CyclingPortalImpl · CyclingPortalImpl(): races: ArrayList<Entity> teams: ArrayList<Entity> nextld: int - getStageCheckpoints(int): int[] getTeams(): int[] · getRankedAdjustedElapsedTimesInStage(int): LocalTime[] removeRiderResults(Rider): void getRaceStages(int): int[] - addCategorizedClimbToStage(int, Double, CheckpointType, Double, Double): int removeCheckpoint(int): void getEntity(int, ArrayList<Entity>): Optional<Entity> - deleteRiderResultsInStage(int, int): void eraseCyclingPortal(): void viewRaceDetails(int): String getRiderResultsInStage(int, int): LocalTime[] · getStageLength(int): double createRider(int, String, int): int registerRiderResultsInStage(int, int, LocalTime[]): void getRidersMountainPointsInStage(int): int[] createRace(String, String): int downcast(ArrayList<Entity>, Class<T>): ArrayList<T> removeRider(int): void toLocalTime(Duration): LocalTime removeRaceById(int): void getRidersPointsInStage(int): int[] getEntity(int, ArrayList<HasChildren>, Class<T>): T getNumberOfStages(int): int validateName(ArrayList<Entity>, String): void getTeamRiders(int): int[] loadCyclingPortal(String): void getRiderAdjustedElapsedTimeInStage(int, int): LocalTime addStageToRace(int, String, String, double, LocalDateTime, StageType): int concludeStagePreparation(int): void - addIntermediateSprintToStage(int, double): int - createTeam(String, String): int getRaceIds(): int[] getRidersRankInStage(int): int[] removeStageById(int): void removeTeam(int): void ⊦ saveCyclingPortal(String): void

