

- веселые картинки-мемы и разнообразные бонусы, поощрения (которые преподаватель может также отключить либо назначить при первоначальном знакомстве с системой);
 - мобильность: учащийся может делать задание на планшете или смартфоне;
 - интеграция сервисов Quizizz и Google classroom способствует облегчению рутинных задач преподавателя, а учащимся позволяет проходить материал интересно и увлекательно;
 - обновление и развитие платформы.
- Недостатки онлайн-квизов:
- помехи связи и обрыв соединения с сетью Интернет;
 - наличие современного смартфона;
 - наличие у преподавателя навыков уверенного пользования персональным компьютером.

Применение квиз-технологий в образовании помогает улучшить и упростить работу преподавателя по контролю и оценке знаний обучающихся, а также повысить мотивацию учащихся к приобретению новых знаний за счет максимальной увлеченности в процесс. Формат квиз-технологий подходит для учеников разных возрастных групп. Визуализация любых знаний – это одна из лучших форм запоминания материала [4; 5].

Список использованных источников

1. Геймификация образовательного процесса // Методическое пособие под ред. Эйхорн М. В. [Электронный ресурс]. – Режим доступа: https://toipkro.ru/content/files/documents/podrazdeleniya/pip/geymifikatsia_obrazovatel'nogo_protsesta.pdf. – Дата доступа: 10.10.2023
2. Бурняшева, Л. А. Активные и интерактивные методы обучения в образовательном процессе высшей школы : методическое пособие / Л. А. Бурняшева. – Москва : КноРус, 2016. – 219 с.
3. Емельянова, Т. В. Игровые технологии в образовании : учебно-методическое пособие / Т. В. Емельянова, Г. А. Медяник. – Тольятти : Издательство ТГУ, 2015. – 88 с.
4. Монахов, Д. Н. Облачные технологии теория и практика : учебное пособие / Д. Н. Монахов, Н. В. Монахов, Г. Б. Прончев, Д. А. Кузьменков. – Москва : МАКС Пресс, 2013. – 128 с.;
5. Зайцев, В. С. Современные педагогические технологии : учебное пособие / В. С. Зайцев. – Челябинск : ЧГПУ, 2012. – 411 с.

УДК 378.147

Г. А. Заборовский, А. В. Наливко

G. Zaborovsky, A. Nalivko

УО «Белорусский государственный педагогический университет имени Максима Танка» (Минск, Беларусь)

ПОСТРОЕНИЕ СИСТЕМ ГЕНЕРАЦИИ УЧЕБНЫХ ДИАЛОГОВ

BUILDING SYSTEMS FOR GENERATING LEARNING DIALOGUES

Рассмотрены особенности построения систем автоматической генерации учебных диалогов по информатике.

The features of constructing systems for automatic generation of learning dialogues in computer science are considered.

Ключевые слова: образование; учебный диалог; информатика; интеллектуальная система.

Keywords: education; learning dialogue; computer science; intelligent system.

Учебный диалог является одним из эффективных методов обучения. В учреждениях образования диалог учащегося по учебным предметам традиционно реализуется путем вербального общения с преподавателем. Применение диалогового метода в рамках существующей классно-урочной системы обучения ограничено дефицитом времени. На современном этапе развития информационных и телекоммуникационных технологий актуальной проблемой является разработка систем автоматической генерации и поддержки учебных диалогов.

В настоящее время предлагаются различные программные средства, способные генерировать вопросы и простейшие эталоны ответов. Их эффективность зависит от наполненности базы данных вопросов-ответов, сохранения и анализа результатов работы учащихся, пополнения базы данных новыми вариантами вопросов и ответов. В большинстве случаев эти системы предлагают ограниченные типы и фиксированные наборы вопросов-ответов, которые чаще всего используются для создания тестов [1].

Наше исследование посвящено рассмотрению особенностей построения систем автоматической генерации учебных диалогов по информатике с широким спектром и различными формами вопросов-ответов разных типов. Дидактической базой разработки и использования диалогового метода являются идеи проблемного, личностно ориентированного обучения и системно-деятельностного подхода. Методологическая основа построения систем автоматической генерации и поддержки учебных диалогов – это идеи математической логики и компьютерной лингвистики, алгоритмики и программирования [2].

