



ИЗКУСТВЕНИЯТ ИНТЕЛЕКТ В ОБУЧЕНИЕТО ПО ПРОГРАМИРАНЕ

доц. д-р Димитър Минчев

Център по информатика и технически науки,
Бургаски свободен университет

***Резюме:** Изкуственият интелект (ИИ) ще играе все по-важна роля в образователният процес. Въпреки абстрактността на понятията и алгоритмите в областта на програмирането, изучаването им с подкрепата на ИИ е силно препоръчително. Организацията на обучение с използването на ИИ включва създаване на учебни програми, структуриране на учебното съдържание и създаване на система от учебни задачи. Ползите на обучението с ИИ включват развитие на аналитични умения и подготовка за бъдещи професии. Предизвикателство представлява необходимостта от адаптиране на образователните системи към изискванията на дигиталното общество.*

***Ключови думи:** Изкуствен интелект, Преподаване, Обучение, Програмиране.*

ARTIFICIAL INTELLIGENCE IN EDUCATION OF PROGRAMMING

Assoc. Prof. Dimitar Minchev,

Faculty of Computer Science and Engineering, Burgas Free University

***Abstract:** Artificial intelligence (AI) will play an increasingly important role in the educational process. Despite the abstractness of concepts and algorithms in the field of programming, studying them with the support of AI is highly recommended. The organization of learning with the use of AI includes the creation of curricula, structuring the learning content and the creation of a system of learning tasks. The benefits of AI training include developing analytical skills and preparing for future professions. A challenge is the need to adapt education systems to the requirements of a digital society.*

***Keywords:** Artificial intelligence, Teaching, Learning, Programming.*

Въведение

ИИ играе ключова роля в обучението по програмиране, като предоставя нови възможности за обучаемите. Настоящата статия предлага иновативен подход за използване на ИИ в обучението по програмиране, като разделя процеса в пет стъпки: създаване на учебна програма; учебен процес; създаване на задачи; решаване на задачи; генератори на програмен код. Изучаването на ИИ в обучението по програмиране също може да подготви обучаемите за бъдещите предизвикателства в технологичния свят [1,2,3,4,5].

1. Създаване на учебна програма

Първата стъпка от процеса е дефинирането на учебна програма. Тя трябва да определи целите, темите и обхвата на обучението. Програмата трябва да бъде актуална и да отразява последните тенденции в областта. Например можем да генерираме примерна индивидуална учебна програма по програмиране с помоща на инструмента Microsoft Copilot [6] използвайки следния текстов промт:

Create a curriculum to learn programming using C++ language, which includes introduction, variables and data types, conditional operator and loops.

В резултат получаваме като отговор следната примерна индивидуална учебна програма по програмиране на езика C++, която е показана на фиг.1:

1. Introduction to C++:

- Understand what C++ is and its significance.
- Explore the basics of computation and programming.
- Familiarize yourself with the C++ environment.

2. Variables and Data Types:

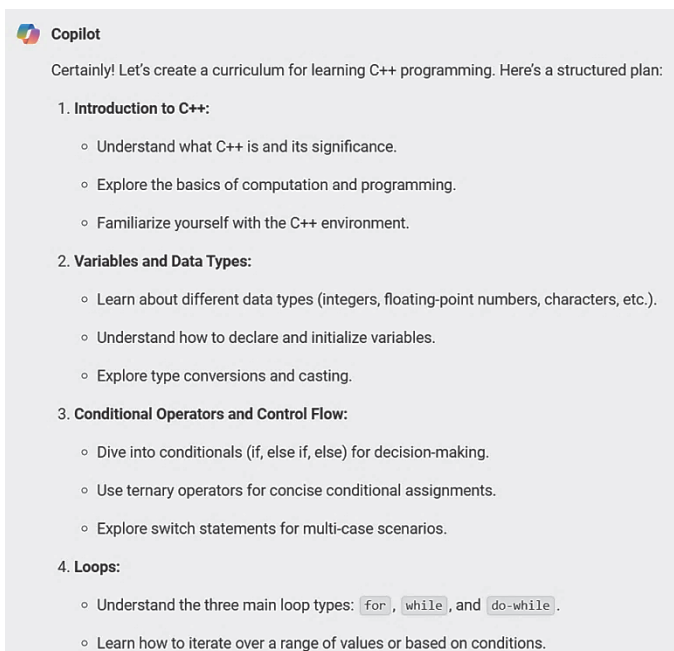
- Learn about different data types (integers, floating-point numbers, characters, etc.).
- Understand how to declare and initialize variables.
- Explore type conversions and casting.

