Strength Manager

프로그래밍 기획서

JAVA WEB개발 김회준

목차

1. 프로그램 개요

2. 프로젝트 환경(언어)

3. 참조 레퍼런스 문헌

4. 기간 별 수행 구상

5. 기능 설계 MVC & CSS

1. 프로그램 개요

헬스 트레이닝을 막 입문하게 된다면 3대 운동에 대해 알게 될 것이다.

3대 운동은 운동의 시작과 끝이라고 할 수 있을 만큼

다른 다양한 운동의 수행능력에 절대적으로 관여한다.

헬스의 초심자 레벨을 넘어선다면 점차 큰 근육들을 사용하는

3대 운동에 대한 중요성을 자연스레 깨달아 갈 것인데

막 **초심자**를 벗어난 사람들은 **3대 운동을 어떻게 측정하는지**,

어떻게 훈련하는지에 대한 지식이 없다.

1. 프로그램 개요

본 프로그램은

이러한 헬스 초심자를 막 벗어난 사람들을 위해

3대 운동에 대한 Wilks Point를 측정해주고

531 Strength트레이닝 루틴 혹은 5x5루틴을 계획해주고

시간에 따라 Wilks Point 를 측정해

변화된 자신의 **Strength를 그래프로 한눈에 보기 쉽게** 해줄 것이다.

이 프로그램의 궁극적 목표는

보다 효율적인 운동이 가능하게끔 도움을 주는 것이다.

* Wilks Point : 3대 운동에 따른 운동 수행 지표
* 531 루틴 & 5X5 루틴 : Strength 훈련 루틴

2. 프로젝트 환경(언어)

JAVA, CSS

3. 참조 레퍼런스 문헌

CSS(김수철)

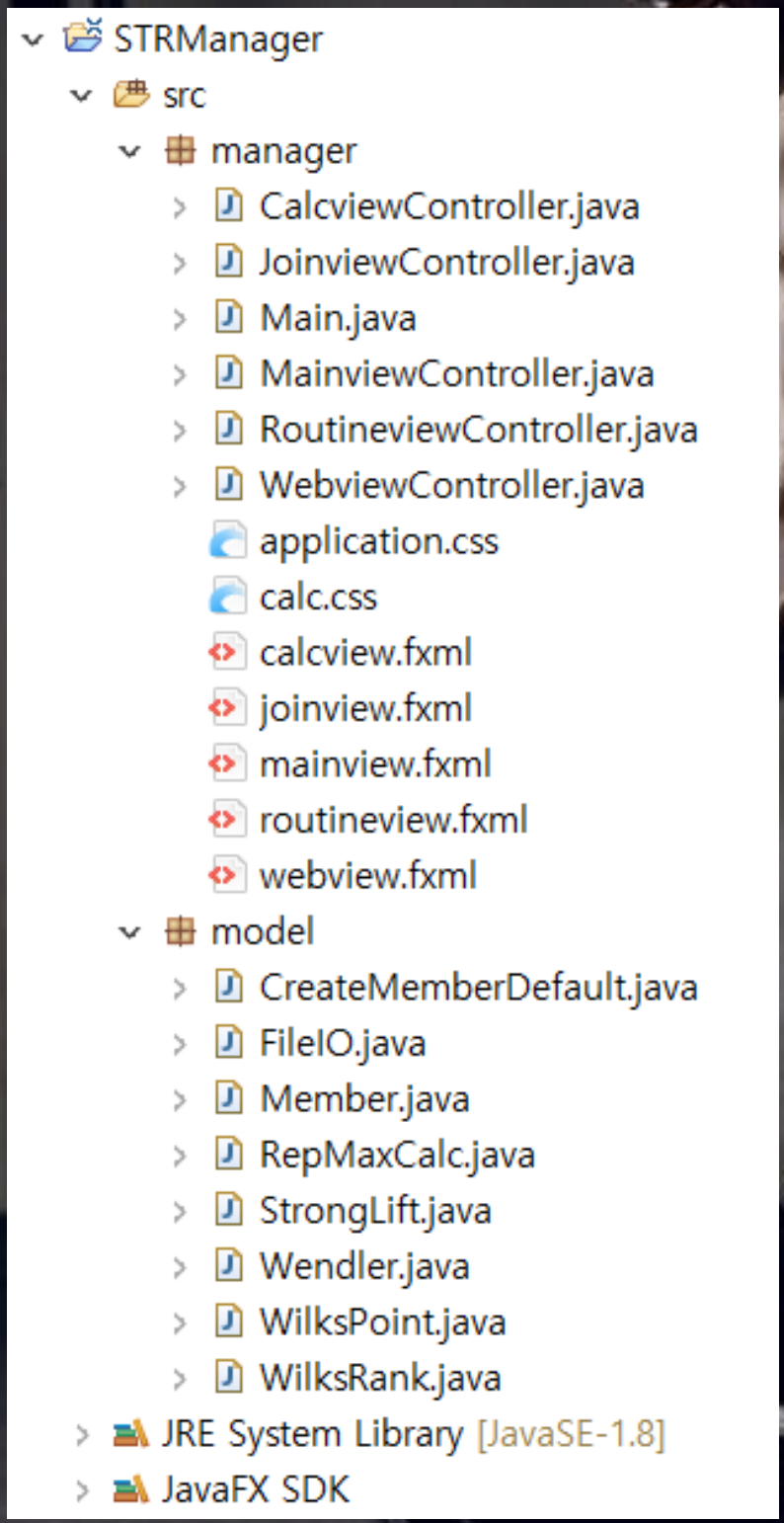
4. 기간 별 수행 구상

기획	(`19.10.15)
설계	(`19.10.16 ~ `19.10.17)
모듈화 전개	(`19.10.18 ~ `19.10.19)
기능 구현	(`19.10.20 ~ `19.10.21)
화면 구현	(`19.10.22 ~ `19.10.24)
1차 테스트	(`19.10.25)

4. 기간 별 수행 구상

일	월	화	수	목	금	토
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15 기획	16	17 설계	18	19 모듈화 전개
20 기능 구현	21	22	23 화면 구현	24	25 테스트	26
27	28	29	30	31	1	2

5. 기능 설계 MVC & CSS



5. 기능 설계 MVC & CSS

Model(package)

Member(회원 정보, 3대 웨이트 포함)

FileIO(실질적 데이터 처리)

CreateMemberDefault(기본 데이터 파일 생성)

RepMaxCalc(RM 공식)

StrongLift(5X5 루틴 공식)

Wendler(531 루틴 공식)

WilksPoint(계산 공식, 공식에 따른 등급)

WilksRank(3대 평균)

Member (회원 정보, 3대 웨이트 포함)

```
private String name;
private String age;
private String gender;
private String address;
private String phonenumber;
private String email;
private double weight;
private double bp1rm;
private double sq1rm;
private double dl1rm;
private double ohp1rm;
private double pr1rm;
private ArrayList<String[]> wendlerRoutine;
private ArrayList<String[]> strongliftRoutine;
```

FileIO(실질적 데이터 처리)

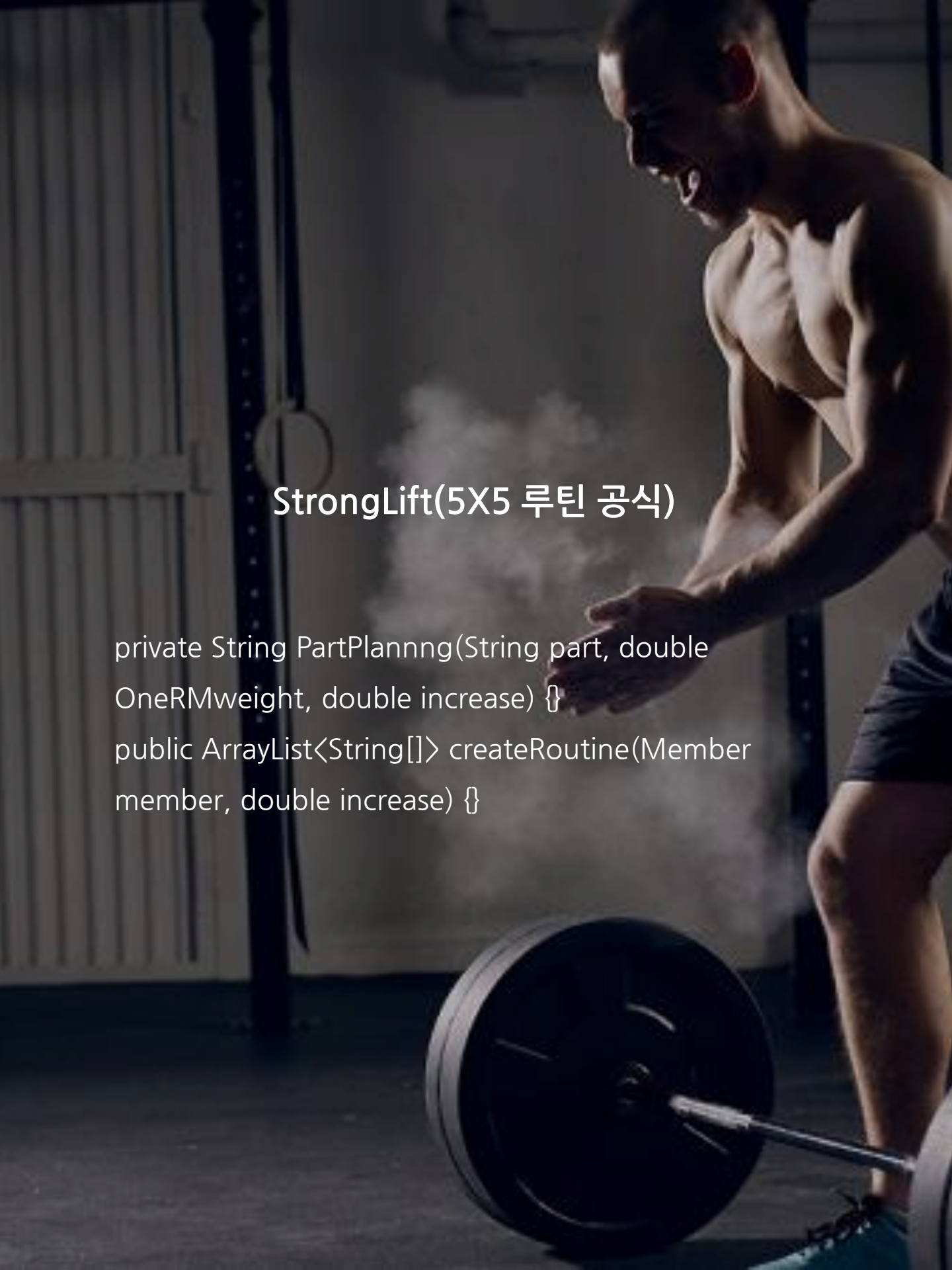
```
public ArrayList<Member> list;
public ObservableList<Member> oblist;
public void savedata() {}
public void savedata(File file) {}
public void loaddata() {}
public void loaddata(File file) {}
public void initModel() {}
```

CreateMemberDefault(기본 데이터 파일 생성)

public CreateMemberDefault() {}

RepMaxCalc(RM 공식)

```
private double lomrm, brzrm, eplrm, mayrm, ocorm,
watrm, lanrm, avgrm;
private double lomonerm, brzonerm, eplonerm,
mayonerm, ocoonerm, watonerm, lanonerm,
avgonerm;
private void calculateOneRM(double weight, double
reps) {}
private void calculateRMs(double i) {}
public ArrayList<String[]> rmCalc(double weight,
double reps) {}
public double Mround(double weight, double
customUnit) {}
```



