- id: int - id: int - name: String - name: String description: String description: String - riders: Map<Integer, Rider> - riders: ArrayList<Rider> - orderedRiderIds: ArrayList<Integer> - stages: Map<Integer, Stage> - orderedStageIds: ArrayList<Integer> + Team(name: String, description: String, teams: Map<Integer, Team>) adjustedTimes: Map<LocalTime, Integer> Stage + getId() : int - sprinterPoints: Map<Integer, Integer> - id: int - mountainPoints: Map<Integer, Integer> + getName() : String - name: String + getOrderedRiderIds(): ArrayList<Integer> - sprinterClassification: Map<Integer, Integer> startTime: LocalDateTime + getRiders() : Map<Integer, Rider> - mountainClassification: Map<Integer, Integer> length: double + addRider(name: String, yearOfBirth: int) : int + Race(name: String, description: String, races: Map<Integer, Race>) - race: Race + findRider(riderId: int, teams: Map<Integer, Team>): Rider + getId() : int description: String occurs in + findTeam(teamId: int, teams: Map<Integer, Team>): Team + getName() : String - type: StageType + getDescription() : String checkpoints: Map<Integer, Checkpoint> + getLength() : double - state: String is/are part of + getSprinterClassification: Map<Integer, Integer> riderTimes: Map<Integer, LocalTime[ ]> + getMountainClassification: Map<Integer, Integer> adjustedTimes: Map<Integer, LocalTime> + sprinterPointsSorted(): ArrayList<Integer> Rider - sprinterPoints: Map<Integer, Integer> + mountainPointsSorted(): ArrayList<Integer> - id: int mountainPoints: Map<Integer, Integer> + addStage(name: String, description: String, length: double, startTime: LocalDateTime, type: StageType) : int name: String riderPositions: ArrayList<Integer> + getRiders() : ArrayList<Rider> yearOfBirth: int sprintCheckpointIds: ArrayList<Integer> + getStages() : Map<Integer, Stage> team: Team take part in + findStage(stageId: int, races: Map<Integer, Race>) : Stage sprintCheckpointRef: ArrayList<Integer> racesEnrolled: ArrayList<Integer> + findStagesRace(stageId: int, races: Map<Integer, Race>) : Race mountainCheckpointIds: ArrayList<Integer> + Rider(name: String, yearOfBirth: int, id: int, team: Team) + toString(): String {overridden} mountainCheckpointRef: ArrayList<Integer> getId() : int - FLATSTAGEPOINTS: int[50,30,20,18,16,14,12,10,8,7,6,5,4,3,2] {readOnly} getRacesEnrolled() : ArrayList<Integer> + getTeam() : Team - MEDIUMSTAGEPOINTS: int[30,25,22,19,17,15,13,11,9,7,6,5,4,3,2] {readOnly} - HIGHMOUNTAINPOINTS: int[20,17,15,13,11,10,9,8,7,6,5,4,3,2,1] {readOnly} - TIMETRIALPOINTS: int[20,17,15,13,11,10,9,8,7,6,5,4,3,2,1] {readOnly} Checkpoint {Abstract} + Stage() - location: double CheckpointType + Stage(name: String, description: String, length: double, startTime: LocalDateTime, type: StageType, id: int, race: Race) occurs in type: CheckpointType SPRINT + getld() : int riderTimes: Map<LocalTime, Integer> + getName() : String - sortedTimes: ArrayList<LocalTime> + getLength() : double + Checkpoint(location: double, id: int) + getState() : String getId(): int {abstract} + setState() : void - getType() : CheckpointType {abstract} + getType() : StageType - addResult() : void {abstract} + addResults(id: int, times: LocalTime[]) : void + sortResults() : void {overridden} + getResults(id: int) : LocalTime[ ] + getRiderPointReward(riderId: int) : int {abstract} + addMountainCheckpoint(location: double, type CheckpointType, gradient: double, length: double): int - id: int + addSprintCheckpoint(location: double) : int + getMountainPoints() : Map<Integer, Integer> + getSprinterPoints() : Map<Integer, Integer> + getRiderMountainPoints(riderId: int) : int + getRiderSprinterPoints(riderId: int) : int MountainCheckpoint + adjustTimes(index: int) : Map<Integer, LocalTime> - id: int + getAdjustedTimes() : Map<Integer, LocalTime> SprintCheckpoint - location: double + getCheckpoints() : Map<Integer, Checkpoint> - id: int - gradient: double + findCheckpointsStage(checkpointld: int, races: Map<Integer, Race>, stageIds: ArrayList<Integer>): Stage - location: double - length: double + getRiderTimes() : Map<Integer, LocalTime[]> - type: CheckpointType type: CheckpointType + getRiderPositions() : ArrayList<Integer> - riderTimes: Map<LocalTime, Integer> - riderTimes: Map<LocalTime, Integer> + getRace() : Race - sortedTimes: ArrayList<LocalTime> - sortedTimes: ArrayList<LocalTime> - POINTS: int[20,17,15,13,11,10,9,8,7,6,5,4,3,2,1] {readOnly} - CATFOURPOINTS: int[1] {readOnly} - CATTHREEPOINTS: int[2, 1] {readOnly} + SprintCheckpoint(location: double, id: int) - CATTWOPOINTS: int[5, 3, 2, 1] {readOnly} + getId() : int {overridden} + getType() : CheckpointType {overridden} - CATONEPOINTS: int[10, 8, 6, 4, 2, 1] {readOnly} + addResult(riderId: int, time: LocalTime) : void {overridden} - CATHORSPOINTS: int[20,15,12,10,8,6,4,2] {readOnly} + sortResults() : void {overridden} + MountainCheckpoint(location double, type: CheckpointType, gradient: double, length: double, id: int) + getRiderPointReward(riderId: int) : int {overridden} + getId() : int {overridden} + getType() : CheckpointType {overridden} + addResult(riderId: int, time: LocalTime) : void {overridden}

CyclingPortalImpl

is/are stored in

Team

races: Map<Integer, Race>
teams: Map<Integer, Team>

is/are stored in

+ sortResults() : void {overridden}

+ getRiderPointReward(riderId: int) : int {overridden}

Race

usedStageIds: ArrayList<Integer>

usedRiderIds: ArrayList<Integer>

+ functions as given in specification

usedCheckpointlds: ArrayList<Integer>

StageType

FLAT

MEDIUM\_MOUNTAIN

HIGH\_MOUNTAIN