Parte 1- Classe Gerenciar Jogador (Main)

package com.company;  
  
import java.util.Scanner;  
  
class GerenciarJogador {  
  
 public int i = 0;  
 public int j = 0;  
 com.company.Time times[] = new com.company.Time[4];  
 com.company.Jogador jogs[] = new com.company.Jogador[12];  
 Scanner sc = new Scanner(System.*in*);  
  
  
  
 public static void main(String[] args) {  
 GerenciarJogador ger = new GerenciarJogador();  
 Scanner sc = new Scanner(System.*in*);  
 int opt = 0;  
 do{  
 System.*out*.println("--- Menu Principal ---");  
 System.*out*.println();  
 System.*out*.println("1.. Cadastrar time");  
 System.*out*.println("2.. Cadastrar jogador");  
 System.*out*.println("3.. Cadastrar jogador no time");  
 System.*out*.println("4.. Remover jogador do time");  
 System.*out*.println("5.. Listar jogadores do time");  
 System.*out*.println("6.. Listar dados do time");  
 System.*out*.println("7.. Listar times");  
 System.*out*.println("9.. Sair");  
 System.*out*.println();  
 System.*out*.println("Opção: ");  
 opt = ger.validaInt("Opção");  
 switch (opt){  
 case 1:  
 ger.execCadTime();  
 break;  
 case 2:  
 ger.execCadJogador();  
 break;  
 case 3:  
 ger.execCadJogadorNoTime();  
 break;  
 case 4:  
 ger.execRemJogadorDoTime();  
 break;  
 case 5:  
 ger.listarJogadoresDoTime();  
 break;  
 case 6:  
 ger.listarDadosTime();  
 break;  
 case 7:  
 ger.listarTimes();  
 break;  
 case 9:  
 System.*out*.println("Fechando programa..");  
 break;  
 default:  
 System.*out*.println("Opção Inválida!");  
 break;  
 }  
 }while (opt!=9);  
  
 }  
  
  
  
 public void execCadTime(){  
 if (i < 4) {  
 times[i] = new com.company.Time();  
 times[i].setIdTime(i);  
 System.*out*.println("Nome do time: ");  
 times[i].setNomeTime(sc.nextLine());  
 System.*out*.println("Time " + times[i].getNomeTime() + " cadastrado com sucesso!");  
 System.*out*.println();  
 i++;  
 } else {  
 System.*out*.println("Limite de times atingido!");  
 }  
 }  
  
 */\*\*  
 \* Metodo para cadastrar um jogador criando um objeto Jogador.  
 \*/* public void execCadJogador(){  
 if (j < 12) {  
 jogs[j] = new com.company.Jogador();  
 jogs[j].setIdJogador(j);  
 System.*out*.println("Nome do jogador: ");  
 jogs[j].setNomeJogador(sc.nextLine());  
 System.*out*.println("Posição do jogador: ");  
 jogs[j].setPosicaoJogador(sc.nextLine());  
 System.*out*.println("Jogador " + jogs[j].getNomeJogador() + " cadastrado com sucesso!");  
 System.*out*.println();  
 j++;  
 } else {  
 System.*out*.println("Limite de jogadores atingido!");  
 }  
 }  
  
  
  
 public void execCadJogadorNoTime(){  
 if(seTiverTime()) {  
 System.*out*.println("JOGADORES SEM TIMES");  
 System.*out*.println();  
 for (com.company.Jogador jog : jogs) {  
 if (jog != null && !jog.isTaNumTime())  
 System.*out*.println(jog.getIdJogador() + ".. " + jog.getNomeJogador() + ".. " + jog.getPosicaoJogador());  
 }  
 System.*out*.println();  
 System.*out*.println("Id do jogador:");  
 int idJog = validaInt("Id do jogador");  
 if (idJog < j && idJog >= 0) {  
 System.*out*.println("TIMES");  
 System.*out*.println();  
 for (com.company.Time t : times) {  
 if (t != null) {  
 if (t != null) t.listar();  
 }  
 }  
 System.*out*.println();  
 System.*out*.println("Id do time: ");  
 int idTime = validaInt("Id do time");  
 if (idTime < i && idTime >= 0) {  
 times[idTime].adicionarJogador(jogs[idJog]);  
 } else {  
 System.*out*.println("Id de time não existente!");  
 }  
 } else {  
 System.*out*.println("Id de jogador não existente!");  
 }  
 }  
 }  
  
