Part 1: UML class diagram

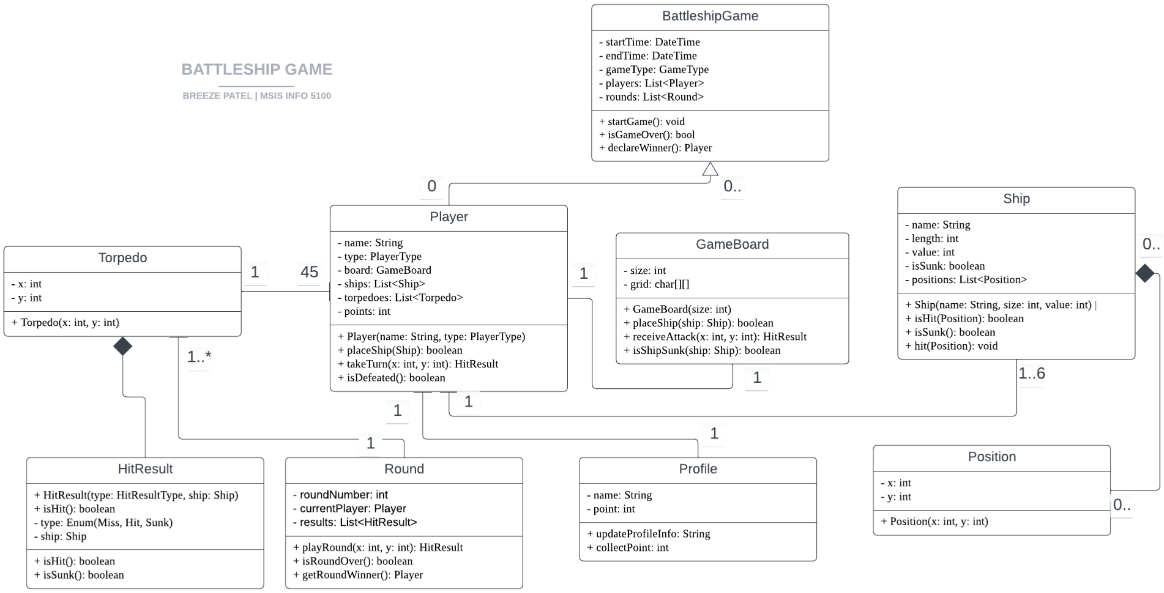
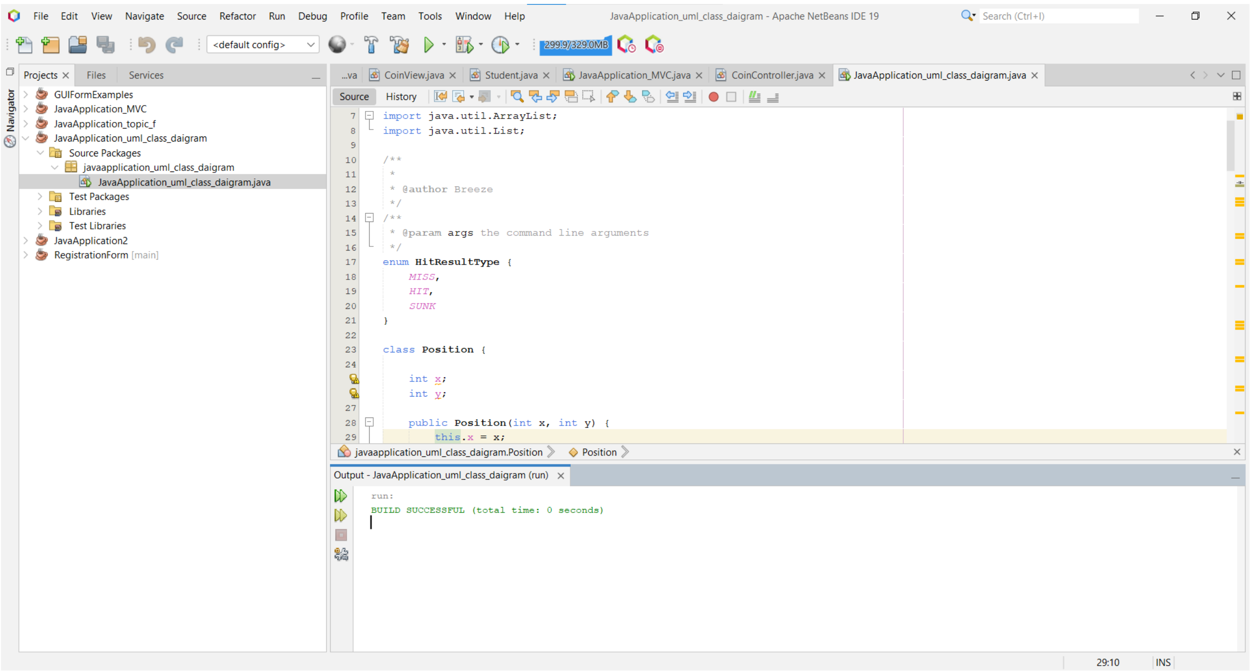


