

The Egyptian League



Java Project

|  |  |
| --- | --- |
| Ahmed Ibrahem Al-Mahallawy  Ahmed Fathy Ali Mosa  Osama Mohamed Wageh  Ammar mohammed Al-Sayed  Ahmed Al-Shawadfey sa`d | **No3**  **No10**  **No**  **No**  **No** |

**About Project**

We have package called *“The Egyptian League”.*

It was created to mange a two big clubs in Egypt , Zamalek & Ahly . We can mange the information of Players , Employee , Buildings & Stadiums.

For example we use the next options :-

* ***Add*** *,* ***show & calculate*** *methods to set & print all these information .*
* ***Class*** *options and all* ***method*** *options.*
* ***Inheritance*** *and* ***Polymorphism*** *options.*
* ***Private*** *,* ***Super*** *and* ***this****definitions.*
* ***toString() , @override*** *and* ***tryCatch*** *definitions.*
* *Arrays , loops ,* ***void*** *&* ***returned*** *methods.*
* *Etc…*

Finally , You can use the program as a small data base to store & mange your clubs information.

**Class Club**

public class Club {

//creating a club's fields

private String name;

private Player[] players;

private Building[] buildings;

private Employee[] employees;

private Stadium[] stadiums;

private int playersNo;

private int buildingsNo;

private int employeesNo;

private int stadiumsNo;

// creating a club constructor having only The club name as a parameter

public Club(String name) {

this.name = name;

playersNo = 0;

buildingsNo = 0;

employeesNo = 0;

stadiumsNo = 0;

//initialize the arrays

players = new Player[30];

buildings = new Building[10];

employees = new Employee[500];

stadiums = new Stadium[10];

}

// a method to add player to the club

public void addPlayer(Player player){

players[playersNo] = player;

playersNo++;

}

// method to add employee to the club

public void addEmployee(Employee employee){

employees[employeesNo] = employee;

employeesNo++;

}

//method to add building to the club's array of buildings

public void addBuilding(Building building){

buildings[buildingsNo] = building;

buildingsNo++;

}

// method to add stadium to the club

public void addStadium(Stadium stadium){

stadiums[stadiumsNo] = stadium;

stadiumsNo++;

}

public String getName() {

return name;

}

public int getPlayersNo() {

return playersNo;

}

public int getBuildingsNo() {

return buildingsNo;

}

public int getEmployeesNo() {

return employeesNo;

}

public int getStadiumsNo() {

return stadiumsNo;

}

public void showPlayers(){

try {

System.out.println("-------------------");

for (int i = 0; i < this.players.length; i++) {

System.out.println(this.players[i].toString());

}

}

catch (Exception e){

}

}

public void showEmployees(){

try {

System.out.println("-------------------");

for (int i = 0; i < this.employees.length; i++) {

System.out.println(this.employees[i].toString());

}

}

catch (Exception e){

}

}

public void showbulidings(){

try {

System.out.println("-------------------");

for (int i = 0; i < this.buildings.length; i++) {

System.out.println(this.buildings[i].toString());

}

}

catch (Exception e){

}

}

public void showStadiums(){

try {

System.out.println("-------------------");

for (int i = 0; i < this.stadiums.length; i++) {

System.out.println(this.stadiums[i].toString());

}

}

catch (Exception e){

}

}

@Override

public String toString() {

String str = "Welcome to "+this.name+" Club!\n this club has : "+playersNo+

" Players \n"+employeesNo+" employees\n"

+buildingsNo+" Building\n"

+stadiumsNo+" Stadium";

return str;

}

}

**Class Employee**

public class Employee {

private String name;

private int salary;

private String position;

public Employee(String name, int salary, String position) {

this.name = name;

this.salary = salary;

this.position = position;

}

public String getName() {

return name;

}

public int getSalary() {

return salary;

}

public String getPosition() {

return position;

}

@Override

public String toString() {

return "Employee{" +

"name='" + name + '\'' +

", salary=" + salary +

", position='" + position + '\'' +

'}';

}

}

**Class Accountant extends Employee**

public class Accountant extends Employee {

private boolean doubleShift;

public Accountant(String name, int salary, String position, boolean doubleShift) {

super(name, salary, position);

this.doubleShift = doubleShift;

}

public boolean isDoubleShift() {

return doubleShift;

}

@Override

public String toString() {

return "Employee{" +

"name='" + this.getName()+ '\'' +

", salary=" + this.getSalary() +

", position='" + this.getPosition() + '\'' +

" double shift "+this.doubleShift;

}

}

**Class Security extends Employee**

public class Security extends Employee {

public Security(String name, int salary) {

super(name, salary, "Security Guard");

}

}

**Class Player**

public class Player extends Employee {

private int power;

private String nationality;

public Player(String name, int power, int salary,String position, String nationality) {

super(name,salary,position);

this.nationality=nationality;

this.power = power;

}

public int getPower() {

return power;

}

public String getNationality(){return nationality;}

@Override

public String toString() {

String str = this.getName()+": - Salary = "+this.getSalary()+" Power = "+this.getPower()+" Nationality : "+this.getNationality()+" position: "+this.getPosition();

return str;

}

}

**Class Stadium**

public class Stadium {

private int capacity;

private String name;

public Stadium(String name,int capacity) {

this.capacity = capacity;

this.name = name;

}

public int getCapacity() {

return capacity;

}

@Override

public String toString() {

return "Stadium{" +

"capacity=" + capacity +

", name='" + name + '\'' +

'}';