  
 public void execRemJogadorDoTime(){  
 if(seTiverTime()) {  
 System.*out*.println("TIMES");  
 System.*out*.println();  
 for (com.company.Time t : times) {  
 if (t != null) if (t != null) t.listar();  
 }  
 System.*out*.println();  
 System.*out*.println("Id do time:");  
 int idTime = validaInt("Id do time");  
 if (idTime < i && idTime >= 0) {  
 times[idTime].removerJogador();  
 } else {  
 System.*out*.println("Id de time não existente!");  
 }  
 }  
 }  
  
  
  
 public void listarDadosTime(){  
 if(seTiverTime()) {  
 System.*out*.println("ID do time:");  
 int idDesejado = validaInt("Id do time");  
 for (com.company.Time t : times) {  
 if (t != null) {  
 if (t.getIdTime() == idDesejado) {  
 t.listar();  
 return;  
 }  
 }  
 }  
 System.*out*.println("Id de time não existente!");  
 }  
 }  
  
  
  
 public void listarJogadoresDoTime(){  
 if(seTiverTime()) {  
 System.*out*.println("ID do time:");  
 int idDesejado = validaInt("Id do time");  
 for (com.company.Time t : times) {  
 if (t != null) {  
 if (t.getIdTime() == idDesejado) {  
 t.listarJogadores();  
 return;  
 }  
 }  
 }  
 System.*out*.println("Id de time não existente!");  
 }  
 }  
  
  
  
 public void listarTimes(){  
 if(seTiverTime()){  
 for (com.company.Time t : times) {  
 if (t != null) {  
 System.*out*.println();  
 System.*out*.println("\_\_\_\_\_\_\_\_ TIME\_\_\_\_\_\_\_\_\_\_\_");  
 t.listar();  
 System.*out*.println();  
 if(t.getQtdJogadores()>0) {  
 System.*out*.println("\_\_\_\_\_\_ JOGADORES\_\_\_\_\_\_\_\_");  
 t.listarJogadores();  
 System.*out*.println();  
 }  
 }  
 }  
 }  
 }  
  
  
  
 public boolean seTiverTime(){  
 if(i>0) {  
 return true;  
 }else{  
 System.*out*.println();  
 System.*out*.println("Não há times cadastrados ainda, por favor, cadastre um time.");  
 System.*out*.println();  
 return false;  
 }  
 }  
  
  
 public int validaInt(String campo){  
 int opt;  
 while(true) {  
 try {  
 opt = Integer.*parseInt*(sc.nextLine());  
 break;  
 }catch (java.lang.NumberFormatException e){  
 System.*out*.println(campo + " deve ser inteiro!");  
 System.*out*.println("Digite novamente: ");  
 }  
 }  
 return opt;  
 }  
}

Parte 2- Classe Jogador

package com.company;  
  
public class Jogador {  
 private int idJogador;  
 private String nomeJogador;  
 private String posicaoJogador;  
 private boolean taNumTime;  
  
  
 public void listarJogador(){  
 System.*out*.println("ID Jogador: " + idJogador);  
 System.*out*.println("Jogador: " + nomeJogador);  
 System.*out*.println("Posicao: " + posicaoJogador);  
 System.*out*.println("\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_");  
 System.*out*.println();  
 }  
  
  
 public int getIdJogador() {  
 return idJogador;  
 }  
  
  
 public void setIdJogador(int idJogador) {  
 this.idJogador = idJogador;  
 }  
  
 public String getNomeJogador() {  
 return nomeJogador;  
 }  
  
  
  
 public void setNomeJogador(String nomeJogador) {  
 this.nomeJogador = nomeJogador;  
 }  
  
