**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ВТ**

отчет

**по лабораторной работе №1**

**по дисциплине «Объектно-ориентированное программирование»**

Тема: Разработка классов

|  |  |
| --- | --- |
| Студент гр. 9308 | Семенов А.И. |
| Преподаватель | Гречухин М. Н. |

Санкт-Петербург

2021

Содержание

[Введение 3](#_Toc66815880)

[1. Уточнение этапа разработки ПК 4](#_Toc66815881)

[2. Описание классов 5](#_Toc66815882)

[3. UML-Диаграмма 6](#_Toc66815883)

[Приложение А. Листинги классов 7](#_Toc66815884)

# Введение

Цель работы: разработать базовые классы для последующей разработки программного комплекса (ПК) на объектно-ориентированном языке Java, научиться создавать UML-диаграммы для поставленной задачи.

Описание ПК:

ПК для менеджера соревнований автогонщиков. В ПК должны храниться сведения о командах, гонщиках и трассах. Менеджер соревнований может добавлять, изменять и удалять эти сведения. Ему может потребоваться следующая информация:

• состав гонщиков команды;

• призеры гонок по каждой трассе;

• график проведения соревнований;

• количество очков, набранных гонщиками.

# 1. Уточнение этапа разработки ПК

Для поставленной задачи был выбран следующий вариант базовых классов для последующей работы, которые вытекают из самого описания ПК: гонщик, команда, трасса и соревнование.

Все 4 объекта могут существовать независимо друг от друга:

- гонщик может существовать без команды, например, в случае, если он только покинул команду и в поиске новой

- команда может распасться (все гонщики покинуть команду), быть в процессе набора нового состава

- трасса физически и очевидно существует отдельно

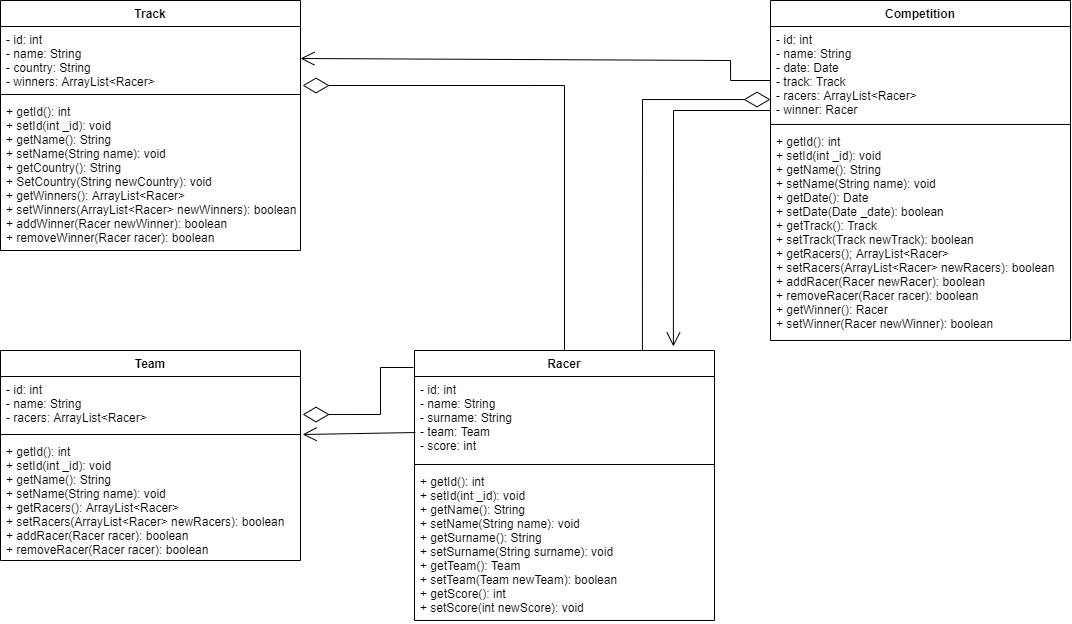
- соревнование, хоть и содержит гонщиков и трассу, но имеет право на существовании отдельно, т.к. может быть случай, когда соревнование анонсировано, но неизвестны ни трасса, ни список гонщиков.

