

ПРОЕКТНО БАЗИРАН ПОДХОД ЗА ВЪВЕЖДАНЕ НА ЛОГИЧЕСКО ПРОГРАМИРАНЕ В СРЕДНОТО УЧИЛИЩЕ

Магдалена Мъглижанова¹, Венета Табакова-Комсалова^{1,2},
Станмир Стоянов^{1,2}, Ласка Костадинова-Цанкова¹

¹ ПУ „Паисий Хилендарски“, Пловдив

² Институт по информационни и комуникационни технологии при Българска академия на науките, София

A PROJECT-BASED APPROACH TO INTRODUCING LOGIC PROGRAMMING IN THE SECONDARY SCHOOL

Magdalena Maglizhanova¹, Veneta Tabakova-Komsalova^{1,2},
Stanimir Stoyanov^{1,2}, Laska Kostadinova-Tsankova¹

¹ University of Plovdiv „Paisii Hilendarski“, Plovdiv, Bulgaria

² Institute of Information and Communication Technologies, Bulgarian Academy of Sciences, Sofia

* Авторите изказват благодарност към научен проект КП-06-М62/2 „Моделиране на знания в областта на българския фолклор“, финансиран по Фонд „Научни изследвания“, за финансиране на настоящата работа.

Abstract: With the development of technology, artificial intelligence is becoming an indispensable part of our daily lives and has a great impact on many aspects of society. The article explores the main characteristics of modern education, covering artificial intelligence, integrated learning, critical thinking, problem-solving skills, as well as active communication and teamwork abilities. It introduces a new project development strategy that aims to provide students with the necessary skills to engage in technological innovation by applying science and technology knowledge relevant to the real world. Additionally, the article outlines the expected steps involved in implementing STEM education, with an emphasis on promoting career autonomy and empowering students to make informed choices about their careers. These steps include promoting and popularizing logic programming in the Prolog language, providing support for talented students, and encouraging young people's engagement in creative and research activities.

Keywords: education, artificial intelligence, logic programming, project-based learning, STEM.

Въведение. С развитието на технологиите, изкуственият интелект става неизменна част от ежедневието ни и има голямо влияние върху множество аспекти на обществото. В образованието, изкуственият интелект предоставя безброй възможности. Тъй като, няма единна дефиниция за понятието „изкуствен интелект“ ние се спираме на дефиницията съгласно проекта за законодателен акт за Изкуствен интелект (ИИ) на Европейската комисия [1]. В нея е дадено, че „Система с изкуствен интелект“ (система с ИИ) означава софтуер, разработен с една или повече от техниките и подходите, посочени в приложение I, който може по отношение на даден набор от цели, определени от човек, да генерира резултати, като съдържание, прогнози, препоръки или решения, които оказват въздействие върху средите, с които взаимодействат. В приложение I на второ място е посочено – Логически и основани на знанието подходи, включително представяне на знанията, индуктивно (логическо) програмиране, бази от знания, машини за логически изводи и машини за дедукции, (символни) съждения и експертни системи. В тази връзка екипът ни започна въвеждане на логическо програмиране на езика Пролог в средното училище.

Логическото програмиране е парадигма в програмирането, която се основава на използването на логически изрази и правила за описание на проблемите и тяхното решаване. Въвеждането на логическо програмиране в средното училище може да бъде предизвикателство, но същевременно предлага множество ползи, като развиване на аналитичното мислене, решаването на проблеми и разбирането на сложни концепции. През последните години курсове по логическо програмиране и AI се преподават в различни страни и на ученици в различни възрастови групи. Нашият екип въведе логическото програмиране на Пролог в средното училище с дейността „Дигитална България в Пролог“ (част от международната инициатива, известна като „Пролог Education and Thinking“), работейки с група учени от цял свят. В тази група е и един от създателите на езика Робърт Ковалски. Пролог е особено полезен за решаване на проблеми в AI, както и в области като търсене, планиране и представяне на знания. Програмирането на Пролог е лесно и разбираемо, защото е близко до естествения език. Това води до лесно разбиране от ученици от всички възрасти и използва-