  
  
 public String getPosicaoJogador() {  
 return posicaoJogador;  
 }  
  
  
  
 public void setPosicaoJogador(String posicaoJogador) {  
 this.posicaoJogador = posicaoJogador;  
 }  
  
  
  
 public boolean isTaNumTime() {  
 return taNumTime;  
 }  
  
  
  
 public void setTaNumTime(boolean taNumTime) {  
 this.taNumTime = taNumTime;  
 }  
}

Parte 3- Classe Time

package com.company;  
  
import java.util.Scanner;  
  
  
public class Time {  
 private int idTime;  
 private String nomeTime;  
 private com.company.Jogador[] jogadores = new com.company.Jogador[3];  
 private int qtdJogadores = 0;  
 GerenciarJogador ger = new GerenciarJogador();  
  
  
  
 public void adicionarJogador(com.company.Jogador jog){  
 if(qtdJogadores<3){  
 if(!jog.isTaNumTime()){  
 jogadores[qtdJogadores] = jog;  
 jogadores[qtdJogadores].setTaNumTime(true);  
 System.*out*.println("Jogador "+jog.getNomeJogador()+" agora está no time "+getNomeTime()+"!");  
 System.*out*.println();  
 qtdJogadores++;  
 ger.j++;  
 }else{  
 System.*out*.println("Jogador já cadastrado em um time!");  
 }  
 }else{  
 System.*out*.println("Limite de jogadores no time atingido!");  
 }  
 }  
  
  
 public void removerJogador(){  
 Scanner sc = new Scanner(System.*in*);  
 if(getQtdJogadores()>0){  
 System.*out*.println("JOGADORES DO TIME");  
 System.*out*.println();  
 for (com.company.Jogador jog : jogadores) {  
 if (jog != null)  
 System.*out*.println(jog.getIdJogador() + ".. " + jog.getNomeJogador() + ".. " + jog.getPosicaoJogador());  
 }  
 System.*out*.println();  
 System.*out*.println("Id do jogador:");  
 int idJog = ger.validaInt("Id do jogador");  
 int cont=0;  
  
 for (com.company.Jogador jog:jogadores) {  
 if(jog!=null){  
 if(jog.getIdJogador()==idJog){  
 jogadores[idJog] = null;  
 jog.setTaNumTime(false);  
 System.*out*.println();  
 System.*out*.println("Jogador removido do time");  
 System.*out*.println();  
 qtdJogadores--;  
 ger.j--;  
 break;  
 }else{  
 cont++;  
 }  
 }else {  
 cont++;  
 }  
 }  
 if(cont==3){  
 System.*out*.println();  
 System.*out*.println("Não há jogador com esse id no time!");  
 System.*out*.println();  
 }  
 }else{  
 System.*out*.println();  
 System.*out*.println("Não há jogadores no time!");  
 System.*out*.println();  
 }  
 }  
  
  
  
 public void listarJogadores(){  
 for (com.company.Jogador j: jogadores) {  
 if(j!=null) j.listarJogador();  
 }  
 }  
  
  
 public void listar(){  
 System.*out*.println(getIdTime()+".. "+getNomeTime()+".. "+getQtdJogadores()+" jogadores");  
 }  
  
  
  
 public int getIdTime() {  
 return idTime;  
 }  
  
  
  
 public void setIdTime(int idTime) {  
 this.idTime = idTime;  
 }  
  
  
  
  
 public String getNomeTime() {  
 return nomeTime;  
 }  
  
  
 public void setNomeTime(String nomeTime) {  
 this.nomeTime = nomeTime;  
 }  
  
  
 public int getQtdJogadores() {  
 return qtdJogadores;  
 }  
}