StrongLift(5X5 루틴 공식)

private String PartPlannng(String part, double OneRMweight, double increase) {}
public ArrayList<String[]> createRoutine(Member member, double increase) {}

Wendler(531 루틴 공식)

```
private double[] intensityCalc(double rmWeight) {}
private String[] waveA(Member member, double
increase) {}
private String[] waveB(Member member, double
increase) {}
private String[] waveC(Member member, double
increase) {}
private String[] waveD(Member member, double
increase) {}
public ArrayList<String[]> createRoutine(Member
member, double increase) {}
```

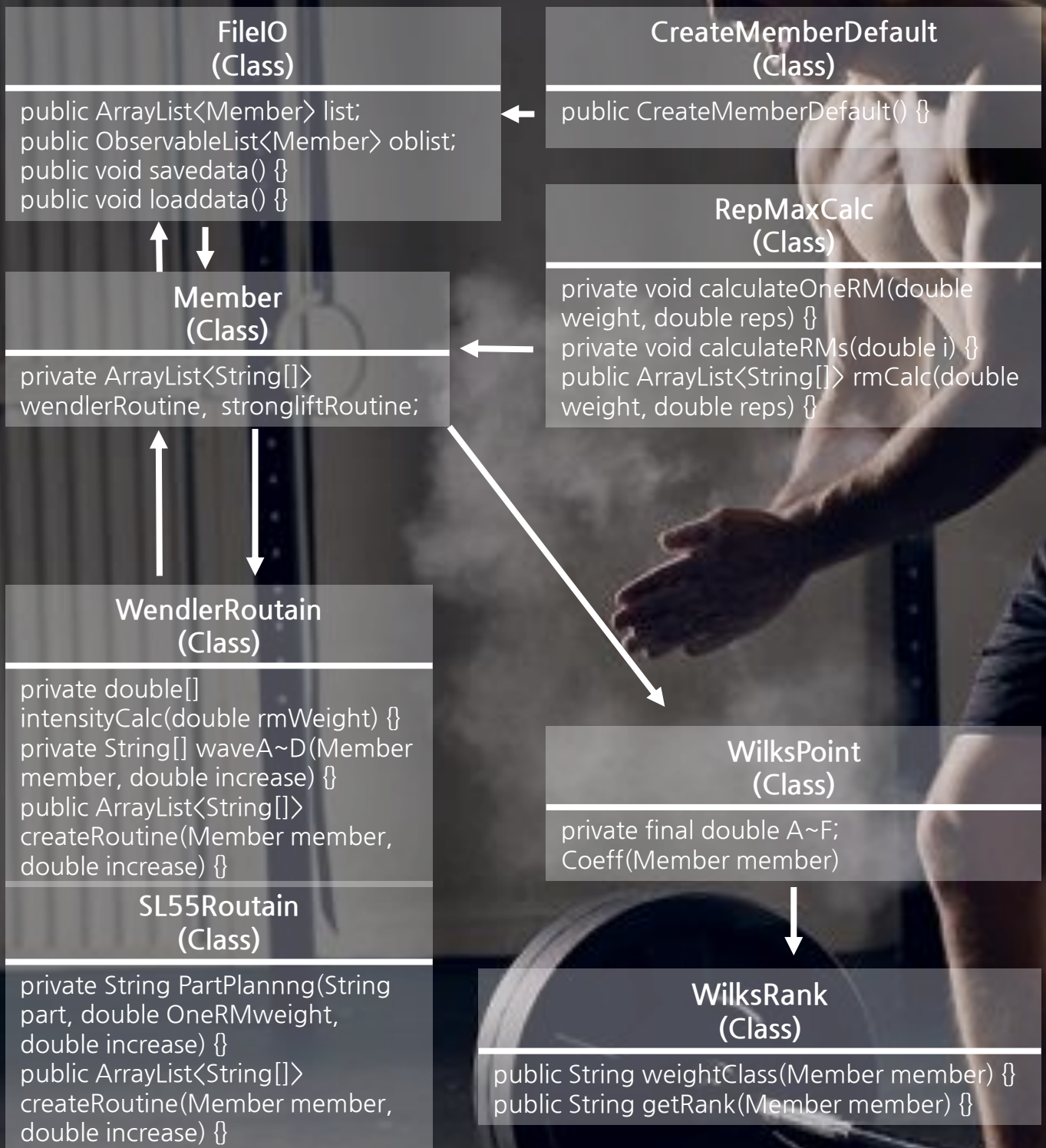
WilksPoint(계산 공식, 공식에 따른 등급)

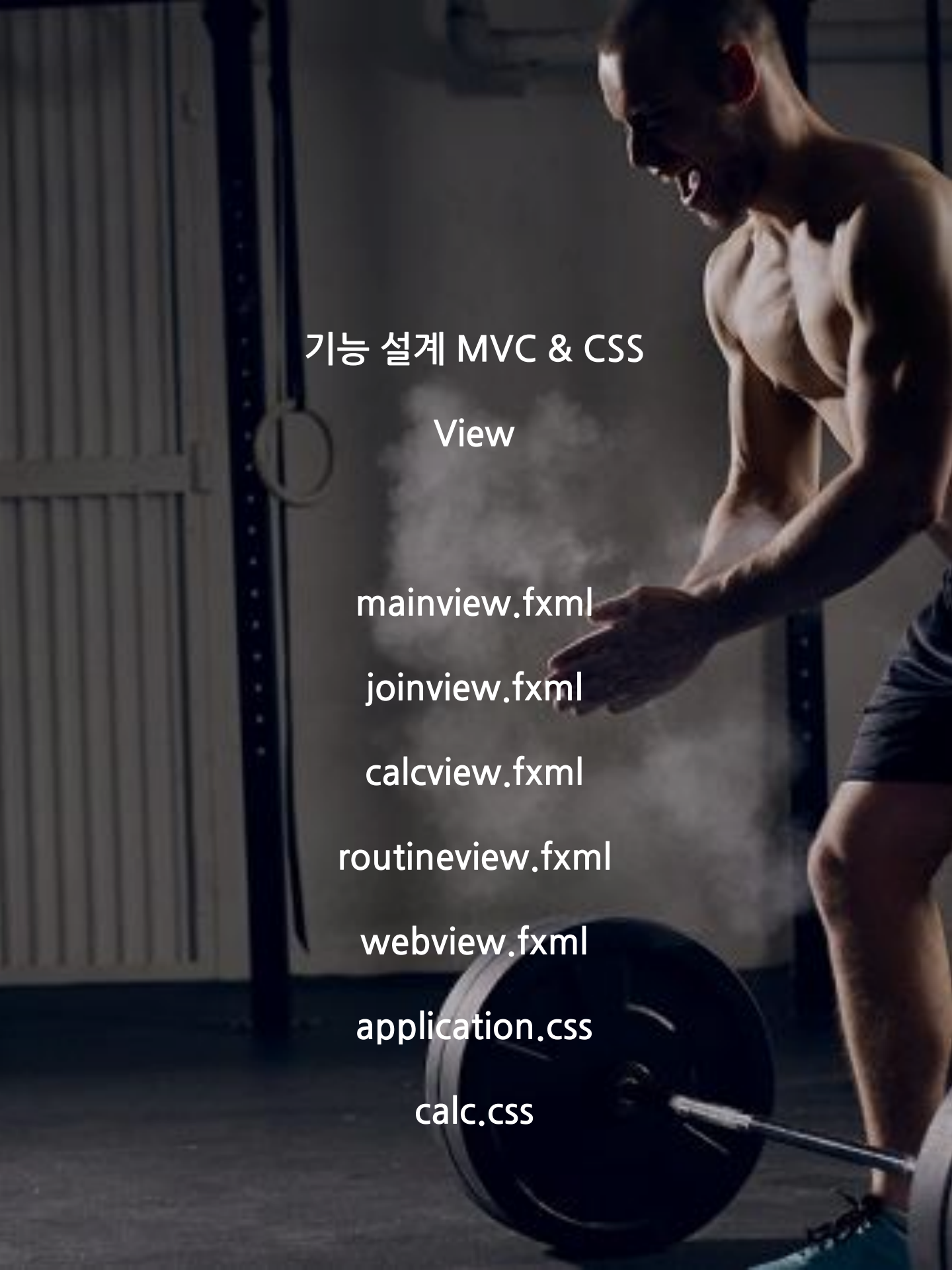
```
private double maleA = -216.0475144;
private double maleB = 16.2606339;
private double maleC = -0.002388645;
private double maleD = -0.00113732;
private double maleE = 7.01863E-06;
private double maleF = -1.291E-08;
private double femaleA = 594.31747775582;
private double femaleB = -27.23842536447;
private double femaleC = 0.82112226871;
private double femaleD = -0.00930733913;
private double femaleE = 4.731582E-05;
private double femaleF = -9.054E-08;
public double Coeff(Member member){}
```

WilksRank(3대 평균)

final private double[][] rankArr = new double [][]{...}
private double[] rankCatch(double weight) {}
public String weightClass(Member member) {}
public String getRank(Member member) {}

기능 설계 - 클래스 관계도


FileIO
(Class)
public ArrayList<Member> list;
public ObservableList<Member> oblist;
public void savedata() {}
public void loaddata() {}
CreateMemberDefault
(Class)
public CreateMemberDefault() {}
RepMaxCalc
(Class)
private void calculateOneRM(double weight, double reps) {}
private void calculateRMs(double i) {}
public ArrayList<String[]> rmCalc(double weight, double reps) {}
Member
(Class)
private ArrayList<String[]> wendlerRoutine, strongliftRoutine;
WendlerRoutain
(Class)
private double[] intensityCalc(double rmWeight) {}
private String[] waveA~D(Member member, double increase) {}
public ArrayList<String[]> createRoutine(Member member, double increase) {}
SL55Routain
(Class)
private String PartPlannng(String part, double OneRMweight, double increase) {}
public ArrayList<String[]> createRoutine(Member member, double increase) {}
WilksPoint
(Class)
private final double A~F;
Coeff(Member member)
WilksRank
(Class)
public String weightClass(Member member) {}
public String getRank(Member member) {}



기능 설계 MVC & CSS

View

mainview.fxml

joinview.fxml

calcview.fxml

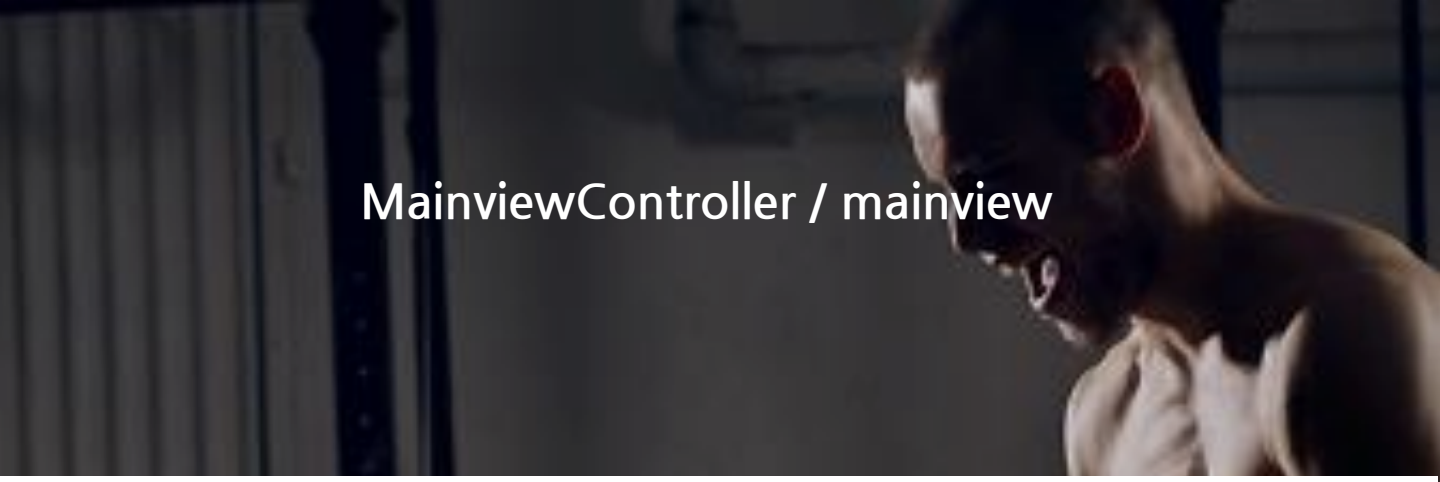
routineview.fxml

webview.fxml

application.css

calc.css


기능 설계 MVC & CSS
Controller
MainviewController
JoinviewController
CalcviewController
RoutineviewController
WebviewController



