Emergency Response Simulation - Short Report

Overview

The **Emergency Response Simulation** is a C# console application that models how different emergency units respond to various incidents. It simulates multiple rounds where incidents occur randomly, and appropriate emergency units must handle them to gain points.

System Structure

1. Abstract Class: `EmergencyUnit`

- Defines common properties and abstract methods for emergency units.
- **Properties**:
  - `Name`: The unit's name.
  - `BaseSpeed`: The base response speed.
- **Abstract Methods**:
  - `CanHandle(string incidentType)`: Determines if the unit can respond to a certain incident type.
  - `CalculateResponseTime(string incidentLevel)`: Calculates the response time based on priority.
  - `RespondToIncident(Incident incident)`: Outputs the unit's response action.

2. Derived Classes

- **`Police`**
  - Handles: Crime, Public Disturbance, Traffic Accident.
- **`Firefighter`**
  - Handles: Fire, Hazardous Material.
- **`Ambulance`**
  - Handles: Medical, Traffic Accident.

Each class customizes how it calculates response time and what action it performs during an incident.

3. Incident Class

- **Properties**:
  - `Type`: Type of incident (e.g., Crime, Fire).
  - `Location`: Location of the incident.
  - `Level`: Priority level (High, Medium, Low).
- **Purpose**: Represents incidents in the simulation.

4. EmergencySimulation Class

- Manages the full simulation process.
- **Responsibilities**:
    - Initializes emergency units.
    - Runs a 5-round simulation.
    - Collects user input for incident type, priority, and location.
    - Finds the suitable emergency unit to respond.
    - Updates and displays the score based on correct or failed responses.

5. Program Class

- Entry point of the application.
- Starts the simulation by invoking `RunSimulation()` from `EmergencySimulation`.

Features

- User selects the type of incident, its priority, and location.
- Points are awarded based on the incident's priority (higher priority, more points).
- If no suitable unit is available, points are deducted.
- Outputs detailed information about each incident and response.

Challenges Faced During Development

During the development of the Emergency Response Simulation, several challenges were encountered:

- **Handling User Input Validation**: Ensuring that invalid inputs (such as wrong menu selections or empty location entries) were correctly handled without crashing the program.
- **Designing Flexible Inheritance**: Creating a good abstract structure that allows easy addition of new types of emergency units if needed.
- **Managing Response Time Calculations**: Each unit type had a slightly different way to adjust response time based on incident priority, requiring careful management.
- **Keeping the Code Organized**: As the project grew, keeping the classes and responsibilities clearly separated was necessary to maintain readability.
- **Balancing the Scoring System**: Setting fair and meaningful point values to reflect the difficulty and importance of incidents.

Conclusion

This project demonstrates the use of:

- Abstract classes and inheritance for code reusability.
- Polymorphism to allow different emergency units to behave differently.

- Basic console input/output for user interaction.
- Simple game mechanics with a scoring system based on decision-making.

The application successfully simulates a simplified real-world emergency response system where different units have specialized roles.