

# РОЗРОБКА ЕЛЕКТРОННОГО КУРСУ "ВСТУП ДО ПРОГРАМУВАННЯ"

Виконав:

студент МП КН-2

Семененко І. В.

Науковий  
керівник:

проф. Глибовець  
М. М.

# Актуальність теми

- Зростаюча потреба у ЕЛЕКТРОННОМУ НАВЧАННІ (ЕН)
- Відсутність ефективного програмного інструментаря розробки та супроводу електронного курсу (ЕК) «вільного застосування» з широкими можливостями інтеграції використання різноманітних засобів мультимедіа та тестування знань

# Мета та завдання роботи

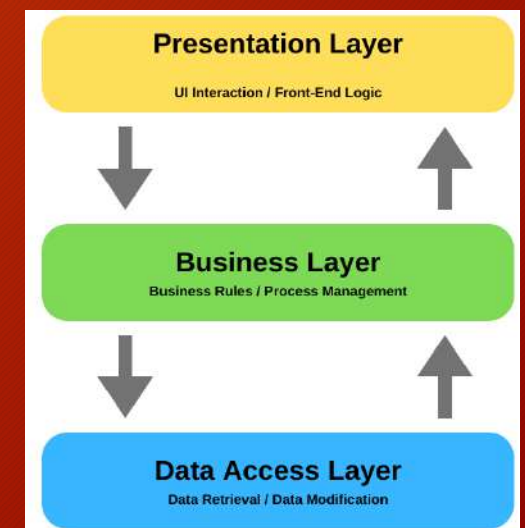
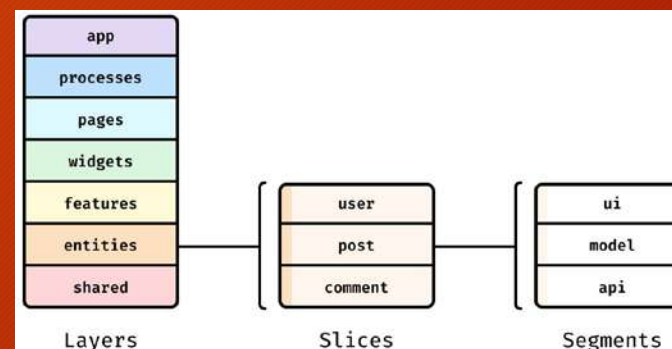
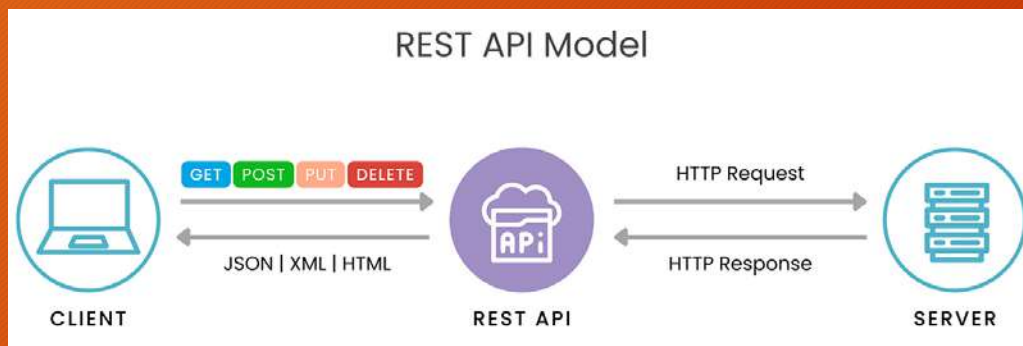
Мета: Розробка та впровадження ефективного програмного інструментарію розробки та супроводу електронного з широкими можливостями інтеграції використання різноманітних засобів мультимедіа та тестування знань на прикладі створення інтерактивного електронного курсу "Вступ до програмування", адаптованого для учнів загальноосвітніх шкіл, з акцентом на оригінальний україномовний контент та можливістю простого розширення навчального матеріалу викладачами.