нето му в различни предмети в средното училище. Например, в [2] се твърди, че логическото програмиране е жизнеспособен и желан избор на парадигма за деца от началното училище, като се експериментира с 8-до 10-годишни деца в две държавни основни училища в Аржентина. В [3] авторите предполагат, че използването и изучаването на Пролог за повечето от темите в учебната програма по математика в гимназията (вероятност, алгебра, анализ или геометрия) позволява по-добро овладяване на концепциите на курса. Статия [4] представя нашите дейности за въвеждане на обучението по програмиране на Пролог. Средното училище в статия [5] предоставя преглед на използването на Пролог и неговите производни, които поддържат изследванията и развитието в областта на биоинформатиката и изчислителната биология. Преглед на инструментите за автоматизирано разсъждение в геометрията, разработени в Пролог, е даден накратко в статия [6]. Всичко това ни мотивира да изберем езика Пролог като основен инструмент за реализиране на проектно базирано обучение и обучение в STEM. И чрез него да се включат средни училища в България към световната инициатива „Пролог Образование и мислене“.

Проектно-базираният подход

Проектно-базираният подход е педагогически метод, при който учениците се учат чрез активно участие в реални и значими проекти. Този подход:

- Поддържа активното учене: Учениците активно участват в процеса на обучение чрез работа по конкретни проекти.
- Развива ключови умения: Учениците развиват умения за сътрудничество, комуникация, критическо мислене и решаване на проблеми.
- Осигурява контекст: Учебният материал се преподава в контекста на реални приложения, което улеснява разбирането и запомнянето.

Проектно-базираният подход представлява ефективна методология за въвеждането на логическо програмиране в училищната програма. Въвеждане на логическо програмиране чрез проекти се осъществява на следните стъпки описани в таблица 1.

Таблица 1. Етапи за въвеждане на логическо програмиране чрез проекти

Етапи	Дейности		
Планиране и Подготовка	Целеполагане: Определете целите на обучението и резултатите, които искате да постигнете чрез въвеждането на логическо програмиране.	Избор на проекти: Изберете подходящи проекти, които ще помогнат на учениците да приложат логическите принципи на програмиране. Примерни проекти могат да включват създаване на логически игри, решаване на математически задачи с помощта на логически програми или разработка на приложения, базирани на изкуствен интелект.	
Изпълнение на Проектите	Въведение в логическото програмиране: Започнете с въвеждането на основните концепции на логическото програмиране, като използвате примери и кратки упражнения. Езици като Пролог са подходящи за това.	Работа по проектите: Разделете учениците на малки групи и им възложете различни проекти. Нека всяка група разработи план за своя проект, включващ задачи, срокове и очаквани резултати.	Наблюдение и насоки: Преподавателите трябва да следят напредъка на учениците, да предоставят насоки и помощ при необходимост, и да организират редовни обсъждания и прегледи на проектите.
Оценка и Анализ	Представяне на проектите: В края на проекта, всяка група трябва да представи своята работа пред останалите ученици и преподаватели. Това може да включва демонстрация на разработените програми, обяснение на използваните методи и обсъждане на предизвикателствата и успехите.	Оценка на успеха: Оценете проектите въз основа на предварително определени критерии, като качество на кода, оригиналност на идеята, приложимост на решението и ефективност на представянето.	Обратна връзка: Осигурете обратна връзка на учениците, като подчертаете силните страни на техните проекти и областите за подобрение. Насърчете ги да рефлектират върху процеса на работа и да извлекат уроци за бъдещи проекти.

Реализация на проектно базиран подход за въвеждане на логическо програмиране в средното училище

