Programming II Fitness Project Class Diagram

+ << get/set>> - averageHeartRate : int

+ << get/set>> - distance : double

+ << get/set>> - reps: int

+ << get/set>> - weightLifted : double

Person (abstract class) Exercise + <<get/set>> - studentID: int + << get/set>> # studentID : int + << get/set>> - exercises : List< Exercise> + << get/set>> # exerciseDate : LocalDate 0..* 1..1 + << get/set>> - firstName : String + << get/set>> # exerciseName : String + << get/set>> - lastName : String + << get/set>> # exerciseDuration : Duration + << get/set>> - height : double + abstract void load(studentID: int, + << get/set>> - weight : double exerciseDate: LocalDate, + << get/set>> - gender : enum exerciseName: String) + << get/set>> - birthdate : LocalDate + abstract void save() + int getAge() + abstract void delete() + void addExercise(exercise : Exercise) + void removeExercise(indexPositionToRemove : int) + void load(studentID: int) + void save() + void delete() **ExerciseAerobic** ExerciseStrength + << get/set>> - maxHeartRate : int + <<get/set>> - sets : int

Programming II Fitness Project Stored Procedure Reference

Person

usp GetAllPeople() --optional to use this stored procedure if you want it.

usp_GetPerson(studentID)

usp_SavePerson(studentID, firstName, lastName, height, weight, gender, birthdate)

usp_DeletePerson(studentID)

ExerciseAerobic

usp_GetAerobicExercisesByPerson(studentID)

usp GetExerciseAerobic(studentID, exerciseDate, exerciseName)

usp_SaveExerciseAerobic(studentID, exerciseDate, exerciseName, exerciseSeconds, maxHeartRate, averageHeartRate, distance)

usp_DeleteExerciseAerobic(studentID, exerciseDate, exerciseName)

ExerciseStrength

usp_GetStrengthExercisesByPerson(studentID)

usp GetExerciseStrength(studentID, exerciseDate, exerciseName)

usp_SaveExerciseStrength(studentID, exerciseDate, exerciseName, exerciseSeconds, sets, reps, weightLifted)

 $usp_Delete Exercise Strength (student ID, exercise Date, exercise Name$