## Завдання роботи:

- Проаналізувати інструментарій створення ЕК та існуючі онлайн-платформи ЕН.
- Розробити архітектуру програмної системи та структурні основи україномовного інтерактивного ЕК.
- Реалізувати бекенд та фронтенд ЕК курсу "Вступ до програмування" з інтерактивним контентом.
- Реалізувати функціонал для додавання кастомних питань викладачами.
- Протестувати розроблений програмний продукт та оцінити його ефективність.

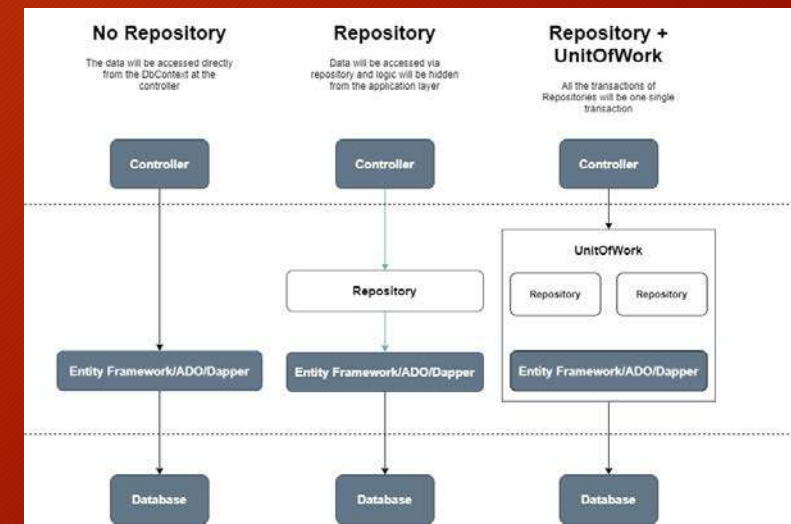
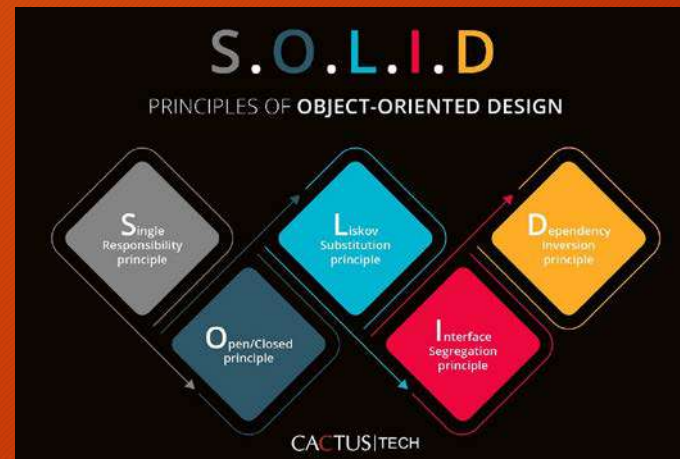
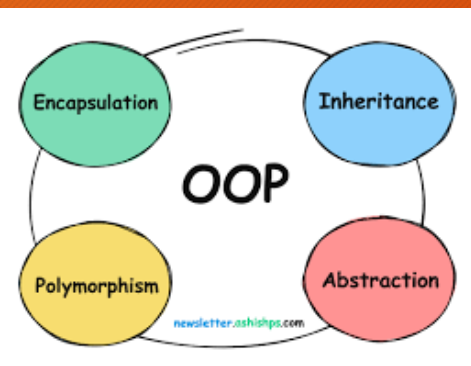
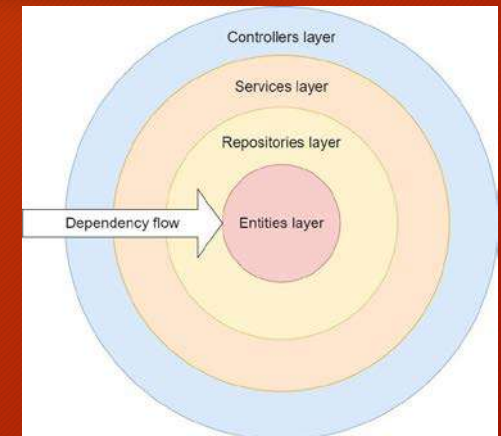
# Архітектура системи та обрані технології