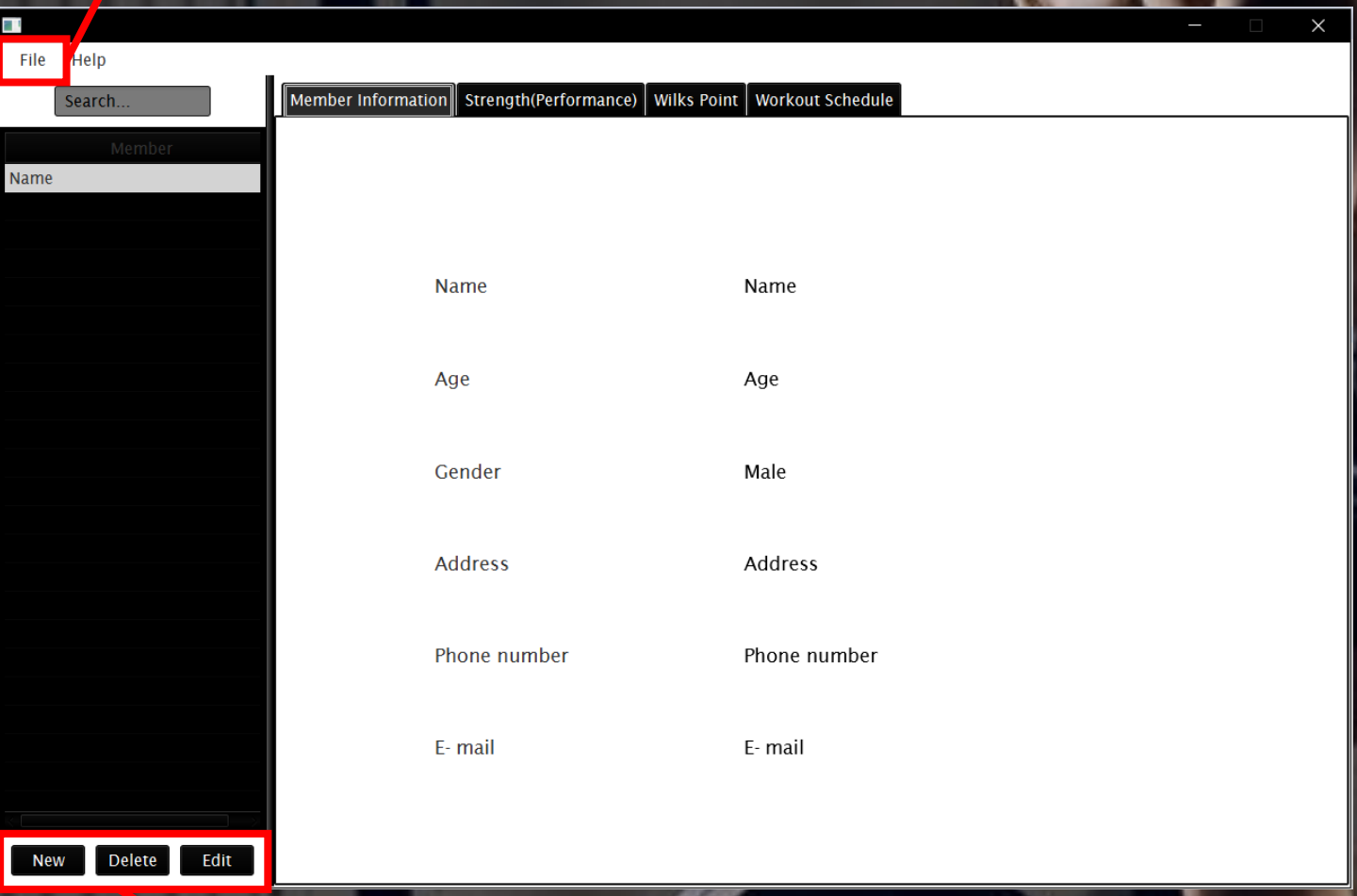
MainviewController / mainview

```
public class MainviewController implements Initializable{
    @FXML AnchorPane ap;
    @FXML TableView<Member> table;
    @FXML TableColumn<Member, String> tableColumn;
    @FXML Button delbtn, editbtn, routinebtn;
    @FXML Text name, age, gender, address, phonenumber, email, weight, bp1rm, sq1rm, dl1rm, ohp1rm, pr1rm,
              wpName, wpBodyweight, wpWilkspoint, wpGender, wpWeightclass, wpLevel;
    @FXML TextField search, editName, editAge, editGender, editPhonenumber, editEmail, editAddress,
              editbodyweight, editbp, editsq, editdl, editohp, editpr;
    @FXML Label strNameGender;
    @FXML GridPane month1, month2, month3;
    @FXML StackPane tableBtnStack, infoStack, strStack, scheduleStack;
    @FXML TabPane tab1, tab2;
    Member member;
    FileIO fio =  new FileIO();
    @Override
    public void initialize(URL location, ResourceBundle resources) {
```

```
@FXML private void open() throws Exception {
@FXML private void save() throws Exception {
@FXML private void quit() throws Exception {    System.exit(0); }
@FXML private void delbtn() {
@FXML private void editbtn() {
@FXML private void modificationbtn() {
private void modification() {
@FXML private void newbtn() {
@FXML private void onermbtn() {
@FXML private void routinebtn() throws IOException {
@FXML private void scheduleRemovebtn() {
private void routineDisplay(ArrayList<String[]> routine) {
private void setHideandShow(String hs, Node...no) {
private EventHandler<KeyEvent> searchEvent = (value)->{
private EventHandler<KeyEvent> editKeyEvent = (value)->{
```

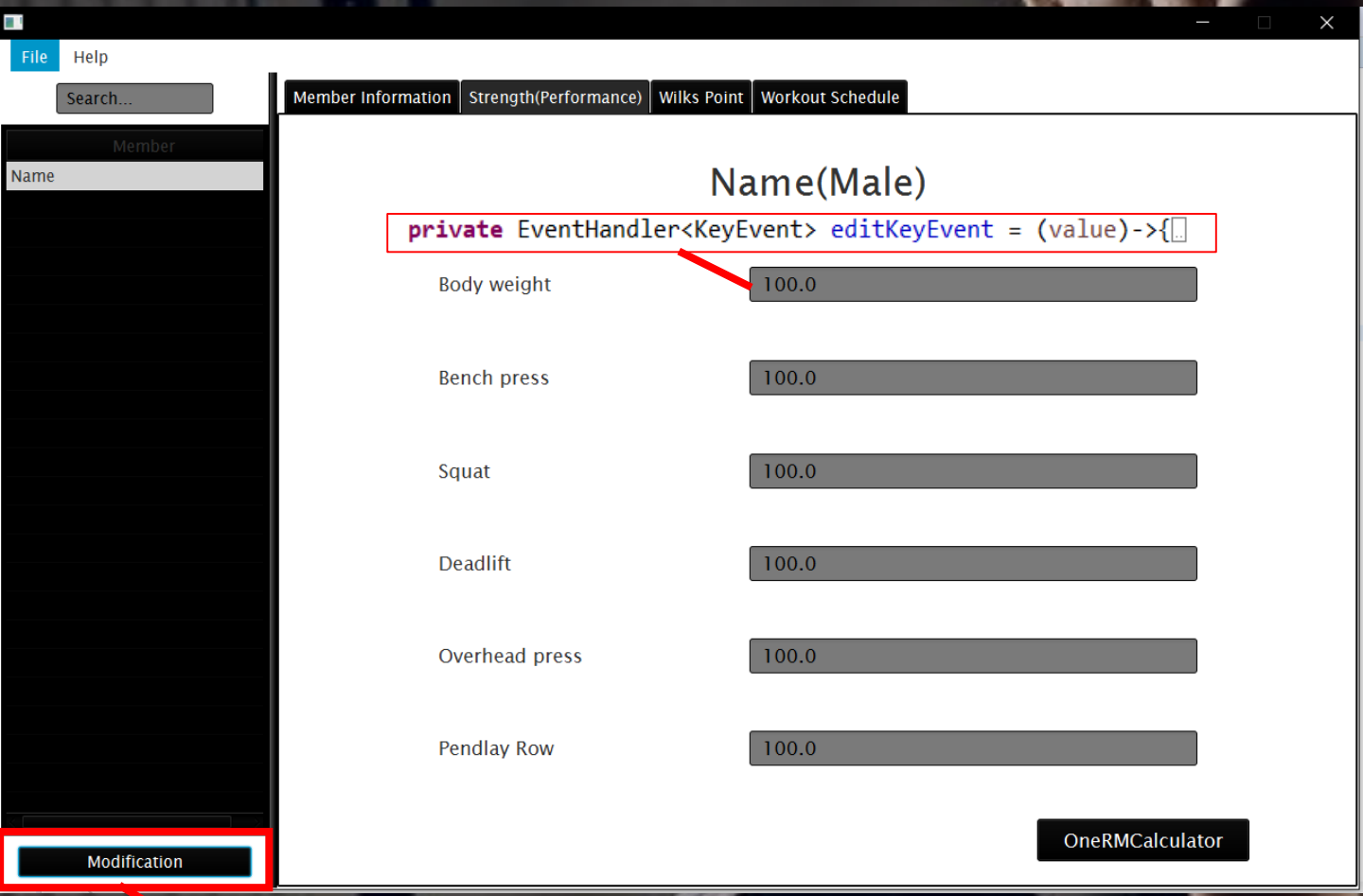
MainviewController / mainview

```
@FXML private void open() throws Exception {
@FXML private void save() throws Exception {
@FXML private void quit() throws Exception {
```



