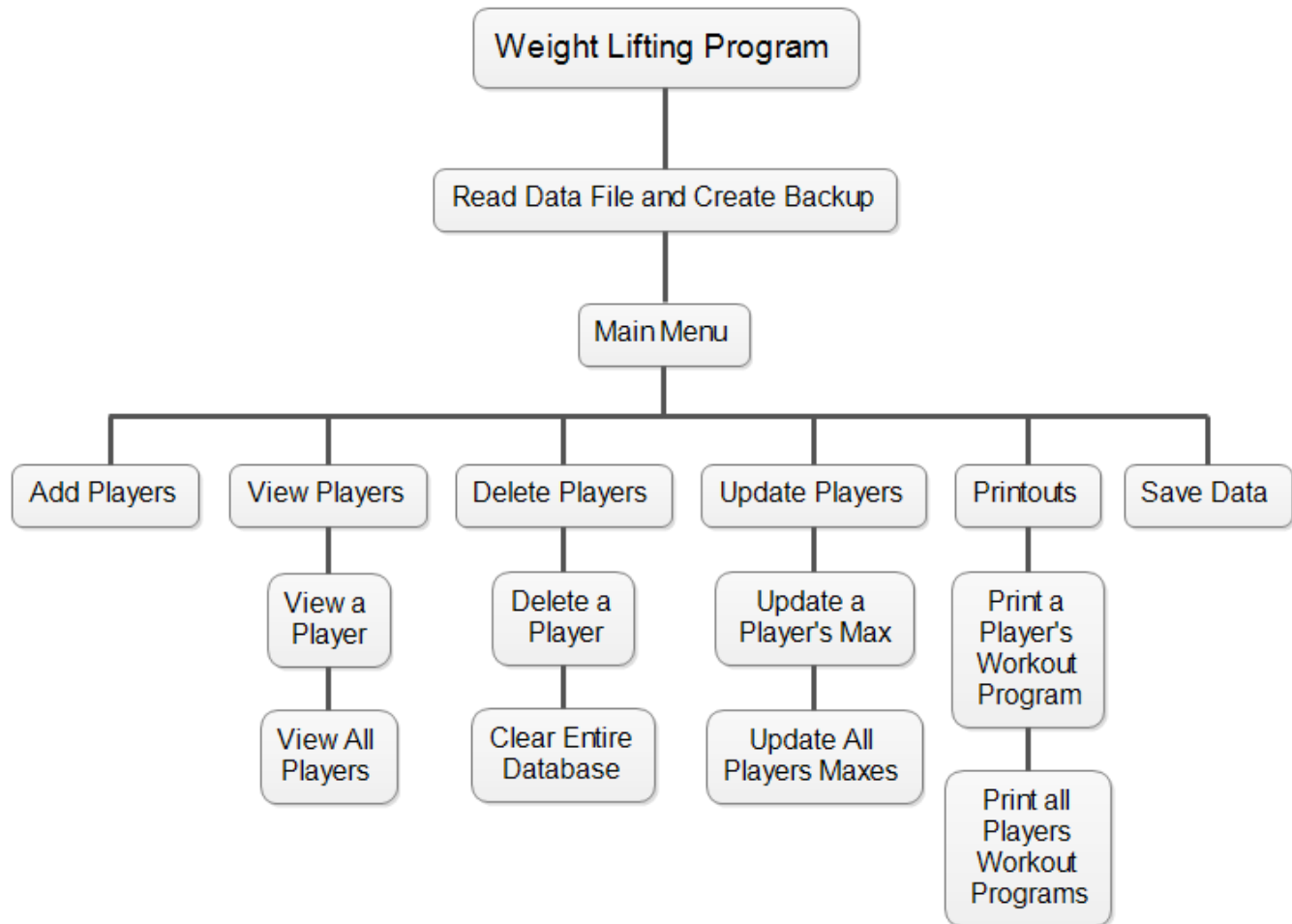


## Criterion B: Design

### Structure Chart

The first step in designing the Weight Lifting Program was to organize my thoughts using a structure chart which gave me the basic design of the program.