- Клієнт-серверна модель
- N-Tier архітектура
- REST API
- Feature-Sliced Design



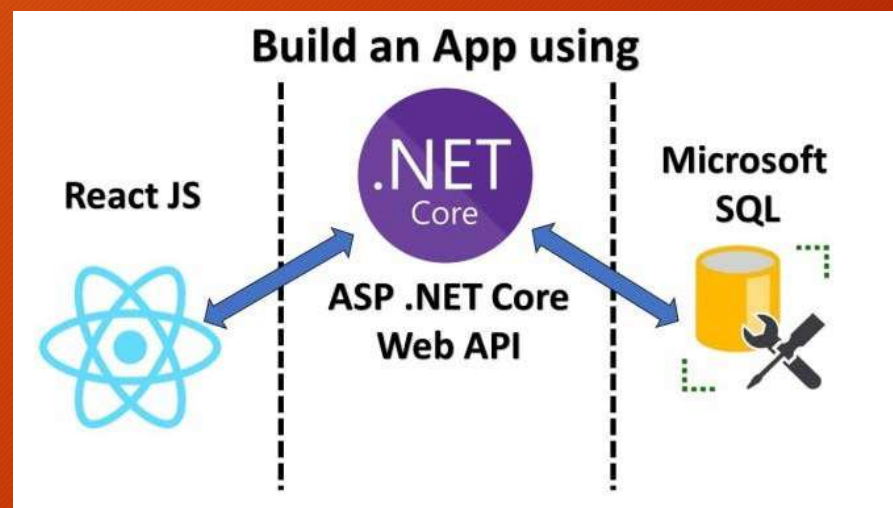
# Принципи Проектування та Патерни

- SOLID
- Об'єктно-орієнтоване програмування
- Ключові патерни
  - Service Layer
  - Repository & Unit of Work
  - Dependency Injection
  - Controller Pattern



# Технологічний стек

- Бекенд: .NET 8.0 (ASP.NET Core Web API)
- База Даних: MS SQL Server
- Фронтенд: Next.js, React, TypeScript