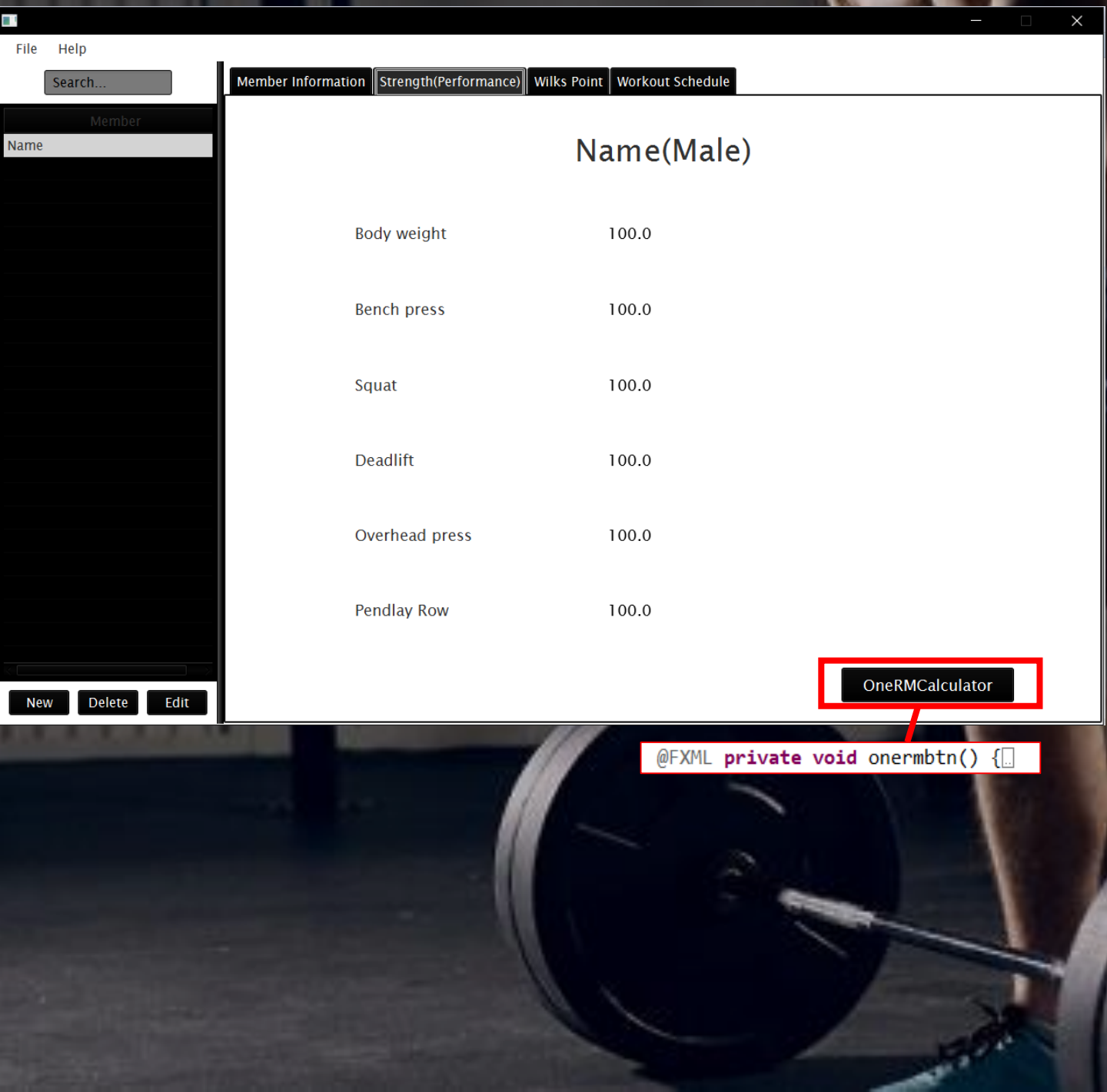

```
@FXML private void delbtn() {
@FXML private void editbtn() {
@FXML private void modificationbtn() {
private void modification() {
@FXML private void newbtn() {
```

MainviewController / mainview

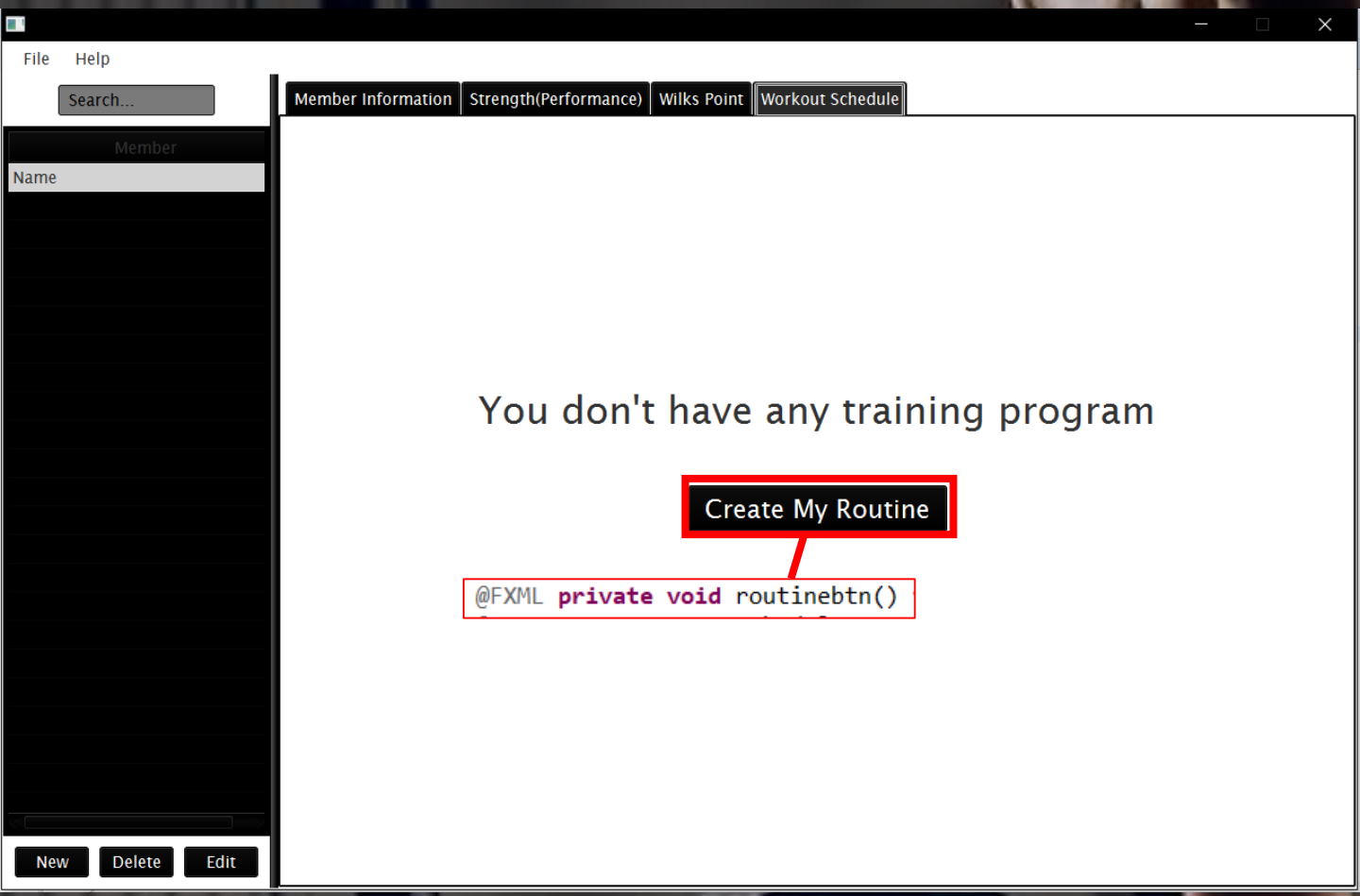



```
@FXML private void modificationbtn() {
private void modification() {
```