Екипът ни предоставя реализиране на проектно базиран подход за въвеждане на логическо програмиране на Пролог в Професионална гимназия по електротехника и електроника (ПГЕЕ), град Пловдив. Избраната тема за проект от учениците е „Растенията в Космоса спасяват Земята“. Основната цел на проекта е да изследва проблематиката, свързана с изкуственият интелект и неговото въвеждане в образованието. Като първа специфична цел е дефинирано: Да се покажат ползите от ИИ, като например способността му да анализира големи обеми данни за кратък период от време. В този проект ИИ е използван за идентифициране на най-добрата генна мутация при растенията с цел проучване на нови сортове култури, устойчиви на климатични промени. За втора специфична цел е дефинирано: Да се реализира проектно-базираното обучение чрез комбинация от императивни и декларативни програмни езици, което ще помогне на учениците от специалност „Приложно програмиране“ да затвърдят знанията си. В същото време ще се демонстрират възможностите на STEM обучението като част от бъдещето на образованието. И като трета специфична цел е въведено: Да се създаде база данни, базирана на логика, с помощта на езика за програмиране Пролог, която да комуникира с потребителя и да предоставя полезна информация. По този начин ще се докаже полезността на Пролог за създаване на интелигентна бази данни и ще помогне на учениците да придобият знания в областта на логическото програмиране. Важно е ИИ да анализира проверена информация, базирана на научна документация. Благодарение на тези цели, проектът демонстрира безопасния начин на използване на ИИ от учениците и показва важноста да създават, а не само да използват ИИ за своите нужди. В този случай обучаващите се придобиват знания в областта на химия, биология, физика и програмиране. А също така се разпространяват знания от съответната област.

Въведение в темата на проекта

През 1969 година Катрин Джонсън създава сложно уравнение, което позволява безопасното излитане и кацане на космическата капсула проектирана от НАСА. Сложните изчисления са отнемали много време. Така се появява нуждата от иновация - компютърът IBM 7090. Ползите от внедряването на нови технологии, като IBM 7090 и език за програмиране Фортран са много. Например, по-бързата обработка на информация води до по-ефективни изчисления. Фортран е използван от програмистите за комуникация с компютрите, но, за съжаление, всеки напредък носи и своите негативи, като един от тях е съкращения на персонала. Затова е важно да се отбележи, че адаптацията е ключов фактор за успешното справяне с промените, породени от новите технологии. Днес предизвикателството пред съвременното общество е изкуственият интелект и единствената възможност е да се приспособим към новите тенденции и изисквания.

В тази статия се разглежда как група хора от дадена област се справят с бързото популяризиране на изкуствения интелект (ИИ). Съвременният учител е изправен пред трудната задача да се адаптира към бързо нарастващата популярност на генеративния ИИ, като ChatGPT. Необходимо е да се открият нови методи на преподаване и подготовка на учебния материал, заради различни предизвикателства. Например, учителите по програмиране срещат следните затруднения: заданията вече могат да бъдат решени за секунди от ChatGPT. Благодарение на партньорството между Пловдивския университет и Професионалната гимназия по електротехника и електроника – град Пловдив, се създава възможност за работа с програмния език Пролог, който е много подходящ за разработването на приложения с изкуствен интелект посредством логика. Използвайки Пролог, програмистът може да посочи набор от факти и правила, които интерпретаторът на Пролог впоследствие анализира и генерира решения. Така се предоставя възможност за създаване, управление и споделяне на голяма база от проверени знания. Логическото програмиране е програмна парадигма, която се основава на математическа логика. За разлика от езици като Java или C, програмите, написани на езици за логическо програмиране като Пролог, не са съставени от последователности от инструкции, а от набор от аксиоми или правила, които определят връзките между обектите. Програмният език Пролог може да бъде използван при създаване на AI-STEM проекти, което предоставя множество възможности. Поради тази причина, именно такъв проект се използва в случая. AI-STEM проектите имат потенциал да обединяват голям обем от данни, а ИИ може да ги анализира за кратък период от време. Този вид проекти предоставят възможност за научни изследвания и развитие в областта на образованието.

Методология

Участието в AI-STEM проекти може да подготви обучаващите се за изискванията на бъдещите пазари на труда, като развива техните технически и аналитични умения, критично мислене и други важни компетенции. Подходящите методи, използвани в тази разработка са:

- Проектно-базиран метод – иновативен метод на преподаване, който насърчава учениците да придобиват знания и умения чрез разработване на проекти, свързани с реални проблеми или автентични изследователски въпроси. Вместо традиционния подход, при който учениците пасивно приемат информация, ПБМ ги поставя в ролята на активни участници, които трябва да изследват, планират и реализират собствените си идеи. Този метод обогатява учебния процес с практическо при-

ложение на новите знания, развива критическо и креативно мислене и насърчава работата в екип.

- Изследователски метод – представляват систематични подходи и техники, използвани за събиране и анализ на данни с цел получаване на нови знания. Те са съществена част от научното изследване и могат да бъдат класифицирани в различни категории в зависимост от изследването и вида на информацията, която се анализира.