# 2. Описание классов

* Класс гонщик (Racer) имеет следующие поля:
  + id – int – уникальный идентификатор гонщика
  + name – String – имя гонщика
  + surname – String – фамилия гонщика
  + team – Team – команда гонщика
  + score – int – кол-во набранных гонщиком очков
* Класс команда (Team) имеет следующие поля:
  + id – int – уникальный идентификатор команды
  + name – String – название команды
  + racers – ArrayList<Racer> - список гонщиков в команде
* Класс трасса (Track) имеет следующие поля:
  + id – int – уникальный идентификатор трассы
  + name – String – название команды
  + country – String – страна, где находится трасса
  + winners – ArrayList<Racer> - список победителей на трассе
* Класс соревнование (Competition) имеет следующие поля:
  + id – int – уникальный идентификатор соревнования
  + name – String – название соревнования
  + date – Date – дата проведения соревнования
  + track – Track – трасса для соревнований
  + racers – ArrayList<Racer> - список участников соревнования
  + winner – Racer – победитель соревнования

Для всех классов реализованы стандартные методы – получения и установления значения в поля. Взаимосвязь классов отражает UML-диаграмма, изображенная на рисунке 1.

# 3. UML-Диаграмма



*Рисунок 1. UML-диаграмма классов*

# Приложение А. Листинги классов

Класс Racer (гонщик)

package Races**;**

public class Racer

**{**

private int id**;**

private String name**;**

private String surname**;**

private Team team**;**

private int score **=** 0**;**

public Racer**(**int \_id**,** String \_name**,** String \_surname**,** Team \_team**)**

**{**

id **=** \_id**;**

name **=** \_name**;**

surname **=** \_surname**;**

team **=** \_team**;**

**}**

public int getId**()**

**{**

**return** id**;**

**}**

public void setId**(**int \_id**)**

**{**

id **=** \_id**;**

**}**

public String getName**()**

**{**

**return** name**;**

**}**

public void setName**(**String newName**)**

**{**

name **=** newName**;**

**}**

public String getSurname**()**

**{**

**return** surname**;**

**}**

public void setSurname**(**String newSurname**)**

**{**

surname **=** newSurname**;**

**}**

public Team getTeam**()**

**{**

**return** team**;**

**}**

public boolean setTeam**(**Team newTeam**)**

**{**

**if(**newTeam **!=** **null)**

**{**

team **=** newTeam**;**

**return** **true;**

**}**

**return** **false;**

**}**

public int getScore**()**

**{**

**return** score**;**

**}**

public void setScore**(**int newScore**)**

**{**

score **=** newScore**;**

**}**

@Override

public String toString**()**

**{**

**return** name **+** " " **+** surname **+** ", score: " **+** score**;**

**}**

**}**

Класс Team (команда)

package Races**;**

**import** java**.**util**.\*;**

public class Team

**{**

private int id**;**

private String name**;**

private ArrayList**<**Racer**>**racers**;**

public Team**(**int \_id**,** String t\_name**,** ArrayList**<**Racer**>** t\_racers**)**

**{**

id **=** \_id**;**

name **=** t\_name**;**

racers **=** t\_racers**;**

**}**

public Team**(**int \_id**,** String t\_name**)**

**{**

id **=** \_id**;**

name **=** t\_name**;**

racers **=** **new** ArrayList**<**Racer**>();**

**}**

public int getId**()**

**{**

**return** id**;**

**}**

public void setId**(**int \_id**)**

**{**

id **=** \_id**;**

**}**

public String getName**()**

**{**

**return** name**;**

**}**

public void setName**(**String newName**)**

**{**

name **=** newName**;**

**}**

public ArrayList**<**Racer**>** getRacers**()**

**{**

**return** **new** ArrayList**<**Racer**>(**racers**);**

**}**

public boolean setRacers**(**ArrayList**<**Racer**>** newRacers**)**

**{**

**if(**newRacers **!=** **null)**

**{**

racers **=** newRacers**;**

**return** **true;**

**}**

**return** **false;**

**}**

public boolean addRacer**(**Racer racer**)**

**{**

**if(**racer **!=** **null)**

**{**

racers**.**add**(**racer**);**

racer**.**setTeam**(this);**

**return** **true;**

**}**

**return** **false;**

**}**

public boolean removeRacer**(**Racer racer**)**

**{**

int index **=** racers**.**indexOf**(**racer**);**

**if(**index **!=** **-**1**)**

**{**

racers**.**remove**(**index**);**

**return** **true;**

**}**

**return** **false;**

**}**

@Override

public String toString**()**

**{**

**return** "Team: " **+** name**;**

**}**

**}**

Класс Track (трасса)

