PascalABC.NET с объяснениями для учебных целей – от подготовки к занятиям до обновления справочной системы.
3. Генерация заданий для обучаемых – создание задач, адаптированных под конкретные темы и уровень подготовки учащихся, с учетом ограничений на используемые языковые конструкции PascalABC.NET.
4. Продвижение PascalABC.NET в образовательной сфере – анализ мнений языковых моделей о стратегиях возвращения интереса к PascalABC.NET в образовательном процессе, учитывая его историческую популярность языка Pascal и текущее вытеснение языком Python. Рассмотрены как традиционные методы, так и креативные подходы для восстановления позиций PascalABC.NET.

5. Создание библиотеки машинного обучения на PascalABC.NET, включая инструменты для работы с данными, загрузку из CSV-файлов, а также реализацию пяти наиболее популярных моделей машинного обучения вместе с примерами их использования на классических датасетах.
6. Подготовка новостей для Telegram-канала PascalABC.NET – генерация или улучшение контента для информирования сообщества о нововведениях, обновлениях и событиях, связанных с платформой.

При генерации кода задач на PascalABC.NET все рассматриваемые языковые модели изначально демонстрировали тенденцию к созданию кода на устаревшем диалекте Паскаля, игнорируя современные конструкции и стандартные функции PascalABC.NET. Однако после нескольких уточняющих запросов модели успешно адаптировались и начинали генерировать корректный и современный код. Поскольку языковые модели не сохраняют контекст между сессиями, были определены ключевые слова и формулировки, позволяющие получать правильный код с первого или почти с первого запроса.

Приведем несколько показательных примеров.

Пример 1. ИИ Grok.

Запрос 1. Сгенерируй код на современном PascalABC.NET, который находит второй минимум в динамическом массиве. Используй описание переменных с автовыводом типа.

Запрос 2. Используй средства Linq. Для заполнения массива используй ArrRandomInteger(n) которая и выделяет память. Для вывода используй метод расширения .Println

Результат (идеальный).

```
var arr := ArrRandomInteger(10);
arr.Println;
var min1 := arr.Min;
var min2 := arr.Where(x -> x > min1).Min;
```

Вывод. Данный пример иллюстрирует ситуацию, в которой не надо было ничего исправлять.

Пример 2. ChatGPT.

Запрос. Напиши решето Эратосфена в PascalABC.NET, используя функциональный стиль.

Результат.

```
var n := 100;
var primes := ArrGen(n-1, i -> i + 2)
      .Where(x -> (2..Round(Sqrt(n)))
          .All(d -> (x = d) or (x mod d <> 0)));
```

Вывод. Данный пример показывает важность словосочетания «используя функциональный стиль». Поскольку в старом Паскале этого стиля нет, то ИИ сразу переключается на генерацию современного кода.

Пример 3. Решение задач олимпиады – ChatGPT.

Запрос. Дана строка слов, разделенных пробелами. Найдите два самых часто встречаемых слова. Если есть несколько слов с одинаковой частотой, выберите первые два по алфавиту. (запрос делался в сессии, где стиль кода был уже установлен)

Результат.

```
var s := ReadAllText('input.txt').ToLower;
var words := s.ToWords(AllDelimiters);
var freq := words.GroupBy(w -> w)
  .ToDictionary(g -> g.Key, g -> g.Count);
var top2 := freq.OrderByDescending(p -> p.Value)
  .ThenBy(p -> p.Key).Take(2);
top2.Println;
```

Вывод. ИИ решил считывать данные из файла input.txt, поскольку на олимпиадах обычно делают так. Это ошибка – данные надо было считывать с консоли.

При генерации заданий и примеров к занятиям с помощью ИИ вначале достаточно было сформулировать контекст «генерируй современный код», после чего код всех заданий генерировался стилистически правильно с минимальными ошибками. Наиболее впечатляющей была генерация 10 задач на словари в PascalABC.NET. Задания здесь требовалось упорядочить по сложности и придумать разнообразные. ИИ с этим не справлялся, поэтому задания отбирались вручную, упорядочивались по сложности, а потом ИИ предлагалось догенерировать задания по примерно понятным недостающим конструкциям, например:

Запрос. Сгенерируй задание на словари, индексированные строками, при этом значениями выступают массивы целых.

Ответ ChatGPT.

Дан словарь студентов и их оценок, заданных в виде массива. По нему создать словарь студентов и для каждого вычислить средний балл. Новый словарь выведите с помощью .Println.