В конкретния случай, за да се създаде база данни от факти и правила, е необходимо да се изследва и анализира определена информация за растенията и ДНК мутациите. За тази цел, женският отбор по програмиране на ПГЕЕ – гр. Пловдив, заедно със своя ръководител, посещават Центъра по растителна система биология и биотехнология в гр. Пловдив. Срещата е организирана благодарение на д-р Веселин Петров, който предоставя подробна научна информация за физиологията на растенията и различните мутации. Тази съвместна работа е от съществено значение за създаването на база от знания, която ще служи за разработването на приложение с изкуствен интелект. Примерът с женския отбор по програмиране на ПГЕЕ – гр. Пловдив, демонстрира как изследователските методи могат да бъдат приложени на практика.

Към момента е събрана разнообразна информация, свързана с няколко основни аспекта: растеж и развитие на растенията; физиология на растенията; генетична структура на растенията; фактори за околната среда, като температура, влажност, светлина и др.

Определяне на основните етапи при създаване на този проект:

Планиране: В началото са изследвани нуждите и изискванията. Това включва анализ на целите на проекта и възможностите за изпълнение. След това се определят ролята на всеки участник в екипа, разпределят се задачите и отговорностите. Планират се и срещи, на които да се обменя информация между участниците. Появява се и необходимост от създаване на партньорство с други институции, в случая Центъра по биотехнологии и Пловдивски университет. Всички стъпки са описани в предварителна документация.

Анализиране: В процеса на работа са направени анализ на рисковете и възможностите, за да се идентифицират потенциалните проблеми и начини за тяхното преодоляване. Събраната информация се трансформира във вид на код, за да може да бъде анализирана от машината, в случая компютъра. Този процес помага на учениците да развият логическо мислене, като същевременно усвояват спецификата на материята и подобряват уменията си за работа в екип. Това предоставя възможност за прилагане на техните знания в различни сценарии и развиване на техните умения.

Инструменти за събиране и анализиране на данните: Работа в екип, формулиране на въпроси към научните сътрудници в центъра спрямо целите на проекта – в случая изследване на растенията.

Проектиране и осъществяване: Ако се комбинират няколко програмни езика (императивни: C#, JavaScript, декларативни: Пролог, HTML, CSS) това ще даде възможност за по-добра комуникация между потребител и интелигентна база данни, чрез подходящ интерфейс. В случая е интригуващо как ще се осъществи обединяването на императивни и декларативни езици и дали компилаторът на интегрираната среда „Visual Studio“ ще се справи с тази задача.

Тестване, поява на проблеми: Появява се необходимост от изпращане на различни видове заявки към базата. За съжаление фактите и правилата в базата данни създадена с Пролог са дефинирани само на английски език. Библиотеката, която се използва във „Visual Studio“ средата не работи на български език. Затова Пролог ботът отговаря само на английски.

Подробна информация при проектирането е представена в следните етапи:

Първи етап: комбиниране на няколко програмни езика. За да се направи интелигентна база данни е нужно да се дефинират факти и правила с език за програмиране Пролог. Като начало информацията за растенията е трансформирана във вид на код (Фигура 1) където фактите са свързани с: класификация на частите на растението; определяне характеристиките на растението; физиология на растенията; характеристики на инвитро средата, в която се извършват изследванията. Определят се дефиниции на правила за генетични мутации; въздействие на гравитацията; влияние на радиацията върху растенията; влияние на светлината; атмосферни условия.

Втори етап: Тестване на съвместимостта на двата езика. За да усвоят базови умения в програмирането, учениците от специалност Приложно програмиране в ПГЕЕ – гр. Пловдив използват конзолни приложения. Тези приложения са с по-проста структура в сравнение с уеб-базираните и по-лесни за разбиране, което е полезно за начинаещи програмисти. Поради тази причина, първото разработено приложение с ИИ е конзолно. Като начало са добавени Package CSПролог – Version 6.0.0. След това е дефиниран „class Data“, чиято функция е да създаде интелигентна база данни, с помощта на обект:

```
Prolog = new PrologEngine(persistentCommandHistory: false);.
```

```

public void Add_Data()
{
    Prolog.ConsultFromString(@"
% Семейство растения
family(arabidopsis, 'The Arabidopsis plant is part of
family(palm_tree, palmaceae).
family(apple, rosaceae).
family(wheat, 'Family:poaceae').
family(peas, fabaceae).
family(sunflower, asteraceae).
family(basil, lamiaceae).
family(tomatoes, solanaceae).
family(cactus, cactaceae).
family(orchid, orchidaceae).

% Класификация на частите на растението
vegetative_part(roots).
vegetative_part(stems).
vegetative_part(leaves).