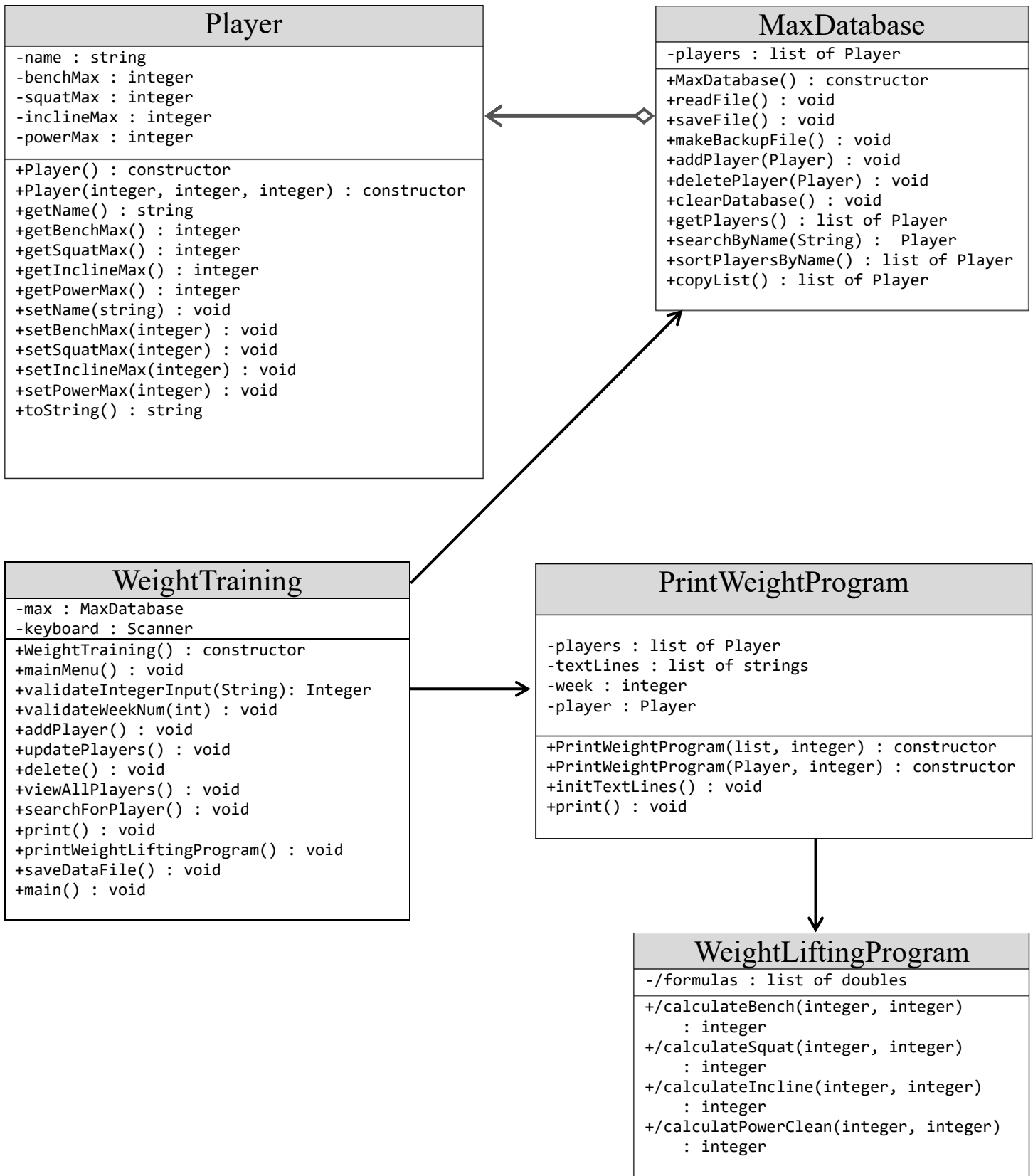
The structure chart helped me to visualize the tasks that the program needed to perform. As a result I decided to organize these tasks into the following Java classes: Player, WeightTraining, MaxDatabase, WeightLiftingProgram, and PrintWeightProgram.

Each class has the following responsibilities:

- **Player** – stores each player's name and weight max for each exercise.
- **WeightTraining** – this class is the program's user interface. It is responsible for interacting with the user through a menu system. It contains the program's main method.
- **MaxDatabase** – maintains a list of Player's. It also provides methods for manipulating this list.
- **WeightLiftingProgram** - provides static methods for calculating weekly workout weights for each of the four main exercises: bench, squat, incline, and power clean.
- **PrintWeightProgram** – this class provides methods that communicate with a printer. It can print a weekly workout program for every player.

## UML Class Diagrams

To help me decompose the problem into classes I used UML class diagrams.



## Prototype

Next I created a prototype ([Appendix B](#)) for the program so that I could take it to my client and he could see the program's user interface. The prototype provided my client the opportunity to give the ok or suggest any changes he would like to make. For the design of the prototype I choose to create a stub program. Since the program is a terminal window program creating a stub program would have the most impact because my client could see the program in action from a visual stand point. As an added benefit the stub program provided me the opportunity to write some code that could be used as a framework for the program.