Рассмотрим особенности программирования хорошо формализуемых диалогов. Так, многие вопросы по информатике и предполагаемые ответы можно формализовать и свести к ряду типовых шаблонов, что позволяет автоматически генерировать текст вопроса и эталоны ответов. Наиболее эффективным и несложно реализуемым является генерирование диалогов для формирования основных понятий информатики: изучения терминов, команд, формул, типовых синтаксических и алгоритмических конструкций. На языке программирования такие шаблоны можно представить структурами данных, которые содержат некоторые ключевые значения (например, термины, формулы, команды), а также дополнительные морфологические и синтаксические единицы текста.

Разработка генератора учебного диалога начинается с выделения инвариантной и вариативной составляющих вопросов, а также эталонов ответов. Приведем примеры такого подхода при разработке генератора диалога для формирования основных понятий языка C#.

Система генерирует и предъявляет учащемуся первый вопрос:

1. Как записывается логическая операция {varQ} в языке C#?

В шаблоне этого вопроса переменной {varQ} случайным образом присваивается одно из возможных значений: «И», «ИЛИ», «НЕ», а эталону ответа соответствующее выражение: &&, ||, !. Ответом может быть ввод выражения или выбор из предлагаемых, в том числе неверных: **And, Or, Xor, Yes, No...**

Логика проверки ответа задается в программном коде генератора, а постоянные и переменные составляющие можно хранить в массивах. Подобную систему легко модифицировать для другого языка программирования, заменив название на *PascalABC* или *Python* и соответствующие эталоны ответов. Более того, возможно создание универсального генератора с выбором темы и языка.

При задании в шаблонах нескольких вариативных составляющих возможно построение генераторов заданий с более сложной логикой и сценариями предъявления. Приведем пример генератора заданий для формирования навыков использования алгоритмических конструкций:

2. Дан фрагмент кода программы вычисления суммы чисел на языке C#.

```
int sum = 0;
for ( int i = {varQ1}; i <= {varQ2}; i += {varQ3} )
    sum += i;
```

Чему равен результат вычисления *sum*?

В этом примере случайным образом предъявляются три переменные составляющие {varQ1}, {varQ2}, {varQ3} из заданных диапазонов чисел, а введенный ответ сравнивается с вычисленным в программе результатом {varOtv}. Оптимального сочетания надежности, простоты и гибкости при использовании системы можно достичь, если значения инвариантных и вариативных составляющих хранить в закрытых базах данных, а последовательность предъявления и уровни сложности (веса) заданий в открытых сценариях обучения, которые может редактировать сам преподаватель.

Отметим некоторые особенности программирования систем генерации учебных диалогов, допускающих ввод текстовых ответов. Такой диалог формируется последовательностью пар: строка вопроса strQ – строка ответа strR. Вопросы генерируются подстановкой ключевых значений в соответствующие шаблоны и преобразованием к строковому типу. Строки введенных ответов перед сравнением с эталонами подвергаются обработке: удаляются незначимые символы и пробелы, выделяются ключевые для данного контекста лексемы (токены) – слова или словосочетания. При необходимости выделяются числовые данные. Существенно упростить обработку ответов и уменьшить количество требуемых эталонов позволяет использование регулярных выражений.

Построение генераторов простых хорошо формализуемых диалогов реализовано нами на языках программирования C#, Python, JavaScript и апробировано в рамках дисциплин «Системы и технологии программирования» и «Информационные системы и сети». При разработке на C# удобно использовать обобщенные типы данных: списки List<T> и словари Dictionary<T, U>, хранящие объекты в виде пар ключ – значение, где ключ – термин {varTerm}, а значение – определение данного термина {varDefTerm}. Необходимые шаблоны, вариативные части вопросов и ответов хранятся в базе данных SQL. Сгенерированные вопросы и задания предъявляются учащимся в отдельных окнах. Продолжение и развитие диалога определяется сценарием и логикой проверки ответа, которая задается для каждого типа диалога. В простейших случаях это пояснения и подсказки, возвраты и повторения вопроса в новом варианте.