3. Conditional Operators and Control Flow:

- Dive into conditionals (if, else if, else) for decision-making.
- Use ternary operators for concise conditional assignments.
- Explore switch statements for multi-case scenarios.

4. Loops:

- Understand the three main loop types: for, while, and do-while.
- Learn how to iterate over a range of values or based on conditions.



Фиг. 1. Генериране на примерна индивидуална учебна програма по програмиране



2. Учебен процес

Втората стъпка включва избор на подходящи методи и инструменти за обучение. Това може да включва лекции, упражнения, проекти и интерактивни учебни материали. Използването на ИИ за персонализирано обучение може да подобри учебния процес. Например обучаемият желае да се запознае по-добре с тема 3 от примерната индивидуална учебна програма по програмиране, затова той задава следния въпрос:

Explain more about conditional operators and control flow.

В резултат получава примерен отговор на въпроса за конкретната тема, който е показан на фиг. 2.

Copilot

Certainly! Let's dive deeper into **conditional operators** and **control flow** in C++:

1. **Conditional Statements:**

- o **if Statement:**
 - The `if` statement allows you to execute a block of code if a specified condition is true.
 - Example in C++:

```
#include <iostream>
using namespace std;

int main() {
    int a = 5;
    if (a == 5) {
        cout << "a is equal to 5";
    }
    return 0;
}
```
- o **if-else Statement:**
 - The `if-else` statement executes one block of code if a specified condition is true, and another block if the condition is false.

Фиг. 2. Генериране отговор по въпрос за конкретна тема

3. Създаване на задачи

Третата стъпка е създаването на задачи, които да развиват уменията на обучаемите в областта на програмирането. Тези задачи могат да включват алгоритмични проблеми, обработка на данни и създаване на модели. Например продължавайки диалога с помощника, обучаемият желае генерирането на десет задачи по темата, което той заявява със следния текстов промт:

Generate 10 tasks that can be solved by conditional operators and control.

В резултат като отговор се получават примерни десет задачи по програмиране, свързани с проверки, които са показани на фиг. 3.



Copilot

Certainly! Here are ten tasks that can be solved using conditional operators and control flow in C++:

1. Grading System:

- Given a student's score, determine their grade (A, B, C, etc.) based on predefined score ranges.

2. Leap Year Checker:

- Check if a given year is a leap year (divisible by 4 but not divisible by 100, unless divisible by 400).

3. Calculator:

- Implement a simple calculator that performs addition, subtraction, multiplication, or division based on user input.

4. Positive/Negative/Zero Checker:

- Determine whether a given number is positive, negative, or zero.

5. Even/Odd Number Checker:

- Identify whether a given integer is even or odd.

6. Vowel/Consonant Checker:

- Determine if a character is a vowel or a consonant.

7. Maximum of Three Numbers:

- Find the maximum value among three input numbers.

8. Month Name Lookup:

- Given a month number (1 to 12), display the corresponding month name (e.g., 1 → "January").

9. Age Group Classifier:

- Categorize a person's age into groups (child, teenager, adult, senior) based on predefined age ranges.

10. Simple Authentication System:

- Ask the user for a password and grant access if it matches a predefined secret password.

Фиг. 3. Създаване на задачи по конкретна тема

4. Решаване на задачи

Четвъртата стъпка от процеса е активното решаване на задачи. Обучасмите трябва да прилагат наученото в реални сценарии. Например от предложения списък със задачи търсим решението на задачата за проверка дали дадено число е положително, отрицателно или нула, със следния текстов промпт:

Show me and explain the solution of task 4. „Positive/Negative/Zero Checker“.