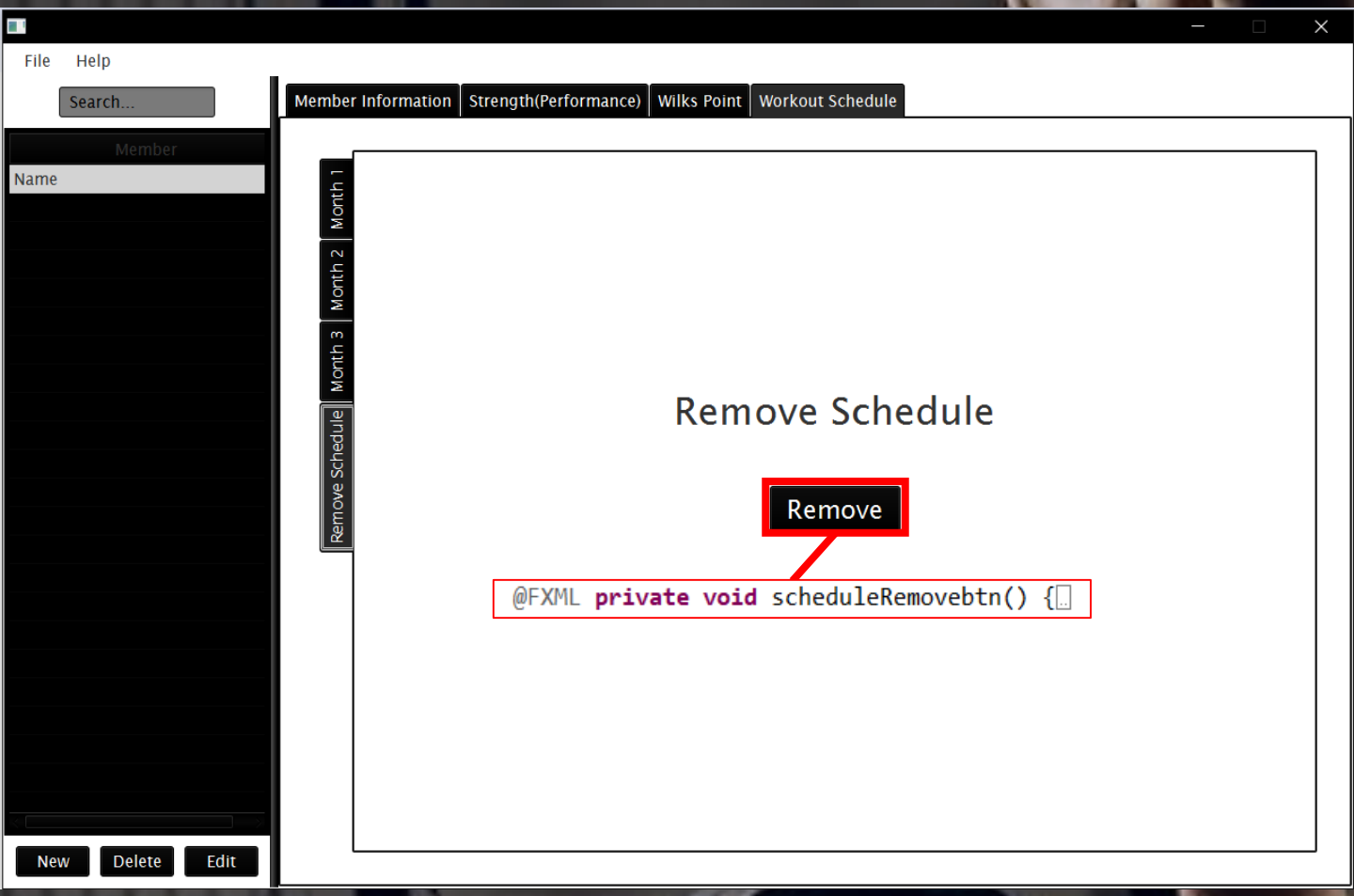
MainviewController / mainview



MainviewController / mainview



MainviewController / mainview

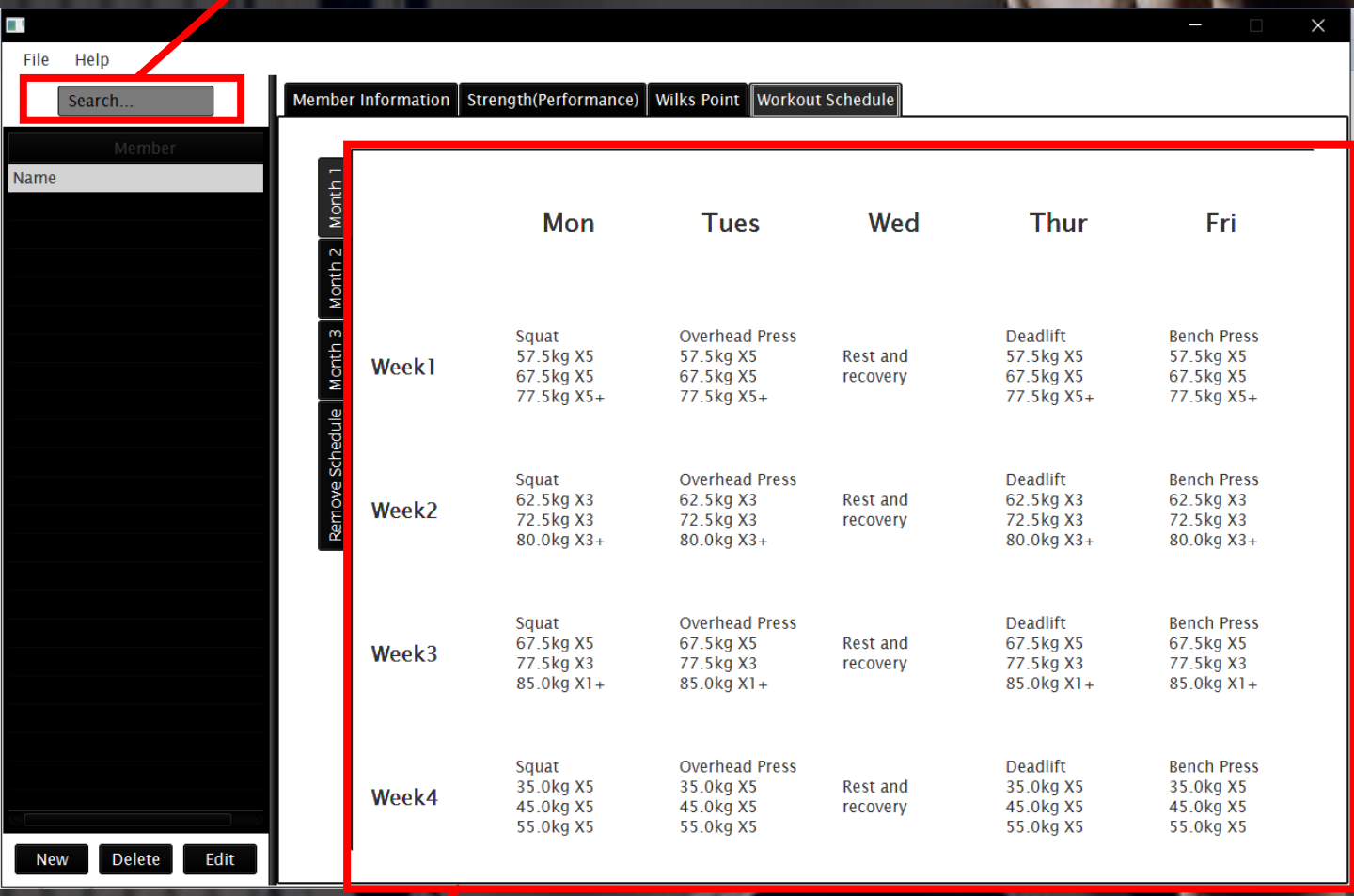



```
private void setHideandShow(String hs, Node...no) {
```

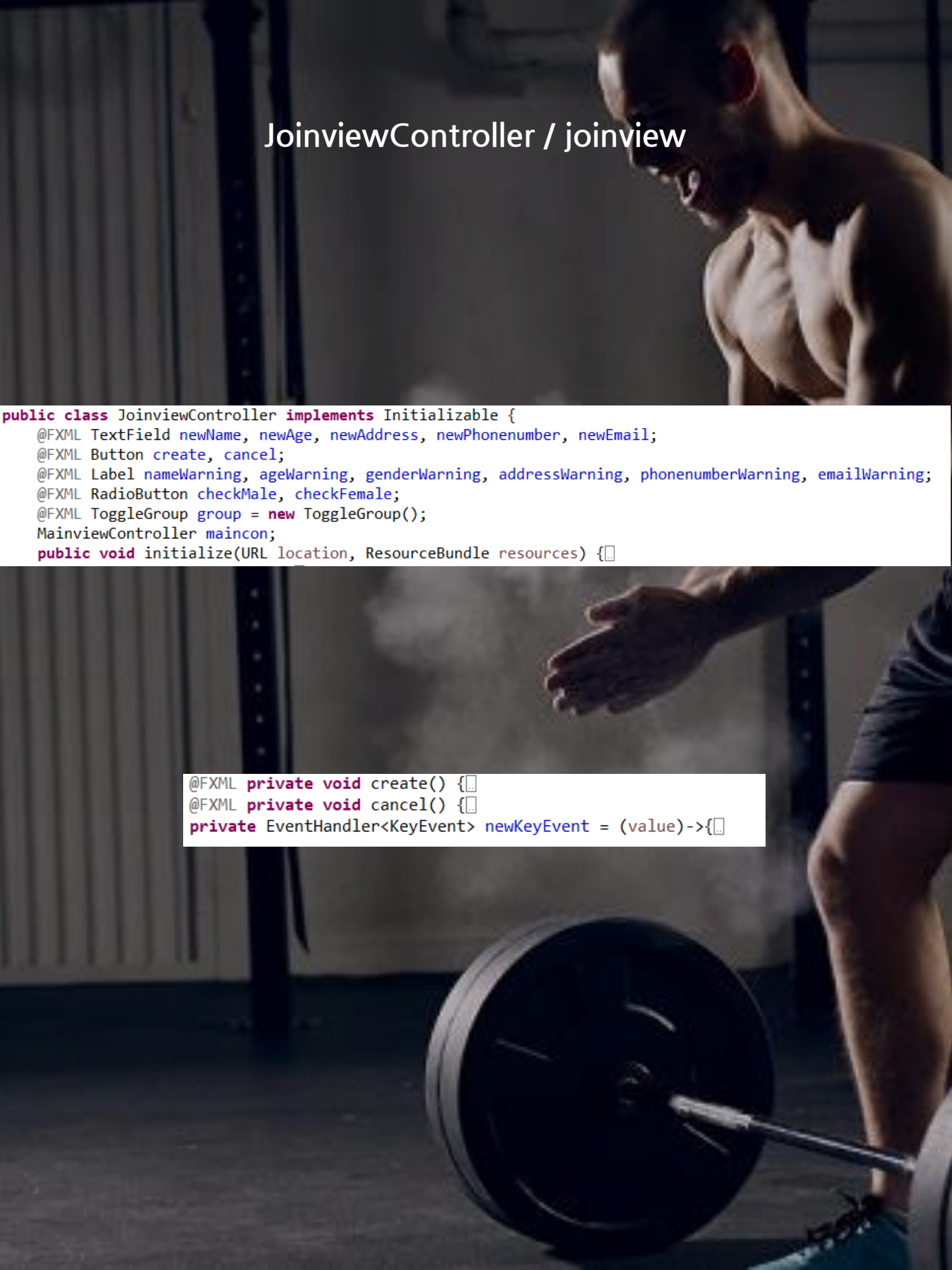
StackPane의 setDisable, setOpacity 설정

MainviewController / mainview

```
private EventHandler<KeyEvent> searchEvent = (value)->{
```




```
private void routineDisplay(ArrayList<String[]> routine) {
```




```java
public class JoinviewController implements Initializable {
    @FXML TextField newName, newAge, newAddress, newPhonenumber, newEmail;
    @FXML Button create, cancel;
    @FXML Label nameWarning, ageWarning, genderWarning, addressWarning, phonenumberWarning, emailWarning;
    @FXML RadioButton checkMale, checkFemale;
    @FXML ToggleGroup group = new ToggleGroup();
    MainviewController maincon;
    public void initialize(URL location, ResourceBundle resources) {
```

```java
@FXML private void create() {
@FXML private void cancel() {
private EventHandler<KeyEvent> newKeyEvent = (value)->{
```

JoinviewController / joinview

```
private EventHandler<KeyEvent> newKeyEvent = (value)->{
```


```
@FXML private void create() {
@FXML private void cancel() {
```

CalcviewController / calcview

```
@FXML Spinner<Integer> weightSpinner;
SpinnerValueFactory<Integer> weightlist = new SpinnerValueFactory.IntegerSpinnerValueFactory(0, 1000, 0);
```

Enter Weight & Reps to get RM Predictions.

I did 100 for 10 reps.

```
@FXML ChoiceBox<Integer> repsChoiceBox;
ObservableList<Integer> repslist = FXCollections.observableArrayList(1,2,3,4,5,6,7,8,9,10);
```

1RM	131	126	133	133	131	125	135	134
2RM	124	118	129	125	118	119	128	129
3RM	120	113	126	121	115	116	124	125
4RM	117	110	122	117	112	114	120	121
5RM	113	107	118	114	110	111	116	118
6RM	111	105	115	111	108	109	112	114
7RM	108	104	111	108	106	106	109	111
8RM	105	102	107	105	104	104	106	107
9RM	102	101	103	102	102	102	103	104
10RM	100	100	100	100	100	100	100	100

Read more about them on Wikipedia

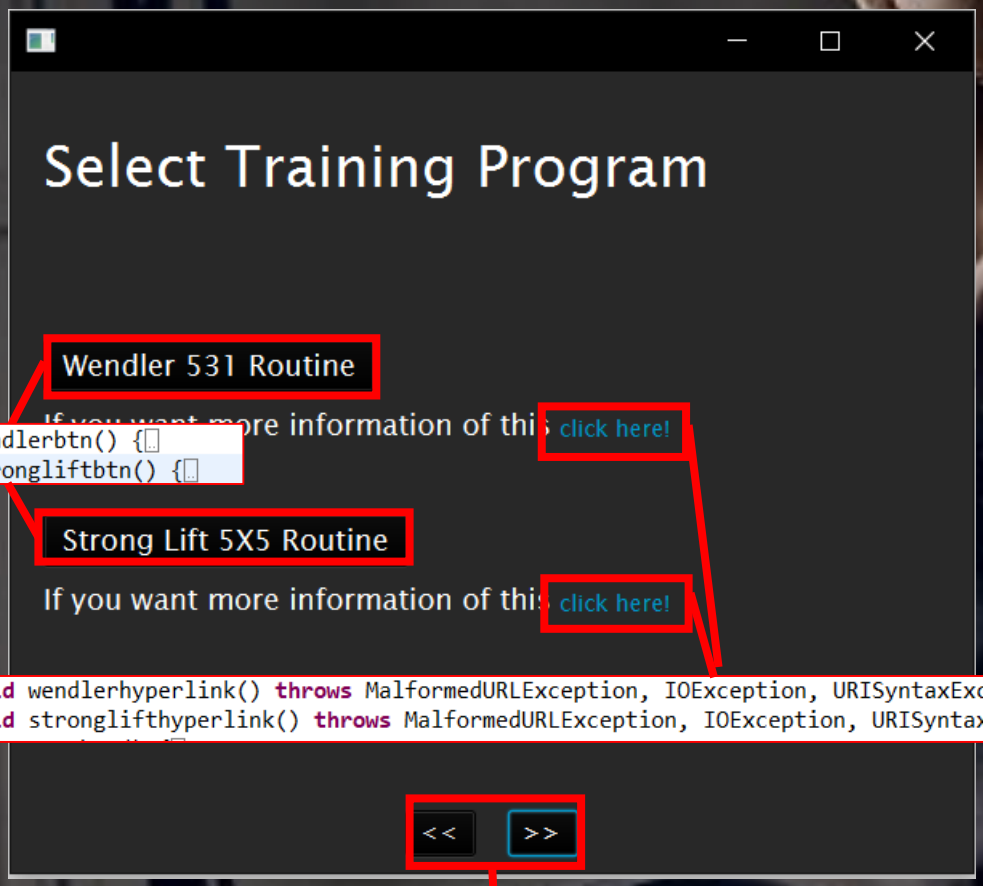
```
@FXML Hyperlink goWikiforFormula;
```