Fig: 1 Battleship Game UML Class Diagram.

Part 2 Java Code:

```
package javaapplication_uml_class_daigram;

import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Breeze
 */
/**
 * @param args the command line arguments
 */
enum HitResultType {
    MISS,
    HIT,
    SUNK
}

class Position {

    int x;
    int y;

    public Position(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

class Ship {

    String name;
    int length;
    int value;
    boolean isSunk;
    List<Position> positions;

    public Ship(String name, int length, int value) {
        this.name = name;
        this.length = length;
        this.value = value;
        this.isSunk = false;
        this.positions = new ArrayList<>();
    }

}

class GameBoard {

    int size;
    char[][] grid;

    public GameBoard(int size) {
        this.size = size;
        this.grid = new char[size][size];

    }
}

class Torpedo {

    int x;
    int y;

    public Torpedo(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

class HitResult {

    HitResultType type;
    Ship ship;
    public HitResult(HitResultType type, Ship ship) {
        this.type = type;
        this.ship = ship;
    }

    public boolean isHit() {
        return type == HitResultType.HIT;
    }

    public boolean isSunk() {
        return type == HitResultType.SUNK;
    }
}

class Player {

    String name;
    List<Torpedo> torpedoes;
    List<Ship> ships;
    int points;

    public Player(String name) {
        this.name = name;
        this.torpedoes = new ArrayList<>();
        this.ships = new ArrayList<>();
        this.points = 0;
    }
}

class Round {

    int roundNumber;
    Player currentPlayer;
    List<HitResult> results;

    public Round(int roundNumber, Player currentPlayer) {
        this.roundNumber = roundNumber;
        this.currentPlayer = currentPlayer;
        this.results = new ArrayList<>();
        // Implement methods for playing a round and determining the winner
    }
}

class BattleshipGame {

    List<Player> players;
    List<Round> rounds;
    HitResultType[][] gameGrid;
    public BattleshipGame() {
        this.players = new ArrayList<>();
        this.rounds = new ArrayList<>();

    }
}

public class JavaApplication_uml_class_daigram {

    public static void main(String[] args) {
        BattleshipGame game = new BattleshipGame();
    }
}
```



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
JavaApplication_uml_class_daigram - Apache NetBeans IDE 19
Search (Ctrl+I)
<default config>
299.9/329.0MB
Projects Files Services
GUIFormExamples
JavaApplication_MVC
JavaApplication_topic_f
JavaApplication_uml_class_daigram
Source Packages
javaapplication_uml_class_daigram
JavaApplication_uml_class_daigram.java
Test Packages
Libraries
Test Libraries
JavaApplication2
RegistrationForm [main]
...va CoinView.java Student.java JavaApplication_MVC.java CoinController.java JavaApplication_uml_class_daigram.java
Source History
javaapplication_uml_class_daigram.Position Position
Output - JavaApplication_uml_class_daigram (run)
29:10 INS

```
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Breeze
 */
/**
 * @param args the command line arguments
 */
enum HitResultType {
    MISS,
    HIT,
    SUNK
}

class Position {

    int x;
    int y;

    public Position(int x, int y) {
        this.x = x;
```

```
run:
BUILD SUCCESSFUL (total time: 0 seconds)
```

Fig 2: Above mentioned code, run Successfully.