```

Фигура 1. Код с информация за растението

```

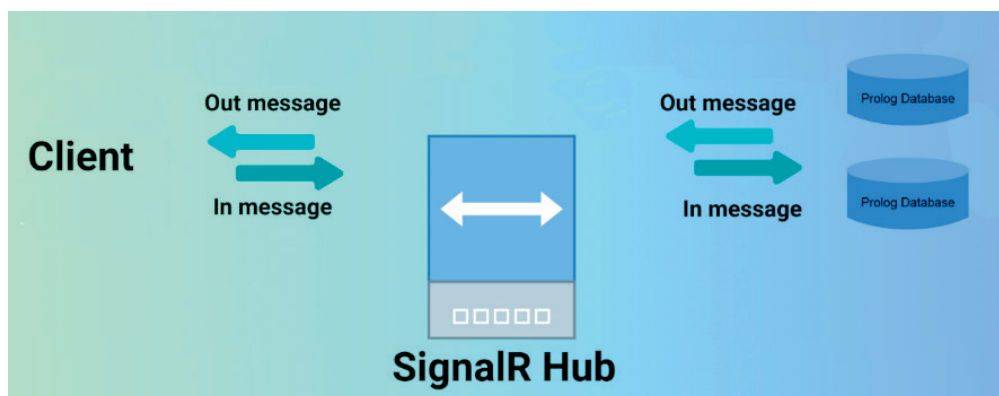
D:\repos\2023-24\Project_Plant - new\Project_Plant\bin\Debug\Project_Plant.exe
Choice (arabidopsis/wheat): arabidopsis
Available genetic modifications for arabidopsis:
- resistance_to_disease
- faster_growth
- herbicide_tolerance
- drought_resistance
That is all.

true (7,875 sec)
Натисни клавиш и се върни към главното меню.

```

Фигура 2. Код със заявки към базата знания

В базата са дефинирани определени факти и правила за растенията. Като допълнение има дефинирани и заявки към базата, които се изпълняват в определен ред (Фигура 2). Идеята обаче за интелигентна база данни, която да комуникира с потребителя не се осъществява напълно. Причините са много: голям брой заявки, които се изпълняват посредством „switch“ конструкция. Това, от своя страна, води до загуба на смисъла за първоначалната идея, а именно приложението да бъде с изкуствен интелект, който да реагира на нуждите на потребителя. Опитът обаче за създаване на приложение с ИИ е успешен и резултатите са добри. На Фигура 2 е показано също и ефективното изпълнение на заявката към базата със стойност „true“, като в скобите е посочено времето за изпълнение, което е 7,875 секунди.