}

public String getName() {

return name;

}

}

**Class Building**

public class Building {

private String name;

private int capacity;

private int noOfFloors;

public Building(String name, int capacity, int noOfFloors) {

this.name = name;

this.capacity = capacity;

this.noOfFloors = noOfFloors;

}

@Override

public String toString() {

return "Building{" +

"name='" + name + '\'' +

", capacity=" + capacity +

", noOfFloors=" + noOfFloors +

'}';

}

public String getName() {

return name;

}

public int getCapacity() {

return capacity;

}

public int getNoOfFloors() {

return noOfFloors;

}

}

**Class Main**

import java.util.Scanner;

public class Main {

static Scanner in;

public static void main(String[] args) {

Club Zamalek = new Club("Zamalek");

Club Ahly = new Club("Ahly");

System.out.println("Welcome to the egyptian league");

while (true){

System.out.println("-------------------");

System.out.println("choose your team");

System.out.println("1-Zamalek\n2-Ahly");

System.out.println("-------------------");

in = new Scanner(System.in);

Byte option =in.nextByte();

if(option==1){

System.out.println("choose your operation");

System.out.println("1-add player\n2-add accountant\n3-add security\n4-add building\n5-add stadium\n6-About club\n7-Exit");

System.out.println("-------------------");

byte operation = in.nextByte();

switch (operation){

case 1:

System.out.print("Insert player name : ");

String pname = in.next();

System.out.print("Insert player nationality : ");

String pNationality = in.next();

System.out.print("Insert player power :");

int ppower = in.nextInt();

System.out.print("Insert player salary :");

int psalary = in.nextInt();

System.out.print("Insert player position :");

String pposition = in.next();

Zamalek.addPlayer(new Player(pname , ppower ,psalary,pposition,pNationality));

Zamalek.showPlayers();

break;

case 2 :

System.out.print("Insert accountant name : ");

String aname = in.next();

System.out.print("Insert accountant salary :");

int asalary = in.nextInt();

System.out.print("Insert accountant position :");

String aposition = in.next();

System.out.print("Double-Shift :");

boolean adoubleShift = in.nextBoolean();

Zamalek.addEmployee(new Accountant(aname,asalary,aposition,adoubleShift));

Zamalek.showEmployees();

break;

case 3 :

System.out.print("Insert Security name : ");

String sname = in.next();

System.out.print("Insert Security salary :");

int ssalary = in.nextInt();

Zamalek.addEmployee(new Security(sname,ssalary));

Zamalek.showEmployees();

break;

case 4 :

System.out.print("Insert Building name : ");

String bname = in.next();

System.out.print("Insert Building capasity : ");

int bcapacity = in.nextInt();

System.out.print("Insert Building number of floors : ");

int bfloors = in.nextInt();

Zamalek.addBuilding(new Building(bname,bcapacity,bfloors));

break;

case 5 :

System.out.print("Insert Stadium name : ");

String stname = in.next();

System.out.print("Insert stadium capasity : ");

int stcapacity = in.nextInt();

Zamalek.addStadium(new Stadium(stname,stcapacity));

break;

case 6:

System.out.println(Zamalek.toString()); break;

case 7 : System.exit(0); break;

default:

System.out.println("wrong choice");

}

}

else if (option==2){

System.out.println("choose your operation");

System.out.println("1-add player\n2-add accountant\n3-add security\n4-add building\n5-add stadium\n6-About club\n7-Exit");

System.out.println("-------------------");

byte operation = in.nextByte();

switch (operation){

case 1:

System.out.print("Insert player name : ");

String pname = in.next();

System.out.print("Insert player nationality : ");

String pNationality = in.next();

System.out.print("Insert player power :");

int ppower = in.nextInt();

System.out.print("Insert player salary :");

int psalary = in.nextInt();

System.out.print("Insert player position :");

String pposition = in.next();

Ahly.addPlayer(new Player(pname , ppower ,psalary,pposition,pNationality));

Ahly.showPlayers();

break;

case 2 :

System.out.print("Insert accountant name : ");

String aname = in.next();

System.out.print("Insert accountant salary :");

int asalary = in.nextInt();

System.out.print("Insert accountant position :");

String aposition = in.next();

System.out.print("Double-Shift :");

boolean adoubleShift = in.nextBoolean();

Ahly.addEmployee(new Accountant(aname,asalary,aposition,adoubleShift));

Ahly.showEmployees();

break;

case 3 :

System.out.print("Insert Security name : ");

String sname = in.next();

System.out.print("Insert Security salary :");

int ssalary = in.nextInt();

Ahly.addEmployee(new Security(sname,ssalary));

Ahly.showEmployees();

break;

case 4 :

System.out.print("Insert Building name : ");

String bname = in.next();

System.out.print("Insert Building capasity : ");

int bcapacity = in.nextInt();

System.out.print("Insert Building number of floors : ");

int bfloors = in.nextInt();

Ahly.addBuilding(new Building(bname,bcapacity,bfloors));

break;

case 5 :

System.out.print("Insert Stadium name : ");

String stname = in.next();

System.out.print("Insert stadium capasity : ");

int stcapacity = in.nextInt();

Ahly.addStadium(new Stadium(stname,stcapacity));

break;

case 6:

System.out.println(Ahly.toString()); break;

case 7 : System.exit(0); break;

default:

System.out.println("wrong choice");

}

}

else{

System.out.println("wrong option");

}

}

}

}