В резултат като отговор се извежда примерно решение на конкретната задача заедно с кратки пояснения на решението, както е показано на фиг. 4.



Copilot

Certainly! Let's explore a simple solution to check whether a given number is positive, negative, or zero in C++.

```
#include <iostream>
using namespace std;

int main() {
    double number;

    // Take input from the user
    cout << "Enter a Number: ";
    cin >> number;

    // Check conditions
    if (number > 0) {
        cout << "The number is positive." << endl;
    } else if (number < 0) {
        cout << "The number is negative." << endl;
    } else {
        cout << "The number is zero." << endl;
    }

    return 0;
}
```

Here's how the program works:

1. We declare a `double` variable called `number` to store the user input.
2. The user is prompted to enter a number.
3. We use `if`, `else if`, and `else` statements to check the conditions:

Фиг. 4. Примерно решение на конкретна задача

5. Генератори на програмен код

Последната стъпка е използването на генератори на програмен код, които използват ИИ. Тези инструменти могат да помогнат на обучаемите да създават ефективен и качествен код. Например използвайки инструмента Microsoft GitHub Copilot [7] във Microsoft Visual Studio [8], обучаемият може да поиска генериране на цялостен програмен фрагмент за решаване на конкретен проблем. Ето един пример:

Write me a class that can do CRUD operation on a MSSQL database called Students.

В резултат от тази заявка се генерира цялостен програмен фрагмент за решаване на конкретния проблем, който е показан на фиг. 5.



```
using System;
using System.Data;
using System.Data.SqlClient;

public class StudentRepository
{
    private string connectionString = "Data Source=(local);Initial Catalog=Students;I

    public DataTable GetAllStudents()
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            con.Open();
            SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM Student", con);
            DataTable dt = new DataTable();
            da.Fill(dt);
            return dt;
        }
    }

    public void AddStudent(string name, int age)
    {
        using (SqlConnection con = new SqlConnection(connectionString))
        {
            con.Open();
            SqlCommand cmd = new SqlCommand("INSERT INTO Student (Name, Age) VALUES (
            cmd.Parameters.AddWithValue("@Name", name);
            cmd.Parameters.AddWithValue("@Age", age);
            cmd.ExecuteNonQuery();
        }
    }
}
```

Фиг. 5. Генериране на цялостен програмен фрагмент за решаване на конкретен проблем

Заклучение

Настоящата публикация предлага иновативен подход за използване на ИИ в обучението по програмиране, като разделя процеса в пет стъпки: създаване на учебна програма; учебен процес; създаване на задачи; решаване на задачи; генератори на програмен код.

Литература

- [1]. Изкуственият интелект в образованието и науката, <https://www.mon.bg/nfs/2020/07/mon%20ai%20doc.pdf>
- [2]. Изкуственият интелект – възможности и перспективи за интеграция в българската образователна сфера, <https://pedagogicnews.uni-guse.bg/изкуственият-интелект-възможности/>
- [3]. Учебните задачи в обучението по изкуствен интелект, https://www.edutechjournal.org/wp-content/uploads/2021/01/1_2020_233-240.pdf
- [4]. ИЗУЧАВАНЕ НА ИЗКУСТВЕН ИНТЕЛЕКТ И ЛОГИЧЕСКО ПРОГРАМИРАНЕ В СРЕДНОТО УЧИЛИЩЕ, http://srem2020.fmi-plovdiv.org/wp-content/uploads/2020/10/6_5_Glushkova_Stoyanov_Malinova.pdf
- [5]. Програмиране на изкуствен интелект - как се прави?, <https://advanceacademy.bg/blog/programirane-na-izkustven-intelekt-kak-se-pravi>
- [6] Microsoft Copilot, <https://copilot.microsoft.com/>
- [7] Microsoft GitHub Copilot, <https://github.com/features/copilot>
- [8] Microsoft Visual Studio, <https://visualstudio.microsoft.com>