Фигура 3. Схема за комуникация клиент/база знания

Трети етап – разширение и допълнение. Следваща стъпка е да се направи по-сложно приложение, а именно Пролог Бот, който да отговаря на определени въпроси зададени от потребител. Желанието на разработчиците е само с една заявка потребителя да се свърже с базата и след това да има директна комуникация със системата. Идеята за Пролог Бот е породена от нарастващата популярност на ChatGPT. За целта е създадено уеб-базираното приложение като се използва рамка ASP.net. След проучване е избрана библиотеката SignalR, която е специално за ASP.NET. SignalR предоставя възможност за „real-time“ комуникация между уеб клиенти и сървър. На фигура 3 е представена схема за осъществяване на комуникацията клиент/база знания. След създаването на class Data, който съдържа факти и правила за растенията, следва опит за създаване на още една база данни, която съдържа информация свързана от друга област.

```
private static readonly Dictionary<string, string> userDbContext = new Dictionary<string, string>();
private readonly Data data = new Data();
private readonly Game game = new Game();

0 references
public async Task SendMessage(string user, string message)
{
    string response = ProcessMessage(user, message);

    // Генериране на глас
    byte[] audioData = await SynthesizeSpeechFromController(response);

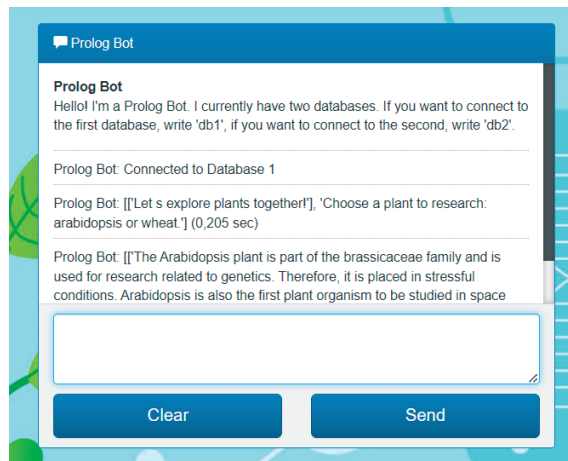
    await Clients.Caller.SendAsync("ReceiveAudio", audioData);
    await Clients.Caller.SendAsync("ReceiveMessage", "Prolog Bot", response);
}

1 reference
private async Task<byte[]> SynthesizeSpeechFromController(string text)
{
```

Фигура 4. Функционалност за генериране на глас

Превключването между класовете се осъществява с помощта на структура от тип Dictionary. Добавена е и функционалност за генериране на глас чрез съответната библиотека, което подобрява комуникацията с ИИ (Фигура 4).

Благодарение на заявката: string? query = \$"word({word}, Message)."; потребителя се „свързва“ към базата. На фигура 5 е показан интерфейсът, както и отговорите, получени от базата данни.



Фигура 5. Интерфейс на Пролог ботът

Четвърти етап – тестване. При тестването на приложението се появяват някои проблеми, които по-късно са отстранени:

- „Зареждане“ на базата на Пролог.

Още в самото начало се появява проблем при тестването и той е отстранен, чрез смяна на позицията на обекта „Prolog“ (Фигура 6).

```
public class Data
{
    3 references
    public PrologEngine Prolog { get; private set; }
    2 references
    public object? Solution1 { get; set; }
    0 references
    public object? Solution2 { get; set; }
    0 references
    public object? Solution3 { get; set; }

    1 reference
    public Data()
    {
        Prolog = new PrologEngine(persistentCommandHistory: false);
        Add_Data();
    }
}
```

Фигура 6. Интерфейс на Пролог ботът

- Друг съществен проблем е изпълнението на предикатите „print“ и „write“. Те не могат да бъдат „извикани“ при работа с ASP.net. Затова дълго се търсят други възможности и накрая е открит компромисен такъв, а именно използването на друг предикат: findall(A, family(arabidopsis, A), Families).

- Важно е да се отбележи, че някои заявки се осъществяват по един начин при Пролог интерпретатора и по друг начин в „Visual Studio“ средата.

Популяризиране на методите за работа с Пролог и възможностите за внедряване на ИИ в образованието. За постигане на тази цел е създаден женски отбор, който е подготвен да разработва СТЕМ проекти, използвайки съответната методология. Учениците се научават да работят в екип, да планират своята работа и да усъвършенстват своите комуникационни умения. Ръководителят на отбора дава идеи и спомага за ефективна екипна работа. В резултат на тези усилия, отборът печели награди и получава покани за участие в различни събития.

Заклучение

Основната идея на проекта е изпълнена. В бъдеще ще се разшири базата от знания и ще се извършат различни изследвания с генетиката на растенията. При разработването на самото приложение са направени много интересни изводи и са открити какви са разликите между императивните и декларативните езици. Важно е да се отбележи, че учениците, за да усвоят различни компетентности свързани с Пролог, е добре да стъпят на някаква основа, в случая знанията им свързани със C#. Това е и една от причините да се направи това изследване. А при програмирането на Пролог – фактите и правилата се „зареждат“ веднъж и след това се използват във всички последващи заявки. Базата със знания свързани с растенията предстои да се обновява от разработчиците.