# Структура PL

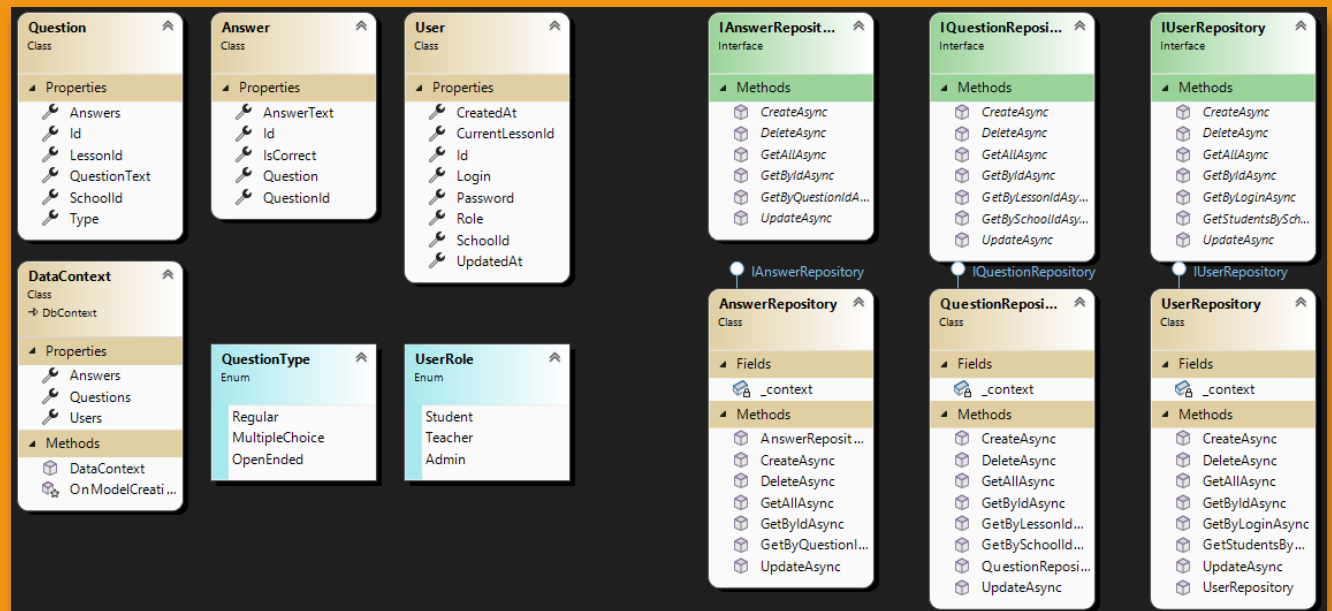
```
src
├── app
│   ├── actions
│   │   ├── auth.ts
│   │   └── logout.ts
│   ├── api \tests \[id]
│   ├── lesson \[...params]
│   │   ├── layout.tsx
│   │   ├── page.tsx
│   │   └── login
│   │       ├── page.tsx
│   │       ├── student \add
│   │       └── test
│   ├── # globals.css
│   ├── layout.tsx
│   └── page.tsx
├── components \Nav
├── pages
├── shared
│   ├── api
│   ├── config
│   ├── TS backend.ts
│   ├── TS content.ts
│   ├── TS roles.ts
│   ├── lib
│   │   ├── auth.ts
│   │   └── lessonNavigation.ts
│   ├── types
│   │   ├── auth.ts
│   │   └── domain.ts
│   ├── ui
│   │   ├── Sidebar
│   │   └── Loader.tsx
│   └── utils
│       └── TS getUserFromToken.ts
└── TS middleware.ts
```

# Структура BL

The screenshot displays the following classes and interfaces:

- AuthController** (Class, ControllerBase):
  - Fields: \_userService
  - Methods: AuthController, Login, VerifyRole
- UserController** (Class, ControllerBase):
  - Fields: \_userService
  - Methods: CreateTestUsers, CreateUser, DeleteUser, GetAllUsers, GetStudentsBy..., GetUserId, GetUserByLogin, UpdateUser, UserController
- QuestionController** (Class, ControllerBase):
  - Fields: \_questionService
  - Methods: CheckAnswers, CreateQuestion, GetQuestionBy..., QuestionContr...
- TestQuestionCo...** (Class, ControllerBase):
  - Fields: \_questionService
  - Methods: AddFifthBatch..., AddFirstBatch..., AddFourthBat..., AddSecondBat..., AddSixthBatch..., AddThirdBatch..., TestQuestionCo...
- IQuestionService** (Interface):
  - Methods: CheckAnswersAsync, CreateQuestionAsync, GetQuestionsBy...
- QuestionService** (Class, IQuestionService):
  - Fields: \_answerRepository, \_questionRepository
  - Methods: CheckAnswers..., CreateQuestion..., GetQuestions..., MapToResponse, NormalizeText, QuestionService
- IUserService** (Interface):
  - Methods: CreateUserAsync, DeleteUserAsync, GetAllUsersAsync, GetStudentsBy..., GetUserByIdAsync, GetUserByLoginAsync, UpdateUserAsync, VerifyPasswordAsync, VerifyUserAsync
- UserService** (Class, IUserService):
  - Fields: \_configuration, \_userRepository
  - Methods: CreateUserAsync, DeleteUserAsync, GenerateJwtToken, GetAllUsersAsync, GetStudentsBy..., GetUserByIdAs..., GetUserByLogi..., HashPassword, MapToResponse, UpdateUserAsy..., UserService, VerifyPassword..., VerifyUserAsyn...
- AnswerResponse** (Class):
  - Properties: AnswerText, Id, IsCorrect
- AnswerToCheck** (Class):
  - Properties: Answers, AnswerText, Type
- CheckAnswersR...** (Class):
  - Properties: Answers
- CreateAnswerRe...** (Class):
  - Properties: AnswerText, IsCorrect
- CreateQuestion...** (Class):
  - Properties: Answers, LessonId, QuestionText, SchoolId, Type
- CreateUserReq...** (Class):
  - Properties: Login, Password, Role, SchoolId
- LoginRequest** (Class):
  - Properties: Login, Password
- LoginResponse** (Class):
  - Properties: CurrentPage, Role, SchoolId, Token
- QuestionRespon...** (Class):
  - Properties: Answers, Id, LessonId, QuestionText, SchoolId, Type
- TestAnswer** (Class):
  - Properties: AnswerId, Text
- TestQuestion** (Class):
  - Properties: Answers, Question, QuestionId, QuestionType
- TestSubmission** (Class):
  - Properties: Answer, QuestionId, QuestionType
- UpdateUserReq...** (Class):
  - Properties: CurrentLessonId, Login, Password, Role, SchoolId
- UserResponse** (Class):
  - Properties: CreatedAt, CurrentLessonId, Id, Login, Password, Role, SchoolId, UpdatedAt