RoutineviewController / routineview

```java
public class RoutineviewController implements Initializable {
    @FXML VBox firstpane, secondpane, thirdpane;
    @FXML Button nextbtn, prevbtn, wendlerbtn, strongliftbtn;
    @FXML Hyperlink wendlerhyperlink, stronglifthyperlink;
    private ArrayList<VBox> panelist = new ArrayList<VBox>();
    private int stackintex = 0;
    private MainviewController maincon;
    public void setMaincon(MainviewController maincon) {
    public void initialize(URL location, ResourceBundle resources) {
```

```java
@FXML void wendlerbtn() {
@FXML void strongliftbtn() {
@FXML private void wendlerhyperlink() throws MalformedURLException, IOException, URISyntaxException {
@FXML private void stronglifthyperlink() throws MalformedURLException, IOException, URISyntaxException {
@FXML private void nextbtn() {
@FXML private void prevbtn() {
```

RoutineviewController / routineview



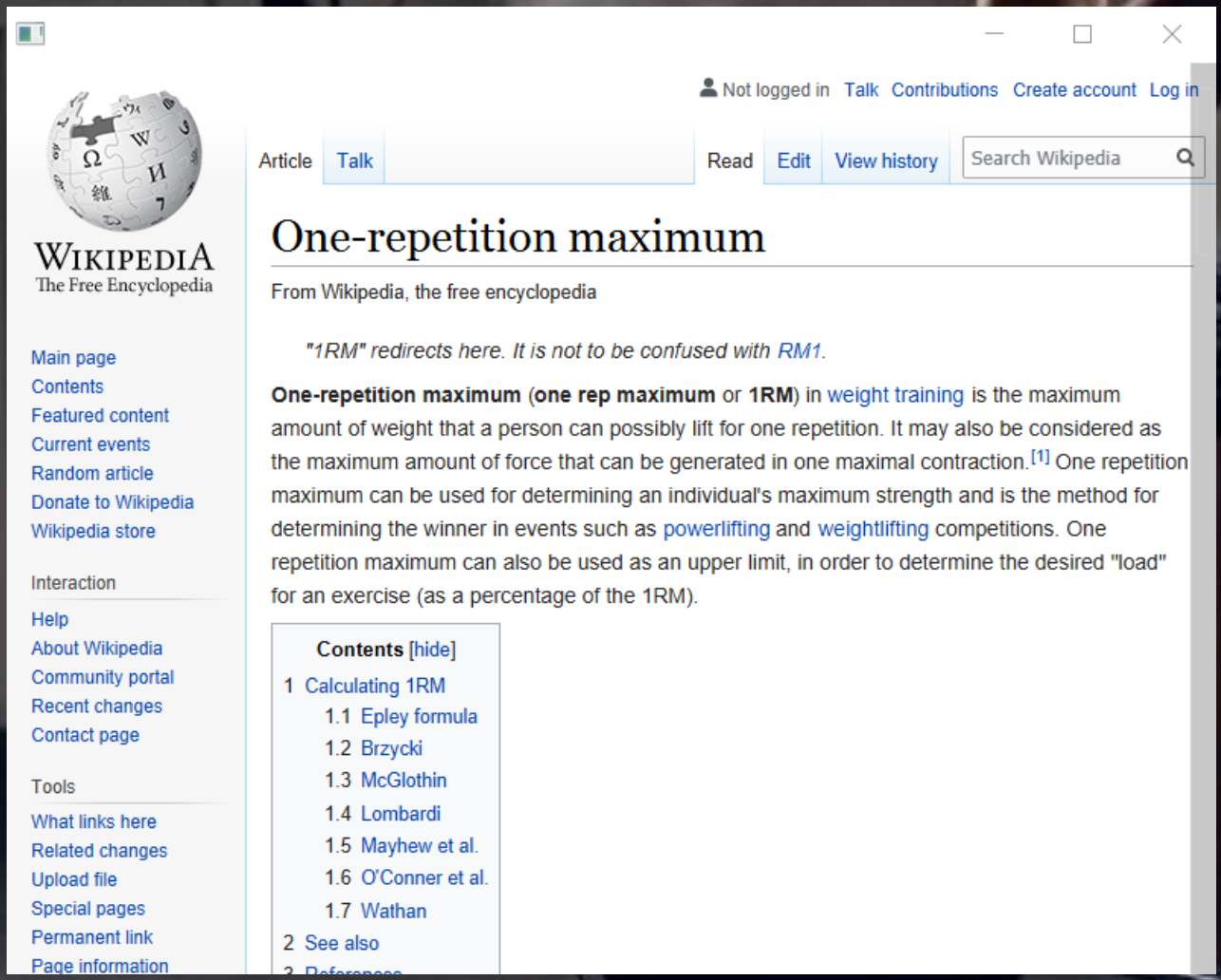

```
@FXML void wendlerbtn() {
@FXML void strongliftbtn() {
```

```
@FXML private void wendlerhyperlink() throws MalformedURLException, IOException, URISyntaxException {
@FXML private void stronglifthyperlink() throws MalformedURLException, IOException, URISyntaxException {
```

```
@FXML private void nextbtn() {
@FXML private void prevbtn() {
```

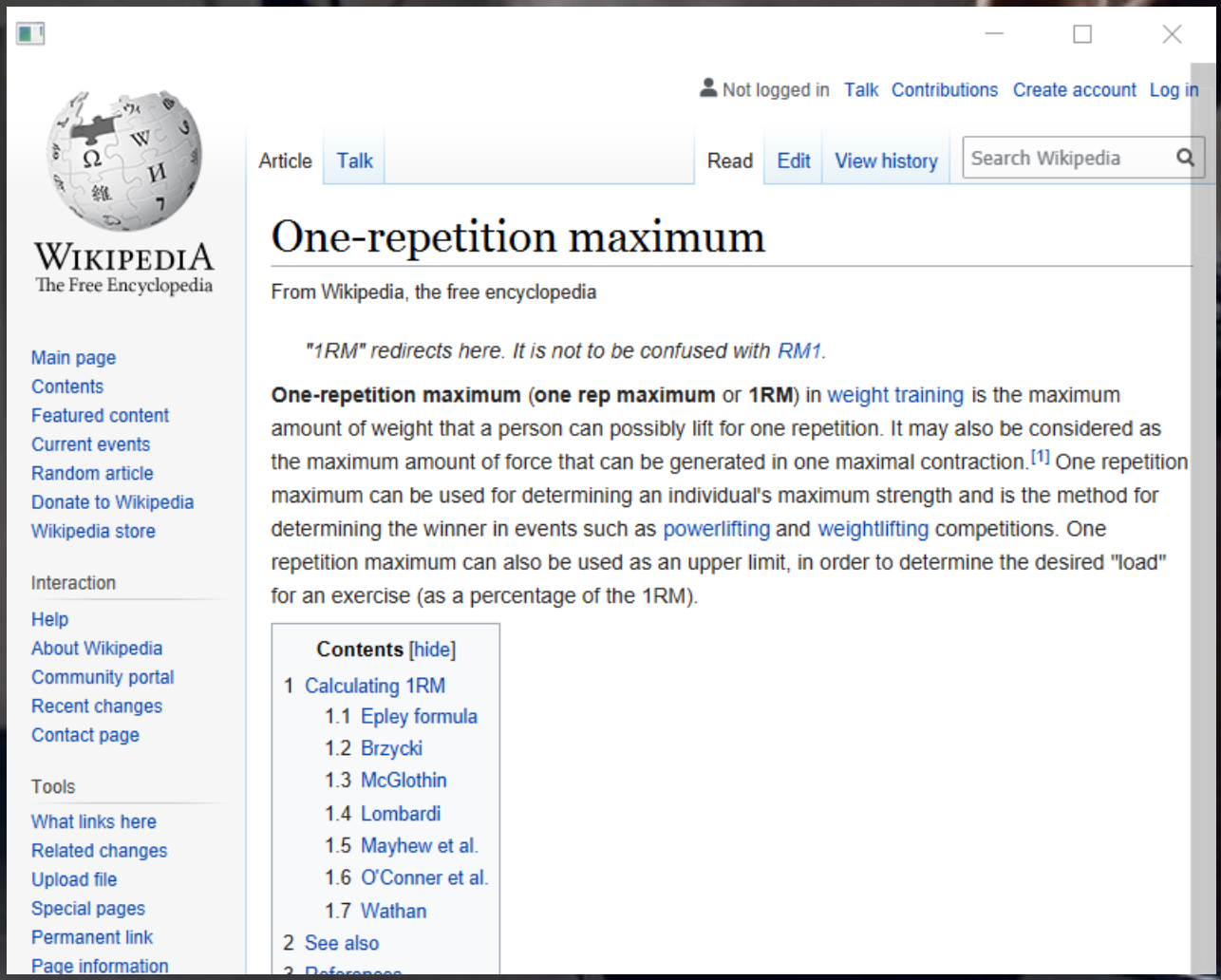
WebviewController / webview

```java
public class WebviewController implements Initializable {
    @FXML WebView webview;
    WebEngine engine;
    @Override
    public void initialize(URL location, ResourceBundle resources) {
        this.engine  = webview.getEngine();
        engine.load("https://en.wikipedia.org/wiki/Wilks_Coefficient");
    }
}
```



WebviewController / webview

```java
public class WebviewController implements Initializable {
    @FXML WebView webview;
    WebEngine engine;
    @Override
    public void initialize(URL location, ResourceBundle resources) {
        this.engine  = webview.getEngine();
        engine.load("https://en.wikipedia.org/wiki/Wilks_Coefficient");
    }
}
```



application.css / calc.css

```css
.root{
    -fx-base: #000000;
}
.split-pane{
    -fx-base: #000000;
}
.table-view {
    -fx-base: #000000;
    -fx-control-inner-background: #000000;
    -fx-background-color: #000000;
    -fx-padding: 5;
    -fx-accent: #ffffff;
}
.label {
    -fx-base: #ffffff;
}
.tab-pane {
    -fx-base: #000000;
    -fx-focus-color: #ffffff;
    -fx-faint-focus-color: #ffffff;
}
.text-field {
    -fx-base: #000000;
    -fx-focus-color: #ffffff;
    -fx-faint-focus-color: #ffffff;
}
```

```css
.root{
    -fx-base: #000000;
}
.label {
    -fx-font-size: 12pt;
    -fx-text-fill: white;
}
```

Strength Manager

프로그래밍 보고서

JAVA WEB개발 김회준

목차

1. 자료 수집, 필수 기능 기획 단계

2. 화면 구현 단계

3. 수정 단계

4. 시행 단계

5. 개선 사항

1. 자료 수집, 필수 기능 기획 단계

모든 Strength 루틴에는 운동 수행자가 가진 각 부위별
One RM(1 Repitition Maximum) 수치가 필요합니다.
따라서 One RM 공식을 wiki를 참조하여 구하고 그에 맞는
java 메서드로 작성할 필요가 있습니다.