Для реализации более сложных диалогов требуются другие модели представления знаний, например, иерархические или сетевые. Заметим, что первоначально базы данных терминов создаются экспертами, затем предполагается обучение системы с использованием идей машинного обучения (Machine Learning).

На данном этапе нами подробно рассмотрены особенности программирования хорошо формализуемых диалогов для формирования основных понятий информатики: терминов, команд, формул. Но даже в этих случаях анализ возможных ответов, вводимых на естественном языке, затруднителен.

Для решения подобных задач необходимо использовать средства более глубокого анализа текста на основе идей машинного обучения, когда диалоговая система будет обучена на множественном наборе данных и сможет более адекватно определять тип, структуру и смысл вопроса, а также предполагаемого ответа. Такую систему можно использовать как для анализа введенного текста, так и для получения ключевых слов, при условии, что форма и содержание вопросов будут достаточно вариативны.

В последнее время разработано множество реализаций машинного обучения для разных языков и систем программирования. Нами изучены возможности использования элементов машинного обучения в системе генерации учебных диалогов средствами открытой библиотеки ML.NET для языка программирования C# в среде MS Visual Studio

[3]. Подключение ML.Net дает возможность реализовать элементы машинного обучения как в сетевых, так и в автономных .NET-приложениях и получать более адекватный анализ категориального соответствия вопросов и ответов. Наиболее типичные задачи, которые можно решать средствами ML.NET: классификация и категоризация, (например, классификация элементов учебного диалога по их содержанию); оценка степени похожести текста вопросов и ответов; обнаружение аномальных отклонений; автоматическое разделение ответов учащихся на положительные и отрицательные.

В заключение отметим, что использование разрабатываемых систем генерации учебных диалогов будет способствовать реализации диалогового лично ориентированного подхода при изучении информатики.

Список использованных источников

1. Чельшкова, М. Б. Теория и практика конструирования педагогических тестов: Учебное пособие. / М. Б. Чельшков. – М. : Логос, 2002. – 432 с.
2. Боярский, К. К. Введение в компьютерную лингвистику. Учебное пособие. / К. К. Боярский. – СПб.: НИУ ИТМО, 2013. – 72 с.
3. Документация по ML.NET. [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/machine-learning/>. – Дата доступа: 20.03.2023.

УДК 378:004

С. И. Зенько, А. А. Маковчик

S. Zenko, A. Makouchyk

*УО «Белорусский государственный педагогический университет
имени Максима Танка» (Минск, Беларусь)*

О РАЗРАБОТКЕ ЭЛЕКТРОННОГО ПРИЛОЖЕНИЯ ДЛЯ ОБУЧЕНИЯ СТУДЕНТОВ МЕТОДИКЕ КОНСТРУИРОВАНИЯ ПЛАНОВ-КОНСПЕКТОВ УРОКОВ ПО ИНФОРМАТИКЕ

ABOUT DEVELOPMENT OF AN ELECTRONIC APPLICATION FOR TEACHING STUDENTS THE METHODOLOGY OF CONSTRUCTING LESSON PLANS IN INFORMATICS

В работе обосновывается необходимость обучения студентов методике конструирования план-конспектов уроков по информатике. В качестве современного средства для осуществления такой деятельности на различных этапах их подготовки предлагается использовать электронное приложение. Предполагается, что в результате у студентов будут развиваться востребованные компетенции для конструирования план-конспектов уроков и формироваться банка актуальных план-конспектов уроков по информатике.

The work substantiates the need to teach students methods of constructing lesson plans in informatics. It is proposed to use an electronic application as a modern means for carrying out such activities at various stages of their preparation. It is expected that as a result, students will develop the required competencies for constructing lessons plans and lessons materials development and form a bank of actual lesson plans in informatics.

Ключевые слова: методика обучения информатике; план-конспект урока; электронное приложение.

Keywords: methodology for teaching informatics; lesson plan; electronic application.