

Supervised by : Dr.laith shehab

presented by : Ahmad Al-bishawi & Mohammad Al-shareef

Football Team Management System

IN JAVA



introduction :

Welcome to our presentation on the Football Team Management System. This project is designed to streamline the management of a football team, providing an efficient way to handle player details, coaching staff, stadium information, and team achievements.



Imports :

1. **import java.util.ArrayList;**

- ArrayList is used in the Team class to maintain a list of Player objects.

2. **import java.util.Scanner;**

- In the Main class, Scanner is used to read user input for team details, player details, coach details, stadium details, and other menu choices



CLASSES

1. PLAYER (MAIN CLASS)

- Represents a generic player with attributes like name, salary, contract length, and jersey number.
- Abstract method: `getPlayerType()` which must be implemented by subclasses.
- Constructor: Initializes name, salary, contract length, and jersey number.
- Getters: Provide access to the player's attributes.



CLASSES

1. GOALKEEPER (DERIVED CLASS)

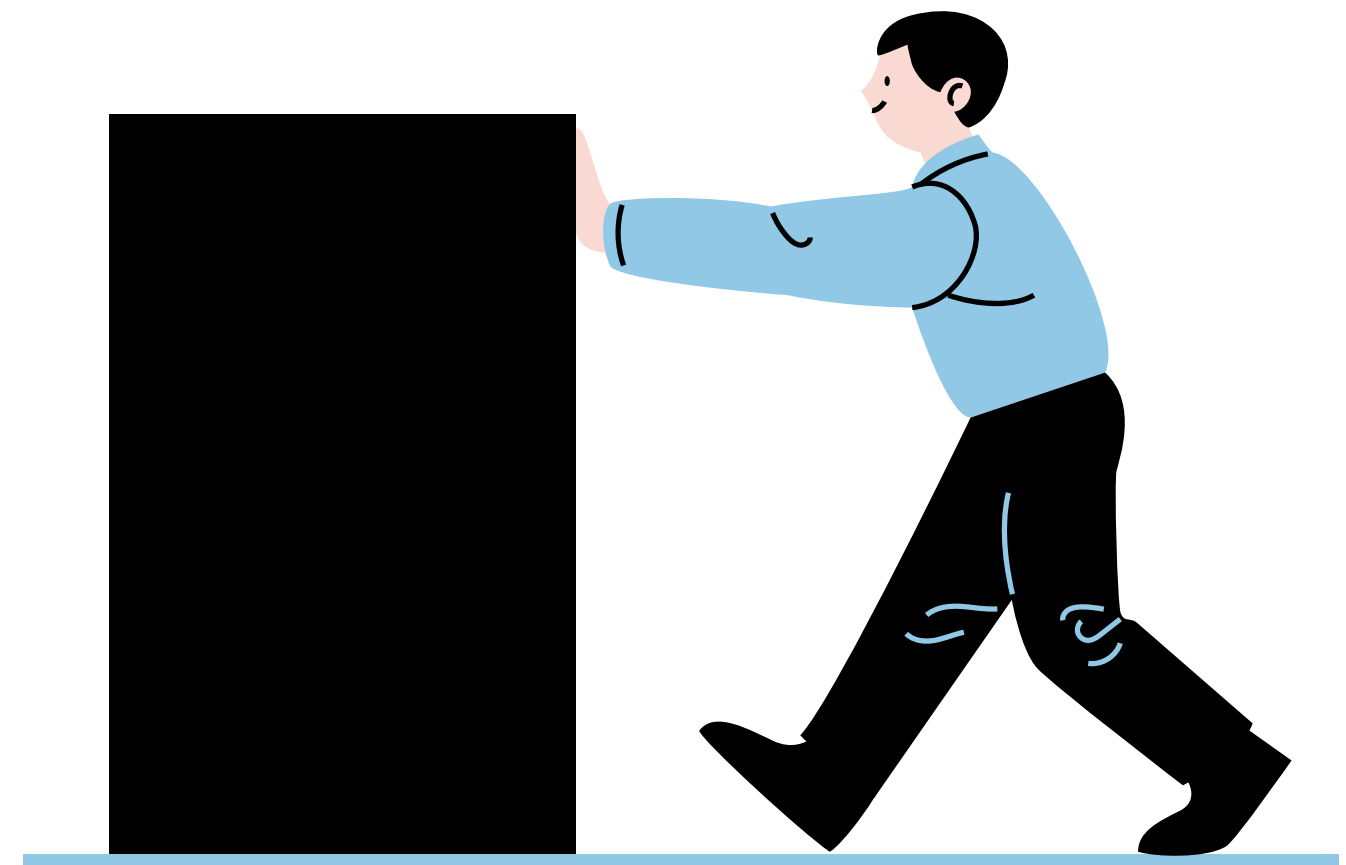
- Extends Player.
- Additional attribute: cleanSheets.
- Constructor: Initializes player's attributes along with clean sheets.
- Getter: `getCleanSheets()`.
- Overrides `getPlayerType()`: Returns "Goalkeeper".



