WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA
z siedzibą w Rzeszowie

# Programming Languages Project:

## Family Feud Quiz

**By Students:**

Aikyn Aryn w65387, Rustem Kuntuganov w66256

# Family Feud Console Game

## Overview

This C# console application is an interactive game based on the popular TV show "Family Feud". Players are presented with a series of questions and must guess the most popular answers to earn points. The game showcases object-oriented programming principles, basic game logic, and console-based user interaction.

## Key Features

- Multiple-choice questions with point-weighted answers
- Three attempts allowed per question
- Running score calculation throughout the game
- Final score evaluation with personalized feedback
- Modular design for easy addition of new questions

## Game Structure

### Question Class

The game uses a custom `Question` class to represent each question:

- `Text`: string property holding the question prompt
- `Answers`: List of tuples, each containing an answer string and its point value

### Questions Repository

- Questions are stored in a static `List<Question>` named `Questions`
- Each question has 5 possible answers with varying point values
- Current implementation includes 4 sample questions

## Gameplay

### Initialization

1. The game starts with a welcome message
2. Total score is initialized to 0

### Main Game Loop

For each question in the `Questions` list:

1. The question is displayed to the player
2. The player has up to 3 attempts to provide a correct answer
3. Input is read from the console and processed

### Answer Processing

1. Player input is trimmed and converted to lowercase for comparison
2. The input is checked against the list of valid answers
3. If a match is found:
   - The corresponding points are added to the total score
   - A success message is displayed
   - The game moves to the next question

4. If no match is found:
    - An error message is displayed
    - The player can try again (up to 3 attempts)

### Score Tracking

- Points are accumulated throughout the game
- The total score is updated after each correct answer

## End Game

After all questions have been answered:

1. The final total score is displayed
2. Feedback is provided based on the score:
    - 100+ points: "Great job! You really know your stuff!"
    - 50-99 points: "Not bad! You did pretty well!"
    - 0-49 points: "Better luck next time!"
3. A thank you message is displayed

## Technical Details

### Language and Framework

- Written in C# targeting the .NET framework
- Uses core C# features including LINQ, string manipulation, and console I/O

### Data Structures

- Uses List<T> for storing questions and answers
- Utilizes tuples for pairing answers with their point values

### LINQ Usage

- FirstOrDefault() is used to find matching answers efficiently

### Error Handling and Input Validation

- Checks for null or whitespace input
- Allows retry without penalty for invalid input

### Customization

- The Questions list can be easily expanded or modified to change the game content

## Potential Enhancements

- Add more questions to increase replay value
- Implement a two-player mode
- Add difficulty levels
- Store high scores
- Randomize question order

## Conclusion

This Family Feud Console Game demonstrates fundamental programming concepts while providing an engaging user experience. Its modular design allows for easy expansion and modification, making it an excellent base for further development or learning exercises