- ***DEBRE BERHAN UNIVERSITY***
- College of Computing
- Department of information system
-
- Individual Assignment IR
-

Name : Abebe Taye Id: 0671/13

Emergency Response Simulator - Report

1. Overview

The Emergency Response Simulator is a **C# console application** designed to simulate emergency units (**Police, Firefighters, Ambulances**) responding to **random incidents** in a city. The project demonstrates **Object-Oriented Programming (OOP) principles**, including **abstraction, inheritance, polymorphism, and encapsulation**.

2. Applied OOP Concepts

✅ Abstraction

The `Emergency Unit` class is **abstract**, defining common properties (`Name`, `Speed`) and methods (`Can Handle()` and `RespondToIncident()`) for all emergency units.

✅ Inheritance

The specific emergency units—**Police, Firefighter, Ambulance**—inherit from `emergency Unit`. Each subclass overrides methods to provide **unique behavior**.

✅ Polymorphism

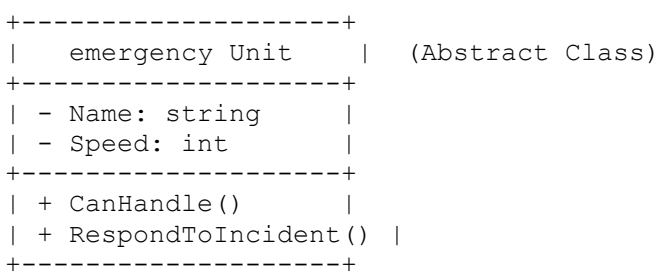
The `Can Handle()` method is overridden in each unit subclass to ensure only the **appropriate** unit responds to an incident type.

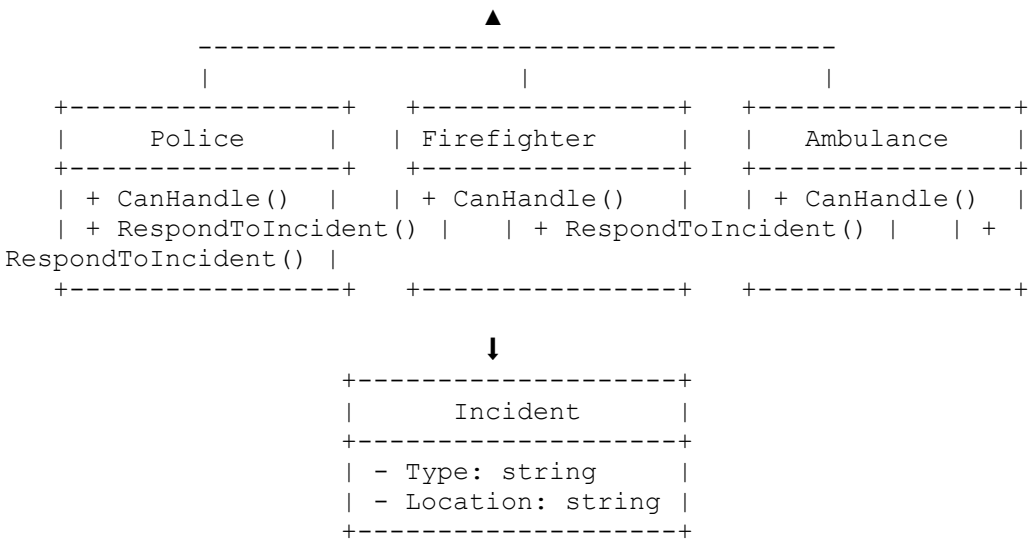
✅ Encapsulation

The `Incident` and `emergency Unit` classes encapsulate properties (`Type`, `Location`, `Name`, `Speed`) using **getter/setter** methods.

3. Class Diagram

Below is the **class structure representation** of the simulator:





4. Functionality & Simulation Logic

✅ Game Loop (5 Rounds)

- Generates **random incidents** (Crime, Fire, Medical).
- Finds the correct **Emergency Unit** using `CanHandle()`.
- Calls `RespondToIncident()` to handle the emergency.
- Updates **scores** (+10 for correct response, -5 for failure).

✅ Console Output Example

```
--- Turn 1 ---
Incident: Fire at Park
Firefighter Unit 1 is extinguishing a fire at Park.
Current Score: 10
```

5. Lessons Learned & Challenges

Key Insights

- **Using abstraction** keeps the base class flexible.
- **Polymorphism** allows dynamic unit selection.
- **Encapsulation** protects properties from accidental modification.

Challenges

- Managing **random incidents** effectively.
- Ensuring **proper object interactions** within the game loop.

6. Future Enhancements (Bonus)

- ✅ Add **more emergency unit types** (e.g., Coast Guard, Search & Rescue).
- ✅ Allow **manual unit selection** instead of automatic assignment.
- ✅ Introduce **difficulty levels** with response time tracking.