Пример кода выполненного задания.

```
var studentGrades: Dictionary<string, array of integer>
  := Dict(
    'Иван' to |5, 4, 4, 3|, 'Мария' to |4, 4, 5, 5|,
    'Алексей' to |3, 3, 4, 4|, 'Ольга' to |5, 5, 5, 4|
  );
var averageGrades := new Dictionary<string,real>;
```

```
foreach var pair in studentGrades do
  averageGrades[pair.Key] := pair.Value.Average;
averageGrades.Println
```

Вывод. Ответ впечатляет, и после некоторой стилистической обработки вручную он был принят в комплект заданий.

Одним из перспективных способов использования генеративного ИИ – перевод кода с одного языка на другой. В частности, при решении одной сложной задачи в известном проекте Rosetta Code ChatGPT с первого запроса почти правильно перевел код с Python на PascalABC.NET с элементами LINQ. Здесь было понятно, что будут мелкие ошибки при переводе, но впечатляет именно использование для перевода совершенно других конструкций, характерных только для малоизвестного языка PascalABC.NET.

Еще одним перспективным примером использования генеративного ИИ является генерация новых библиотек по известным алгоритмам. Таким образом была полностью создана библиотека упрощенного машинного обучения с пятью самыми распространенными моделями: линейной регрессией, логистической регрессией, деревьями решений, методом k ближайших соседей и нейронными сетями прямого распространения. Спроектирована также архитектура упрощенного класса DataSet, содержащего только вещественные характеристики. Сгенерированный код был получен полностью в иллюстративных целях, написан неэффективно по времени работы, но исключительно в современном функциональном стиле и содержит в совокупности около 800 строк кода. Интересно, что на стандартных датасетах обеспечивается точность, совпадающая с точностью стандартных Python-библиотек. Другой интересный момент состоит в том, что в одном методе машинного обучения – логистической регрессии – в первом варианте кода обеспечивалась неприемлемая точность, и ИИ сам определил проблему – необходимость нормализации данных, после чего сам сгенерировал класс шкалирования и вызвал его в основном коде, в результате чего точность стала совпадать с теоретической.

Очень интересными являются беседы с GhatGPT о сравнении понятности кода на PascalABC.NET и Python. Здесь хорошо видно, что ChatGPT собирает стереотипы в сети, что Python – самый эффективный по понятности. Однако после обсуждения конкретных фактов ИИ меняет своё решение. Вот его окончательный вердикт: «Если студенты изучают PascalABC.NET с нуля, метод Where может быть даже интуитивнее и проще для освоения, чем списковое включение в Python».

Итак, данное исследование подтвердило значительный потенциал генеративного ИИ для работы с различными аспектами

PascalABC.NET. Несмотря на то, что модели изначально склонны генерировать устаревший код, правильно сформулированные запросы позволяют получать современный и эффективный код, соответствующий возможностям языка. Это открывает новые перспективы для автоматизации создания учебных материалов, заданий и даже продвижения PascalABC.NET в образовательной сфере.

Особый интерес представляет генерация кода для создания новых библиотек PascalABC.NET – в частности, для задач машинного обучения, что демонстрирует гибкость и адаптивность генеративного ИИ. Хотя пока это не замена специализированным библиотекам, такие эксперименты показывают, что генеративный ИИ способен существенно расширить возможности PascalABC.NET.

Несомненно, мы находимся на пороге революции, связанной с использованием генеративного ИИ в программировании и образовании.

Литература

1. *Михалкович С.С.* Система программирования PascalABC.NET - 20 лет развития / XXX Научная конференция «Современные информационные технологии: тенденции и перспективы развития». Материалы конференции. Ростов-на-Дону, 2023. С. 287–290.
2. *Бондарев И.В., Михалкович С.С.* Система программирования PascalABC.NET: 15 лет развития / XXV Научная конференция «Современные информационные технологии: тенденции и перспективы развития». Материалы конференции. Ростов-на-Дону, 2018. С. 31–34.
3. *Бондарев И.В., Михалкович С.С.* Система программирования PascalABC.NET: новые возможности 2015-16 гг. / Труды XXIII Научно-методической конференции «Современные информационные технологии: тенденции и перспективы развития». Ростов-на-Дону: Изд-во ЮФУ, 2016. С. 69–71.
4. *Михалкович С.С.* Учебная система программирования Pascal ABC / Современные информационные технологии в образовании: Южный Федеральный округ: Тезисы докладов. Ростов-на-Дону, 2004. С. 156–158.