# Структура DAL



# Структура бази даних

Answers
Id
AnswerText
IsCorrect
QuestionId



Questions
Id
QuestionText
Type
LessonId
SchoolId

Users
Id
Login
Password
Role
CurrentLessonId
SchoolId
CreatedAt
UpdatedAt

# Типи доступних питань для створення

Питання 1 Видалити питання

Приведіть загальну характеристику JavaScript

Тип питання  
Одна відповідь

Варіанти відповідей

- JavaScript — це мова програмування для серверів  Правильна
- JavaScript — це компільована мова, яку використовують тільки в мобільних з  Правильна
- JavaScript — це мова розмітки для створення вебсторінок  Правильна
- JavaScript — це динамічна, інтерпретована мова програмування, яку зазвичай  Правильна

[+ Додати варіант](#)

Питання 10 Видалити питання

Яке є ключове слово для визначення незмінної змінної?

Тип питання  
Відкрите питання

Правильна відповідь  
const



Питання 2 Видалити питання

Як в браузері почати писати код JavaScript?

Тип питання  
Кілька відповідей

Варіанти відповідей

- CTRL/SHIFT// для Windows/Linux  Правильна
- COMMAND/OPTION// для Mac OS  Правильна
- Встановити спеціальний плагін для браузера  Правильна
- Запустити окрему програму для написання коду  Правильна

[+ Додати варіант](#)