package Races**;**

**import** java**.**util**.\*;**

public class Track

**{**

private int id**;**

private String name**;**

private String country**;**

private ArrayList**<**Racer**>**winners**;**

public Track**(**int \_id**,** String t\_name**,** String t\_country**)**

**{**

id **=** \_id**;**

name **=** t\_name**;**

country **=** t\_country**;**

winners **=** **new** ArrayList**<**Racer**>();**

**}**

public int getId**()**

**{**

**return** id**;**

**}**

public void setId**(**int \_id**)**

**{**

id **=** \_id**;**

**}**

public String getName**()**

**{**

**return** name**;**

**}**

public void setName**(**String newName**)**

**{**

name **=** newName**;**

**}**

public String getCountry**()**

**{**

**return** country**;**

**}**

public void setCountry**(**String newCountry**)**

**{**

country **=** newCountry**;**

**}**

public ArrayList**<**Racer**>** getWinners**()**

**{**

**return** **new** ArrayList**<**Racer**>(**winners**);**

**}**

public boolean setWinners**(**ArrayList**<**Racer**>**newWinners**)**

**{**

**if(**newWinners **!=** **null)**

**{**

winners **=** newWinners**;**

**return** **true;**

**}**

**return** **false;**

**}**

public boolean addWinner**(**Racer newWinner**)**

**{**

**if(**newWinner **!=** **null)**

**{**

winners**.**add**(**newWinner**);**

**return** **true;**

**}**

**return** **false;**

**}**

public boolean removeWinner**(**Racer racer**)**

**{**

int index **=** winners**.**indexOf**(**racer**);**

**if(**index **!=** **-**1**)**

**{**

winners**.**remove**(**index**);**

**return** **true;**

**}**

**return** **false;**

**}**

@Override

public String toString**()**

**{**

**return** name **+** "(" **+** country **+** ")"**;**

**}**

**}**

Класс Competition (соревнование)

package Races**;**

**import** java**.**util**.\*;**

public class Competition

**{**

private int id**;**

private String name**;**

private Date date**;**

private Track track**;**

private ArrayList**<**Racer**>**racers**;**

private Racer winner**;**

public Competition**(**int \_id**,** String \_name**,** Date \_date**,** Track \_track**,** ArrayList**<**Racer**>** \_racers**)**

**{**

id **=** \_id**;**

name **=** \_name**;**

date **=** \_date**;**

track **=** \_track**;**

racers **=** \_racers**;**

**}**

public int getId**()**

**{**

**return** id**;**

**}**

public void setId**(**int \_id**)**

**{**

id **=** \_id**;**

**}**

public String getName**()**

**{**

**return** name**;**

**}**

public void setName**(**String newName**)**

**{**

name **=** newName**;**

**}**

public Date getDate**()**

**{**

**return** date**;**

**}**

public boolean setDate**(**Date \_date**)**

**{**

**if(**\_date **!=** **null)**

**{**

date **=** \_date**;**

**return** **true;**

**}**

**return** **false;**

**}**

public Track getTrack**()**

**{**

**return** track**;**

**}**

public boolean setTrack**(**Track newTrack**)**

**{**

**if(**newTrack **!=** **null)**

**{**

track **=** newTrack**;**

**return** **true;**

**}**

**return** **false;**

**}**

public ArrayList**<**Racer**>** getRacers**()**

**{**

**return** **new** ArrayList**<**Racer**>(**racers**);**

**}**

public boolean setRacers**(**ArrayList**<**Racer**>** newRacers**)**

**{**

**if(**newRacers **!=** **null)**

**{**

racers **=** newRacers**;**

**return** **true;**

**}**

**return** **false;**

**}**

public boolean addRacer**(**Racer newRacer**)**

**{**

**if(**newRacer **!=** **null)**

**{**

racers**.**add**(**newRacer**);**

**return** **true;**

**}**

**return** **false;**

**}**

public boolean removeRacer**(**Racer racer**)**

**{**

**if(**racer **!=** **null)**

**{**

racers**.**remove**(**racer**);**

**return** **true;**

**}**

**return** **false;**

**}**

public Racer getWinner**()**

**{**

**return** winner**;**

**}**

public boolean setWinner**(**Racer newWinner**)**

**{**

**if(**newWinner **!=** **null)**

**{**

winner **=** newWinner**;**

**return** **true;**

**}**

**return** **false;**

**}**

@Override

public String toString**()**

**{**

**return** "Competition " **+** name **+** ". Track: " **+** track **+** ". Date: " **+** date**;**

**}**

**}**