```java
package org.bwagner;

import java.util.*;

/*
   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.
*/

public class WeightTraining
{
    //instance variables
    private Scanner keyboard;

    // constructor
    public WeightTraining()
    {
        keyboard = new Scanner(System.in);

        mainMenu();
    }

    /*
       This is the main menu for the program. All interaction with the user
       originates from this menu.
    */
    public void mainMenu()
    {
        int ans = 0;

        do
        {
            System.out.println();
            System.out.println("=====================");
            System.out.println("       Main Menu   ");
            System.out.println("=====================");
            System.out.println("  1. Add Player");
            System.out.println("  2. Update Player Maxes");
            System.out.println("  3. View List of Player Names");
            System.out.println("  4. View a Player's Maxes");
            System.out.println("  5. Delete Players");
            System.out.println("  6. Print");
            System.out.println("  7. Save");
            System.out.println("  8. Exit");

            ans = validateIntegerInput("Selection -->");
            System.out.println();
            if(ans == 1)
                addPlayer();
            if(ans == 2)
                updatePlayers();
            if(ans == 3)
```

```
                viewAllPlayers();
            if(ans == 4)
                searchForPlayer();
            if(ans == 5)
                delete();
            if(ans == 6)
                print();
            if(ans == 7)
                saveDataFile();
        }
        while(ans != 8);

        System.out.println();
        System.out.println("Good Bye!");
        System.out.println();
        System.exit(0);        // close terminal window
    }

    /*
       This method allows the user to enter an integer value. It then verifies
       that the input value is an integer. If it is not an integer the method
       prompts the user to re-enter the value again.
       @return the input value
       @param prompt the input prompt
    */
    public int validateIntegerInput(String prompt)
    {
        int ans = 0;
        boolean flag;

        do
        {
            flag = true;
            System.out.print(prompt);  // display input prompt
            try
            {
                ans = keyboard.nextInt();

            }
            catch (Exception e)
            {
                System.out.println("Invalid Entry. Try again.");
                flag = false;
            }
            keyboard.nextLine();      // clear buffer
        }
        while(flag == false);

        return ans;
    }

    /*
```

```java
       This method validates that the parameter week is between
       1 <= week <= 10. If it is not it requires the user to enter
       a valid number.
       @param the week value(1-10)
    */
    public void validateWeekNum(int week)
    {
        while(week < 1 || week > 10)
        {
            week = validateIntegerInput("Enter Program Week (1, 10) -->");
        }
    }

    /*
       This method prompts the user to enter a player's info and then adds
       the player to the database.
    */
    public void addPlayer()
    {
        String ans = "";
        do
        {
            System.out.println("====================");
            System.out.println("     Add Player");
            System.out.println("====================");
            System.out.print("Enter Player Name (lastname, firstname)-->");
            String name = keyboard.nextLine();

            int bench = validateIntegerInput("Enter Bench Max -->");
            int squat = validateIntegerInput("Enter Squat Max -->");
            int incline = validateIntegerInput("Enter Incline Max -->");
            int power = validateIntegerInput("Enter Power Clean Max -->");

            System.out.println();
            System.out.print("Add another player[Y/N]?");

            ans = keyboard.nextLine();
        }
        while(ans.equalsIgnoreCase("y"));
    }

    /*
       This method allows a user to modify all players or single player
       max values.
    */

    public void updatePlayers()
    {
        System.out.println("=======================");
        System.out.println("   Update Players Max");
        System.out.println("=======================");
        System.out.println("1. Update a Player's Max");
        System.out.println("2. Update All Players' Max");
        int ans = validateIntegerInput("Selection -->");

        if(ans == 1)
        {
            String response = "";
            do
            {
                System.out.println();
                System.out.print("Enter Player Name (lastname, firstname)-->");
                String name = keyboard.nextLine();

                System.out.println();

                int bench = validateIntegerInput("Enter new Bench Max -->");
                int squat = validateIntegerInput("Enter new Squat Max -->");
                int incline = validateIntegerInput("Enter new Incline Max -->");
                int power = validateIntegerInput("Enter new Power Clean Max -->");

                System.out.println();
                System.out.print("Update Another Player[Y/N]-->");
                response = keyboard.nextLine();

            }
            while(response.equalsIgnoreCase("y"));
        }
        if(ans == 2)
        {
            System.out.println("=======================");
            System.out.println("   Update Players Max");
            System.out.println("=======================");
        }
    }

    /* This method allows the user to remove a player from the database or
       clear the database of all players.
    */
    public void delete()
    {
        System.out.println("====================");
        System.out.println("   Delete Player");
        System.out.println("====================");
        System.out.println("  1. Delete a Player");
        System.out.println("  2. Clear Database");
        int ans = validateIntegerInput("Selection -->");
    }

    /*
       This method displays a list in alphabetical of all players in the
       database. It displays each player's name and classification.
    */
    public void viewAllPlayers()
    {
        System.out.println("=====================");
        System.out.println("   View All Players");
        System.out.println("=====================");
    }

    /* This method searches the database by player name. If the player is found
       it displays the Player's exercise maxes.
    */
    public void searchForPlayer()
    {
        System.out.println("=====================");
        System.out.println("  Search For Player");
        System.out.println("=====================");
        System.out.print("Enter Player Name (lastname, firstname)-->");
        String name = keyboard.nextLine();
    }

    /* This method allows the user to print two documents.
       1. A player or players workout program.
       2. A list of players organized in groups of four by
          their bench max.
    */
    public void print()
    {
        System.out.println("=====================");
        System.out.println("       Print");
        System.out.println("=====================");
    }

    /*
       This method saves the databasse to the data file.
    */
    public void saveDataFile()
    {
        System.out.println("=====================");
        System.out.println("   Save Data File");
        System.out.println("=====================");
    }

    /*
        This is the program's main menu.
    */
    public static void main(String[] args)
    {
        WeightTraining app = new WeightTraining();
    }
}
```