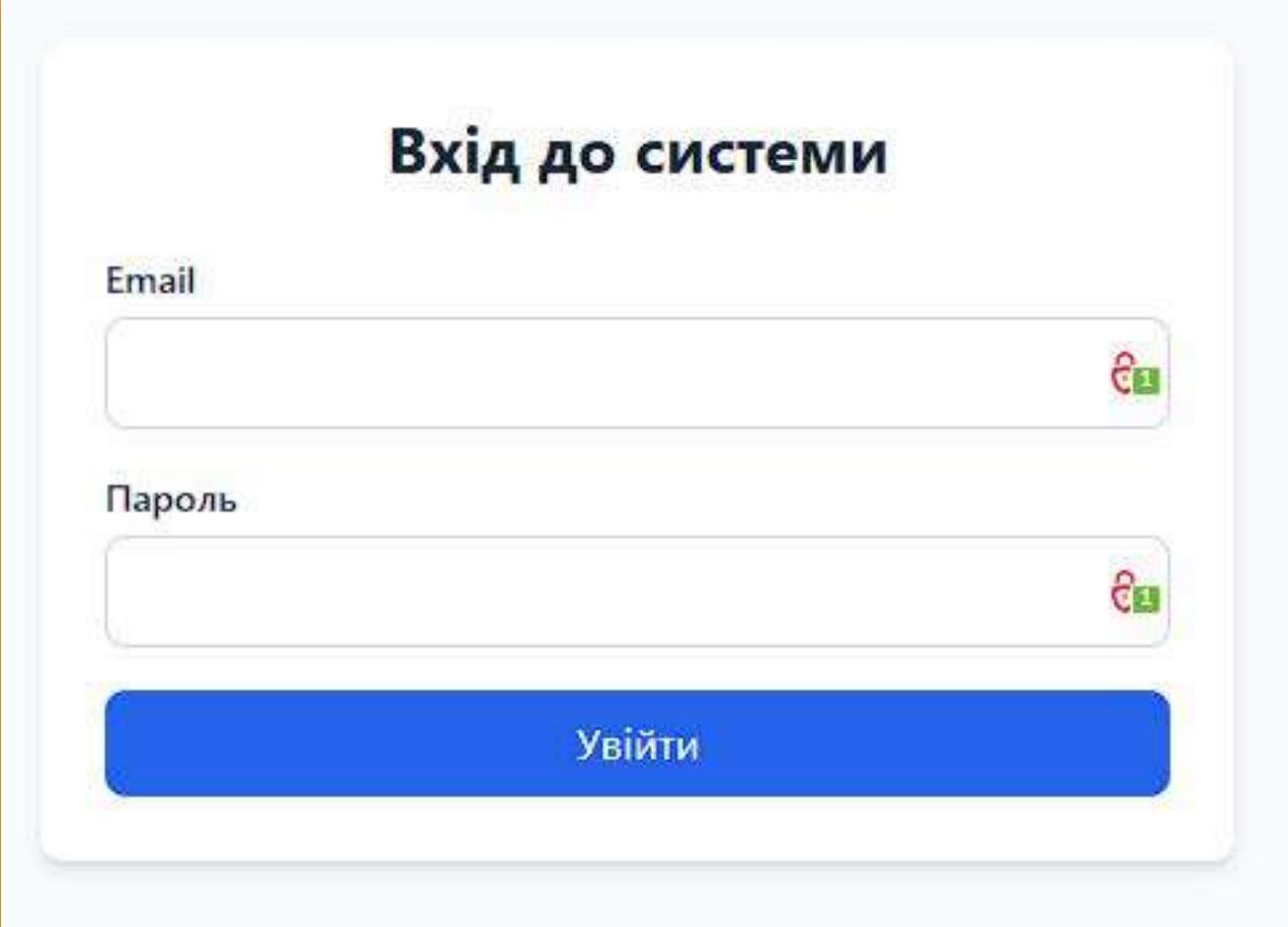
# Приклади інтерфейсу програми

**Вхід до системи**

Email

Пароль

Увійти

A login form titled "Вхід до системи" (Login to system) is displayed on a white background. It features two input fields: "Email" and "Пароль" (Password). Each field has a small red padlock icon with a green "1" next to it, indicating a security warning. Below the fields is a blue button labeled "Увійти" (Login).

