Driver
+PATH: String = info.in
+athleteMap: Map<String, Athlete>
+officialMap: Map<String, Official>
+gameList: List<Game>
+currentGame: Game
+predictId: String
+readData()
+showMenu(): void
+createGame(scanenr:Scanner): void
+selectPlayers(scanner:Scanner): boolean
+predictWinner(scanner:Scanner): void
+runGame(): void
+showAllResults(): void
+showAllPoints(): void

Ozlympic
+main(args:String[]): static void

Comparable
+compareTo(p:Participant): int

Athlete
+getType(): GameType
+setType(type:GameType)
+compete(ran:Random): int

Participant
+id: String
+name: String
+age: int
+state: String
+time: int
+point: int
+Participant(id:String,name:String,age:int,
             state:String)
+compareTo(p:Participant): int
+copy(): Participant
+addPoint(p:int): void

Game
+id: String
+athleteList: List<Athlete>
+official: Official
+type: GameType
+recordList: List<Participant>
+Game(id:String)
+addAthlete(ath:Athlete): void
+getAthletes(): List<Athlete>
+record(pants:List<Participant>): void
+toString(): String
+setType(type:GameType): void
+getType(): GameType

Official

Cyclist

Sprinter

Swimmer

superAthlete

Swimming
+Swimming(id:String)

Running
+Running(id:String)

Cycling
+Cycling(id:String)

1.All athletes' information is stored in the Driver class. The user predictions and selected athletes are stored in the currentGame object, which is hold by Driver. The historical games are stored in the gameList.

2.The class Participant is an abstract class so that the user can't create a generic type of Participant.

3.When running the currentGame, the runGame method will call the compete() method of all selected athletes. After that, the official will summarize the game and sort the athletes' score to give points.

4.The user's prediction is stored in the Driver class. The Official's summarize() method will return the id of the winner. Then the Driver will check if the user's predicition corresponds to the winner.