Classes

2.DEFENDER (DERIVED CLASS)

- Extends Player.
- Additional attribute: tackles.
- Constructor: Initializes player's attributes along with tackles.
- Getter: `getTackles()`.
- Overrides `getPlayerType()`:
Returns "Defender".





3.MIDFIELDER (DERIVED CLASS)

- Extends Player.
- Additional attribute: assists.
- Constructor: Initializes player's attributes along with assists.
- Getter: `getAssists()`.
- Overrides `getPlayerType()`: Returns "Midfielder".

4.FORWARD (DERIVED CLASS)

- Extends Player.
- Additional attribute: goals.
- Constructor: Initializes player's attributes along with goals.
- Getter: `getGoals()`.
- Overrides `getPlayerType()`: Returns "Forward".

Other Classes

6.INVALIDPLAYERTYPEEXCEPTION

- **CUSTOM EXCEPTION FOR HANDLING INVALID PLAYER TYPE ERRORS.**

7.INVALIDINPUTEXCEPTION

- **CUSTOM EXCEPTION FOR HANDLING INVALID INPUT ERRORS.**

8.COACH

- **REPRESENTS A COACH WITH ATTRIBUTES LIKE :NAME, NATIONALITY, AND AGE.**

9.STADIUM

- **REPRESENTS A STADIUM WITH ATTRIBUTES LIKE NAME, LOCATION, AND CAPACITY.**

10.TEAM

- **REPRESENTS A TEAM WITH ATTRIBUTES LIKE NAME, OWNER, VALUE, COACH, STADIUM, CUPS, AND A LIST OF PLAYERS.**

Main Class for Execution:



11.MAIN

- **THE ENTRY POINT OF THE PROGRAM.**
- **MANAGES USER INPUT AND INTERACTION, ALLOWING THE USER TO ADD PLAYERS, COACH, AND STADIUM TO THE TEAM, SET THE NUMBER OF CUPS, AND VIEW TEAM DETAILS.**
- **THIS STRUCTURE ENSURES THAT COMMON ATTRIBUTES AND BEHAVIORS ARE DEFINED IN THE PLAYER CLASS, WHILE SPECIFIC DETAILS AND BEHAVIORS ARE IMPLEMENTED IN THE RESPECTIVE DERIVED CLASSES (GOALKEEPER, DEFENDER, MIDFIELDER, FORWARD).**

Program Flow

1. INITIALIZATION

- THE PROGRAM STARTS BY GREETING THE USER AND PROMPTING FOR TEAM DETAILS.

2. MAIN MENU

- OPTIONS PROVIDED:
 1. ADD PLAYER
 2. ADD COACH
 3. ADD STADIUM
 4. ADD CUPS
 5. VIEW TEAM DETAILS
 6. EXIT
- 3. ADD PLAYER
 - PROMPTS FOR PLAYER DETAILS INCLUDING TYPE.



Program Flow

4.ADD COACH

- PROMPTS FOR COACH DETAILS AND SETS THE COACH FOR THE TEAM.

5.ADD STADIUM

- PROMPTS FOR STADIUM DETAILS AND SETS THE STADIUM FOR THE TEAM.

6.ADD CUPS

- PROMPTS FOR THE NUMBER OF CUPS WON AND UPDATES THE TEAM'S RECORD.

7.VIEW TEAM DETAILS

- DISPLAYS ALL TEAM DETAILS INCLUDING PLAYERS, COACH, STADIUM, AND NUMBER OF CUPS.

8.EXIT

- EXITS THE PROGRAM.



CONCLUSION

This Football Team Management System provides a comprehensive tool for managing a football team's key components effectively. Through its structured approach and user-friendly interface, it simplifies team management tasks, ensuring all details are easily accessible and updatable.