본 프로그램은 운동 초보자에 맞춘 프로그램인 만큼 가장 접근성이 좋은 루틴인
Wendler 531 루틴과 StrongLift 55 루틴을 계획해 줍니다.

따라서 각 두개의 루틴에 대한 자료를 수집하고
루틴에 맞는 공식을 각각의 클래스 파일로 작성하여야 합니다.

관리 프로그램에 대한 기본 기능인 CRUD 에 대한 계획을 하고
그에 맞는 Scene builder 구성이 필요합니다.

회원은 기본정보와 루틴을 가지고 있을 루틴에 관련된 필드가 필요합니다.
getter, setter와 constructer를 자동 작성할 것입니다.

이상의 필수 기능에 대한 자료와 기능들을 작성하고
화면 구현에 대한 작성 단계에 들어갔습니다.

2. 화면 구현 단계

프로그램이 기본으로 가져야할 데이터 입력, 출력을 위한 기능을 위해
MenuBar를 사용하였습니다.

많은 다른 프로그램과 같이 Scene의 상위에 위치하게 하여
사용자로 하여금 사용하기 쉽게 만들었습니다.

MenuBar 에는 멤버에 대한 데이터 파일을 저장하고 불러오기 위해
Load와 Save 기능을 가진 Menu Button을 만들었고
화면 구동을 종료할 때 사용 할 Quit 버튼도
SeparatorMenuItem을 이용하여 경계 하단에 위치 시켰습니다.

Help 메뉴에는 프로그램에 대한 도움말과 사용방법,
어떤 의도를 가지고 만들어 졌는지에대해 알려주는 How to Use 와
About Strength Manager 버튼을 만들었습니다.

MenuBar의 밑에는 SplitPane을 이용하여 좌 우로 화면을 분할하고
Search를 위한 Textfield, 데이터를 표시 해 줄 Tableview, 관리프로그램의
CRUD기능을 위한 버튼들을 화면 좌측에 VBox를 이용하여 위치시켰습니다.

화면 우측에는 회원의 정보, 운동 수행 능력, Wilks Point, 운동 스케줄을 위한
TabPane을 작성하고 각 Tab에 그에 맞는 회원 정보를 나타낼 수 있게
Text와 GridPane을 추가로 위치 시켰습니다.

3. 수정 단계

회원은 루틴에 대한 추가적인 필드를 갖는데 이는
constructer로는 set 할 수 없게 하였습니다. (생성자에서 제외됨)
만약 루틴에 대한 둘 중 하나의 필드의 값이 null이 아니라면
후에 루틴을 가질 수 있도록 구성을 해야 합니다.

본 프로그램은 회원의 이름을 table view를 사용하여 표시하고, 그것을 이용하여
관리하기 위해 Name 필드에 대한 추가적인 getter 메소드가 필요하였습니다.
멤버의Name필드에 대한 value를 SimpleStringProperty로 반환하는
getNameSSP 메서드를 작성하였습니다.

루틴은 일반적으로 8주에서 12주까지의 기간이 소요 되므로,
월, 주, 일의 운동 스케줄에 대한 다차원 배열이 필요하였습니다.
처음에는 월, 주, 일에 대한 3차원의 배열을 만들었지만
화면 표시에 대한 어려움이 있어서 루틴을 return하는 메서드를
월, 주를 가진 2차원 배열을 return하도록 수정하였습니다.
day routine에 대한 workout schedule은
하나의 String으로 반환하도록 만들었습니다.

static 필드에 대한 수정을 하였습니다.
메서드를 객체를 선언하고 사용하지 않기 위해
단순한 계산 메서드들을 static으로 작성하였습니다.
하지만 메서드들이 늘어남에 따라 메서드에서 사용될 필드와
부가적인 또 다른 메서드들이 늘어나 메모리 사용 이점에 대해 고민 후
static 메서드를 보통의 메서드로 수정하는 작업을 하였습니다.

Model 클래스 또한 static으로 작성되어 파일 입 출력에 대해 이점을 갖게 하였지
만 후에 네트워크프로그램으로 프로그램의 활동 범위가 확장되면
오류를 유발 할 수 있어 Model클래스에 대한 수정이 이루어 졌습니다.

4. 시행 단계

FileChooser 에 대해 FilePath를 return받아
Model에서의 File 입출력이 이루어 집니다.

하지만 return받은 String은 ₩가 하나로 입력되어
replace 메서드를 사용하여 ₩를 2회 입력으로 바꾸어 주어야 했습니다.

File을 불러오거나 저장할 때 MainContoller의 filePath String필드에
File에 대한 절대주소값을 저장하고 그것으로 사용하려 했지만
Mainview가 호출(.load)될 때 마다 filePath또한 초기화가 되어
이 부분에 어려움이 있습니다.

또한 MainView의 Controller가 불러질 때마다(.load) 데이터를 관리하는
FileIO(실질적 Model)의 객체가 새로 생성되기에
MainView를 최소한으로 불러올 필요가 있겠습니다.

위와 같은 어려움에 데이터가 load되거나 save될 때 FileIO(Model)의
예전 static data 필드 기능을 대신해줄 임시 데이터 파일이 필요하여
모든 윈도우 사용자가 가지고 있을 C:₩Users₩Public 경로에
Member.dat 파일을 항상 저장하고 불러올 수 있게 만들었습니다.

이 때문에 프로그램을 실행 하게 되면 가장 최근에 작업하고 있던 File이
Member.dat File로서 불러와 지게 됩니다.

5. 개선 사항

만약 Model이 MainViewController에서 만들어 진다면
MainView는 한 번만 load 되는 것이 좋겠습니다.

하지만 데이터에 변경사항이 생긴다면 새로고침 할 필요가 있습니다.

따라서 MainView는 고정 시키되 그 안에 MainViewController가 가진
Model을 참조 받는 새로운 fxml file을 불러와
Scene에 적용시키면서 새로고침과 같은
반복적인 load를 사용하는 것이 좋겠습니다.

Strength Manager

프로그래밍 참조자료

JAVA WEB개발 김회준

목차

1. Model

2. Controller

1. Model - CreateMemberDefault

```
public class CreateMemberDefault {
    FileIO fio = new FileIO();
    public CreateMemberDefault() {
        try {
            Member member = new Member("name", "30", "Male", "address", "8210-nnnn-nnnn",
                    "email@mail.com", 100, 110, 130, 160, 80, 100);
            File f = new File("Member.dat");
            ArrayList<Member> memberList = new ArrayList<Member>();
            memberList.add(member);
            fio.list = memberList;
            FileOutputStream fos = new FileOutputStream(f);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(fio.list);
            oos.close();
        } catch (Exception e) {}
    }
}
```

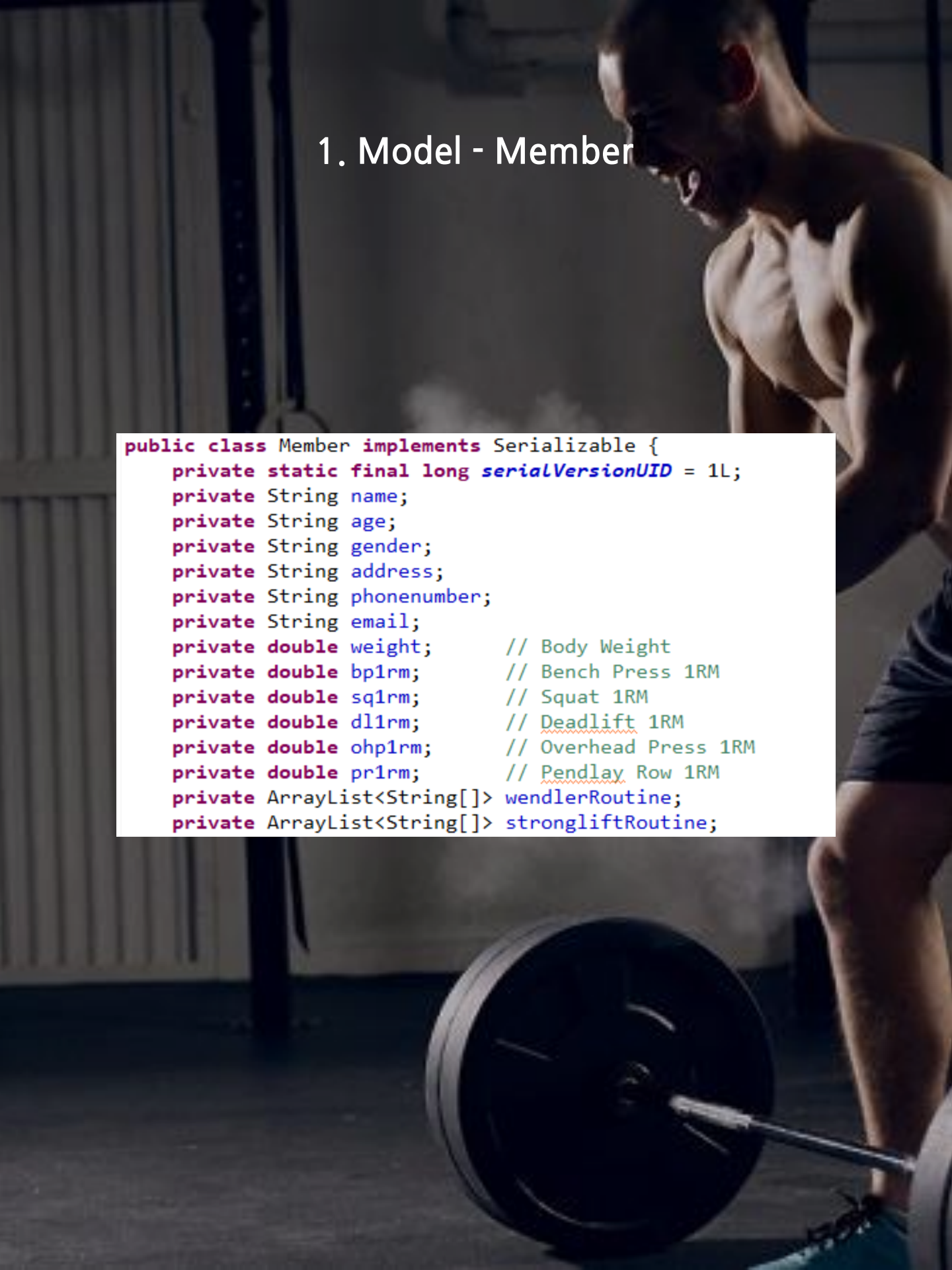
1. Model - FileIO

```java
@SuppressWarnings("serial")
public class FileIO implements Serializable{
    public ArrayList<Member> list;
    public ObservableList<Member> oblist;
    public void savedata() {
        try {
            File f = new File("C:\\Users\\Public\\Member.dat");
            FileOutputStream fos = new FileOutputStream(f);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(list);
            oos.close();
        } catch (Exception e) {}
    }
    public void savedata(File file) {
        try {
            FileOutputStream fos = new FileOutputStream(file);
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(list);
            oos.close();
        } catch (Exception e) {}
    }
```