# Приклади інтерфейсу програми

## Уроки JavaScript

### АЛГОРИТМИ І ОБЧИСЛЕННЯ

### ПОЧАТКИ ПРОГРАМУВАННЯ

Вступ до JavaScript

Коротка характеристика

Початок роботи

Перші кроки

Зуваження щодо синтаксису

Завдання для самоконтролю

Дані

Оператори управління

Функції

Додаткові можливості та  
інструментарій

JavaScript – особливий  
представник сімейства  
клієнтських скриптів

Додати  
студента

Вийти

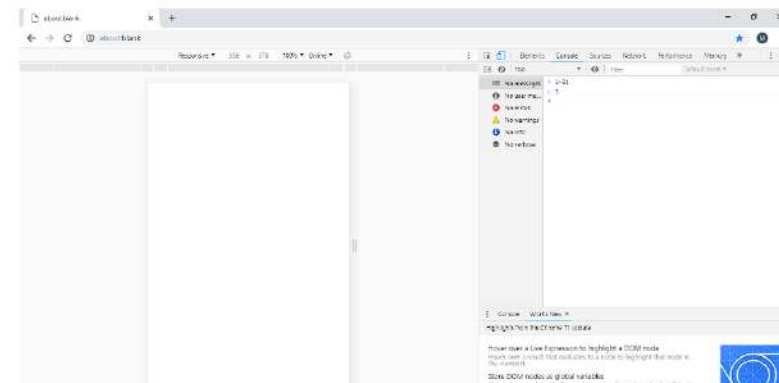
## Перші кроки

Пам'ятаючи, що ми маємо справу із інтерпретатором (а ми вже знаємо, що він перекладає команду програми в машинну команду, готову до виконання) спробуємо одразу реалізувати основні операції калькулятора.

Почнемо з операції додавання. Наприклад, ми хочемо дізнатись результат операції «1+2». Для цього давайте виконаємо таку послідовність кроків.

- Введемо в консоль на місце курсора, що «блимає» таке: 1+2; Зверніть увагу на «;».
- Натисніть клавішу ENTER.

Ви побачили на екрані результат 3 у вигляді, який представлено на рис. 2.5.



# Приклади інтерфейсу програми

ПОЧАТКИ  
ПРОГРАМУВАННЯ

- Вступ до JavaScript
- Дані
- Оператори управління
  - Огляд операторів управління
  - Послідовне управління
  - Групування
  - Розгалуження**
  - Варианти
  - Ітерація
- Інтерфейс: браузер alert, prompt, confirm
- Завдання для самоконтролю
- Функції
- Додаткові можливості та інструментарій
- JavaScript – особливий представник сімейства клієнтських скриптів

Вийти

може бути замінений на еквівалентний по суті оператор:

```
(( a < b && ( a < c ) && ( a < b ) && ( b < c > a ) ) ?  
console.log("трикутник можна побудувати") ;  
console.log("трикутник не можна побудувати");
```

Зробимо цю заміну в нашій програмі та запустимо її на виконання.

```
var a = Number(prompt("Введіть а: "));  
console.log("a=" + a);  
var b = Number(prompt("Введіть b: "));  
console.log("b=" + b);  
var c = Number(prompt("Введіть c: "));  
console.log("c=" + c);  
  
(( a < b && ( a < c ) && ( a < b ) && ( b < c > a ) ) ?  
console.log("трикутник можна побудувати") ;  
console.log("трикутник не можна побудувати");  
  
// a=1  
// b=0  
// c=12  
// трикутник не можна побудувати
```

Тепер Андрійку та його друзям можна перевіряти програмно 20 прикладів домашнього завдання!!!!

← Попередній урок

Наступний урок →

# Приклади інтерфейсу програми

Уроки JavaScript

АЛГОРИТМИ І ОБЧИСЛЕННЯ

ПОЧАТКИ ПРОГРАМУВАННЯ

Вийти

## Тест до розділу 2.3

Яка ключова відмінність між циклами 'while' та 'do/while'?

- 'while' виконується швидше, ніж 'do/while'
- Тіло циклу 'do/while' виконується хоча б один раз, тоді як тіло 'while' може не виконатись жодного разу
- Цикл 'while' перевіряє умову після виконання тіла, а 'do/while' — до
- Між ними немає суттєвої різниці

Які є оператори циклу?

- for
- while
- if
- do/while

Який оператор використовується для негайного виходу з циклу або конструкції switch?

Введіть вашу відповідь...

# Висновки

- Розроблено ефективний програмний інструментарій (застосунок) розробки та супроводу ЕК з широкими можливостями інтеграції використання різноманітних засобів мультимедіа, ШІ та тестування знань
- Інструментарій протестовано на прикладі створення інтерактивного ЕК "Вступ до програмування", адаптованого для учнів загальноосвітніх шкіл, з акцентом на оригінальний україномовний контент та можливістю простого розширення навчального матеріалу викладачами.
- Результати роботи впроваджені в екосистему проекту Digital University – Open Ukrainian Initiative (DigiUni) 101129236-DigiUni-ERASMUS-EDU-2023-CBHE

Дякую за увагу!