Дейността по въвеждане на логическо програмиране в средното училище чрез проектно базиран подход може да се комбинира с паралелна, учебна такава, когато учителите могат да идентифицират предмет във всяка дисциплина в рамките на учебната програма, която трябва да покриват. Това е удобно, разбираемо от учениците и се поддава на количествено описание на първоначално състояние, желана цел и набори от възможни действия, които също могат да помогнат за напредъка към целта. И двете дейности имат достъп до програмни инструменти, позволяващи на децата да експериментират с възможни решения около реални данни и следователно да предложат рационално обосновани подобрения на текущата ситуация в тяхното обучение по различните предмети свързвайки ги и със STEM. Проектно базираният подход за въвеждане на логическо програмиране в средното училище предлага също множество предимства, като мотивиране на учениците, развиване на практически умения и задълбочено разбиране на сложни концепции. Чрез внимателно планиране, активна работа и постоянна оценка, учениците могат да придобият важни знания и умения, които ще им бъдат полезни не само в учебния процес, но и в бъдещата им кариера. Очаква се този интердисциплинарен проект да бъде социализиран, когато така овластените ученици взаимодействат със своите съученици, родители и общността си като цяло, показвайки логическите програми, които са разработили, и насърчавайки използването на Пролог като ценен инструмент за програмиране.

References:

1. Proposal for a Regulation Of The European Parliament And Of The Council Laying Down Harmonised Rules On Artificial Intelligence (Artificial Intelligence Act) And Amending Certain Union Legislative Acts. Brussels, 21.4.2021 <https://eur-lex.europa.eu/legal-content/BG/TXT/?uri=CELEX%3A52021PC0206&qid=1717324550503>, last visit: June 2, 2024.
2. Cecchi, LA, Rodríguez, JP, Dahl, V. (2023). Logic Programming at Elementary School: Why, What and How Should We Teach Logic Programming to Children?. In: Warren, DS, Dahl, V., Eiter, T., Hermenegildo, MV, Kowalski, R., Rossi, F. (eds) Prolog: The Next 50 Years. Lecture Notes in Computer Science(), vol 13900. Springer, Cham. https://doi.org/10.1007/978-3-031-35254-6_11.
3. Laurent Cervoni, Julien Brasseur, Jean Rohmer, Simultaneously Teaching Mathematics and Prolog in School Curricula: A Mutual Benefit, In: Warren, DS, Dahl, V., Eiter, T., Hermenegildo, MV, Kowalski, R., Rossi, F. (eds) Prologue: The Next 50 Years. Lecture Notes in Computer Science(), vol 13900. Springer, Cham. https://doi.org/10.1007/978-3-031-35254-6_10, Online ISBN978-3-031-35254-6, Online ISBN978-3-031-35254-6
4. Tabakova-Komsalova, V., Stoyanov, S., Stoyanova-Doycheva, A., Doukovska, L. (2023). Prolog Education in Selected Secondary Schools in Bulgaria. In: Warren, DS, Dahl, V., Eiter, T., Hermenegildo, MV, Kowalski, R., Rossi, F. (eds) Prolog: The Next 50 Years. Lecture Notes in Computer Science(), vol 13900. Springer, Cham. https://doi.org/10.1007/978-3-031-35254-6_12, Online ISBN978-3-031-35254-6, Online ISBN978-3-031-35254-6.
5. Alessandro Dal Palù, Agostino Dovier, Andrea Formisano, and Enrico Pontelli, Prolog Meets Biology, In: Warren, DS, Dahl, V., Eiter, T., Hermenegildo, MV, Kowalski, R., Rossi, F. (eds) Prologue: The Next 50 Years. Lecture Notes in Computer Science(), vol 13900. Springer, Cham. https://link.springer.com/chapter/10.1007/978-3-031-35254-6_26, Online ISBN978-3-031-35254-6, Online ISBN978-3-031-35254-6].
6. Marinković, Vesna. Prolog in Automated Reasoning in Geometry, In: Warren, DS, Dahl, V., Eiter, T., Hermenegildo, MV, Kowalski, R., Rossi, F. (eds) Prolog: The Next 50 Years. Lecture Notes in Computer Science(), vol 13900. Springer, Cham. https://doi.org/10.1007/978-3-031-35254-6_27, Online ISBN978-3-031-35254-6, Online ISBN978-3-031-35254-6.