When I showed my client the prototype he was very pleased, however he did suggest a couple of changes. First he asked if I could have the program print a list of all the players divided into groups according to their max on the bench press. He said when players are organized into workout groups by weight max it is easier for the players to change out the weights when they are not lifting the same amount of weight. Second he thought it would be helpful if the program could record the players' classification (freshman, sophomore, junior, senior) so that at the end of the school year the seniors could be deleted from the database and the underclassmen promoted to the next grade level.

## Modifications to Initial Design

To accommodate my client's recommendations I made the following changes to my initial design:

1. Added a classification instance variable to the Player class.
2. Added two additional classes to the program's design named Group and PrintGroups.
  - **Group** – this class stores a specified number of Players to form a workout group.
  - **PrintGroups** – this class is responsible for interacting with the printer and printing a list of Groups

## Inputs and Output

**Inputs** - Player name, classification, bench press max, squat max, incline max, and power clean max.

### Sample Input Screen

```
=====
      Add Player
=====
Enter Player Name (lastname, firstname)-->Wayne, John
Enter Player Classification (9,10,11,12)-->11
Enter Bench Max -->300
Enter Squat Max -->450
Enter Incline Max -->270
Enter Power Clean Max -->250
```

**Outputs**

- Display – individual player record, all player names sorted by last name
- Printout – workout cards, workout groups list

**Sample Workout Card Printout**

Name: Wayne, John  
 Program Week: 1  
 Current Maxes: Bench – 300    Squat – 450    Incline – 275    Power Clean – 250  
 -----  
 Reps: 10, 8, 8  
 Squat: 270                      Power Clean: 150  
 Bench: 180                      Incline: 165

**Sample Groups List Printout**

## Group 1

-----

Douglas Kirk	Bench Max = 400
Mitchum, Robert	Bench Max = 350
Cooper, Gary	Bench Max = 325
Holden, William	Bench Max = 305

## Group 2

-----

Wayne, John	Bench Max = 300
Gable, Clark	Bench Max = 300
Peck, Gregory	Bench Max = 260
Fonda, Henry	Bench Max = 250

## Group 3

-----

Hudson, Rock	Bench Max = 240
Heston, Charlton	Bench Max = 225
Stewart, Jimmy	Bench Max = 200
Cagne, James	Bench Max = 200

## Test Plan

The chart below describes my test plan that will determine whether the program is functioning properly and meets all the criteria.

Program Feature to Test	Test Strategy
ProtoType	<ul style="list-style-type: none"> <li>Run the prototype and verify that the menu system works as intended.</li> <li>Test submenus to verify that they can be navigated easily.</li> </ul>
Data Validation System	<ul style="list-style-type: none"> <li>Verify that if user types invalid data he is able to re-enter the data instead of program crashing.               <ul style="list-style-type: none"> <li>Type a letter when program expects a number.</li> <li>Type a number that is not in the expected range.</li> </ul> </li> </ul>
Database - ArrayList	<ul style="list-style-type: none"> <li>Add 5 Player records to the database. Display the database to verify that data was stored as expected.</li> </ul>
Player Update/Deletion Features	<ul style="list-style-type: none"> <li>Update a Player record by changing his max values.</li> <li>Delete a Player from the database.</li> </ul>
File Input/Output	<ul style="list-style-type: none"> <li>Verify that program creates data file the first time program is launched.</li> <li>Save database to data file. Exit program. Execute program again to verify that data is loaded into program properly.</li> </ul>
Backup File System	<ul style="list-style-type: none"> <li>Verify that program copies data from input file into a backup file when program is executed.</li> </ul>
Sorting Feature	<ul style="list-style-type: none"> <li>Verify that sorting database by name works by displaying list.</li> <li>Verify that sorting by bench max works by displaying list.</li> </ul>
Workout Program Formulas	<ul style="list-style-type: none"> <li>Verify that formulas that calculate the workout programs for players is correct by doing some samples manually on a calculator.</li> </ul>
Workout Program Print Feature	<ul style="list-style-type: none"> <li>Print a week 1 workout program for all players.</li> <li>Print a week 10 workout program for a single player.</li> <li>Add 10 additional records to database to verify that multipage printouts work as expected (margins, spacing).</li> </ul>
Workout Groups Feature	<ul style="list-style-type: none"> <li>Print workout groups to verify that players are organized into groups of 4 according to their bench max.</li> </ul>
Closeout School Year Feature	<ul style="list-style-type: none"> <li>Add Player records to database that a varying classifications. Run this feature and verify that seniors are removed from the database and underclassmen are promoted to the next grade level.</li> </ul>

Words: 473