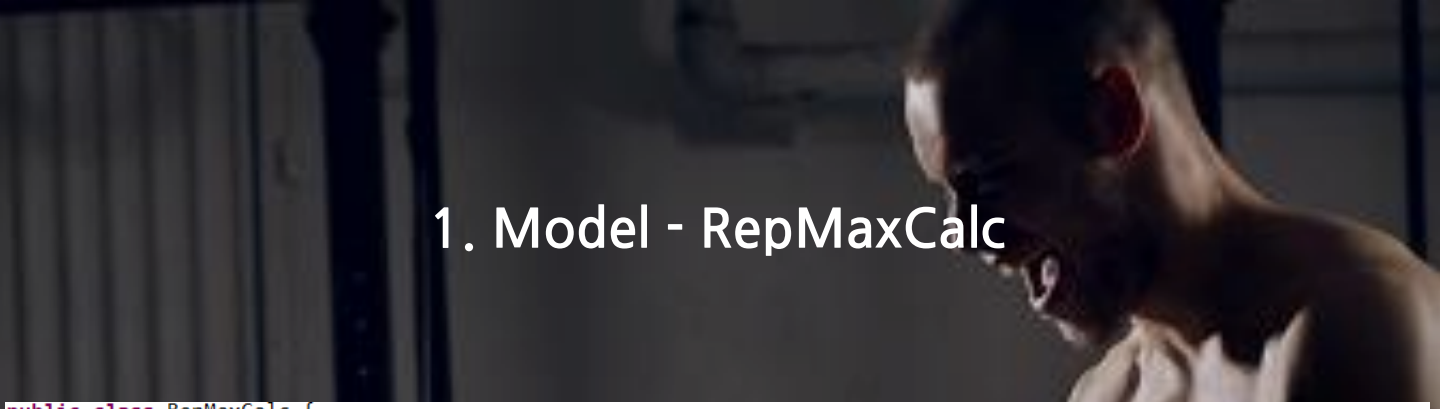
1. Model - FileIO

```
@SuppressWarnings("unchecked")
public void loaddata() {
    try {
        File f = new File("C:\\Users\\Public\\Member.dat");
        FileInputStream fis = new FileInputStream(f);
        ObjectInputStream ois = new ObjectInputStream(fis);
        list = (ArrayList<Member>)ois.readObject();
        ois.close();
    } catch (Exception e) {}
}
@SuppressWarnings("unchecked")
public void loaddata(File file) {
    try {
        FileInputStream fis = new FileInputStream(file);
        ObjectInputStream ois = new ObjectInputStream(fis);
        list = (ArrayList<Member>)ois.readObject();
        ois.close();
    } catch (Exception e) {}
}
public void initModel() {
    oblist = FXCollections.observableArrayList();
    for(int i = 0 ; i < list.size() ; i++) {
        oblist.add(list.get(i));
    }
}
```

1. Model - Member



```java
public class Member implements Serializable {
    private static final long serialVersionUID = 1L;
    private String name;
    private String age;
    private String gender;
    private String address;
    private String phonenumber;
    private String email;
    private double weight;      // Body Weight
    private double bp1rm;       // Bench Press 1RM
    private double sq1rm;       // Squat 1RM
    private double dl1rm;       // Deadlift 1RM
    private double ohp1rm;      // Overhead Press 1RM
    private double pr1rm;       // Pendlay Row 1RM
    private ArrayList<String[]> wendlerRoutine;
    private ArrayList<String[]> stronglift5Routine;
```

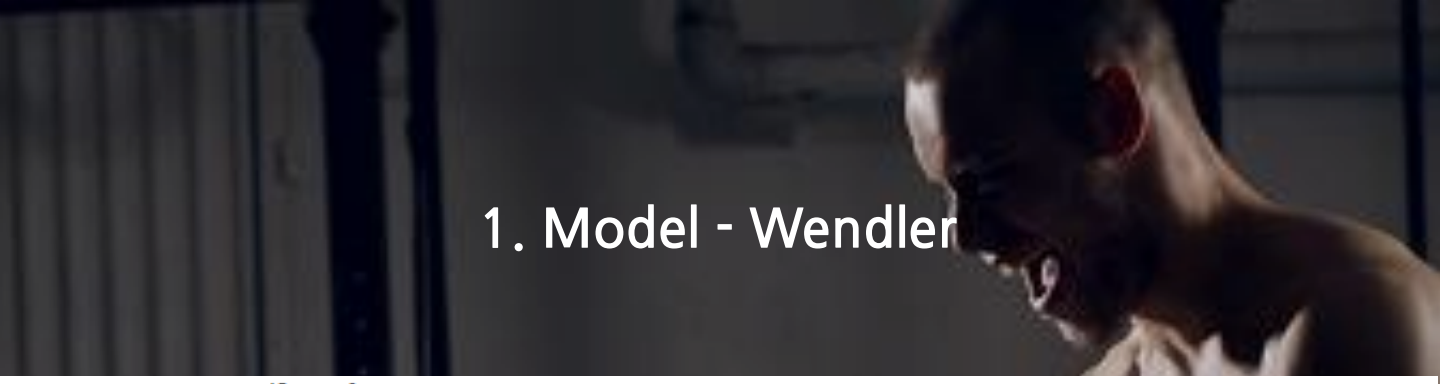


1. Model - RepMaxCalc

```
public class RepMaxCalc {
    private double lomonerm, brzonerm, eplonerm, mayonerm, ocoonerm, watonerm, lanonerm, avgonerm;
    //for all reps > 1 calculate the 1RMs
    private void calculateOneRM(double weight, double reps) {
        lomonerm = Math.round(weight * Math.pow(reps, 0.10));
        brzonerm = Math.round(weight * (36 / (37 - reps)));
        eplonerm = Math.round(weight * (1 + (reps / 30)));
        mayonerm = Math.round((weight * 100) / (52.2 + (41.9 * Math.exp(-1 * (reps * 0.055)))));
        ocoonerm = Math.round(weight * (1 + reps * 0.025));
        watonerm = Math.round((weight * 100) / (48.8 + (53.8 * Math.exp(-1 * (reps * 0.075)))));
        lanonerm = Math.round(weight * 100 / (101.3 - 2.67123 * reps));
        avgonerm = Math.round((lomonerm + brzonerm + eplonerm + mayonerm + ocoonerm + watonerm + lanonerm) / 7);
    }
    private double lomrm, brzrm, eplrm, mayrm, ocorm, watrm, lanrm, avgrm;
    // calculate RMs 2-10 and append it to the table
    private void calculateRMs(double i) {
        lomrm = Math.round(lomonerm / (Math.pow(i, 0.1)));
        brzrm = Math.round((brzonerm * (37 - i)) / 36);
        eplrm = Math.round(eplonerm / ((1 + (i / 30))));
        mayrm = Math.round((mayonerm * (52.2 + (41.9 * Math.exp(-1 * (i * 0.055))))) / 100);
        ocorm = Math.round((ocoonerm / (1 + i * 0.025)));
        watrm = Math.round((watonerm * (48.8 + (53.8 * Math.exp(-1 * (i * 0.075))))) / 100);
        lanrm = Math.round(((lanonerm * (101.3 - 2.67123 * i)) / 100));
        avgrm = Math.round((lomrm + brzrm + eplrm + mayrm + ocorm + watrm + lanrm) / 7);
    }
public ArrayList<String[]> rmCalc(double weight, double reps) {
    calculateOneRM(weight, reps);
    DecimalFormat df = new DecimalFormat("0");
    String[] info = {"RM","Average","Lombardi","Brzycki","Epley","Mayhew","O'Conner","Wathan","Lander"};
    String[] oneRM = {"1RM", df.format(avgonerm), df.format(lomonerm), df.format(brzonerm), df.format(eplonerm),
            df.format(mayonerm), df.format(ocoonerm), df.format(watonerm), df.format(lanonerm)};
    ArrayList<String[]> rm = new ArrayList<String[]>();
    rm.add(info);       rm.add(oneRM);
    for(int i = 2 ; i <= 10 ; i++) {
        calculateRMs(i);
        String[] tmp = {i+"RM", df.format(avgrm), df.format(lomrm), df.format(brzrm), df.format(eplrm),
                df.format(mayrm), df.format(ocorm), df.format(watrm), df.format(lanrm)};
        rm.add(tmp);
    }
    return rm;
}
public double Mround(double weight, double customUnit) {        // MROUND of Excel
    double unitsDigit = weight;
    while(unitsDigit > 10) {unitsDigit %= 10;}
    double remainder = unitsDigit % customUnit;
    double customNum = unitsDigit - remainder;
    double round = 0;
    if(remainder>customUnit / 2) {round = customUnit;}
    else {round = 0;}
    return weight - unitsDigit + customNum + round;
}
```

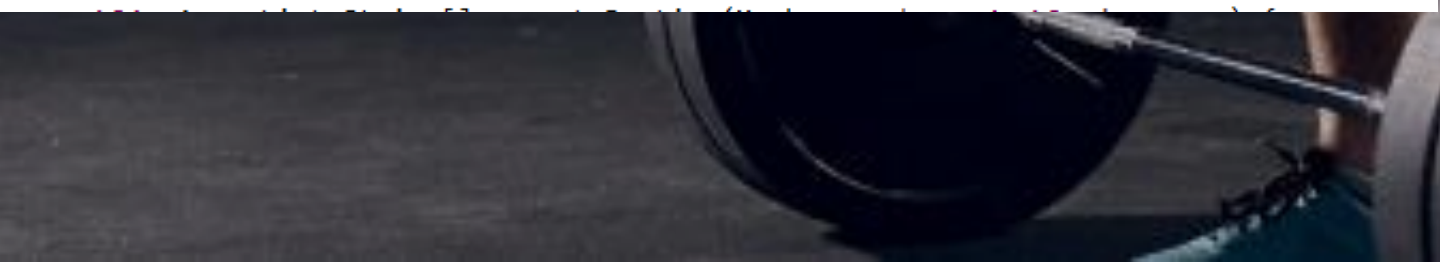
1. Model - StrongLift

```
public class StrongLift {
private String PartPlannng(String part, double OneRMweight, double increase) {
RepMaxCalc rmc = new RepMaxCalc();
double weight = rmc.Mround(OneRMweight, increase) / 2;
String result, workOut = null;
//String warmUp1 = ((weight * 0.7)+increase) + "kg 5X1\n";
//String warmUp2 = ((weight * 0.8)+increase) + "kg 5X1\n";
//String warmUp3 = ((weight * 0.9)+increase) + "kg 5X1\n";
if(part.equals("Deadlift")) {
//warmUp1 = ((weight * 0.7)+increase) + "kg 5X1\n";
//warmUp2 = ((weight * 0.8)+increase) + "kg 5X1\n";
//warmUp3 = ((weight * 0.9)+increase) + "kg 5X1\n";
workOut = (weight+increase) + "kg 1X5\n";}
else if(part.equals("Squat")) {
workOut = (weight+increase) + "kg 5X5\n";
}
else {
workOut = (weight+increase) + "kg 5X5\n";}
result = part + "\n" + /* warmUp1 + warmUp2 + warmUp3 + */ workOut;
return result;
}
public ArrayList<String[]> createRoutine(Member member, double increase) {
double sq = member.getSq1rm();
double bp = member.getBp1rm();
double pr = member.getPr1rm();
double ohp = member.getOhp1rm();
double dl = member.getDl1rm();
ArrayList<String[]> monthSet = new ArrayList<String[]>();
String worka1 = PartPlannng("Squat", sq, increase*0) + PartPlannng("Bench Press", bp, increase*0) + PartPlannng("Pendlay Row", pr, increase*0);
String workb = PartPlannng("Squat", sq, increase*1) + PartPlannng("Overhead Press", ohp, increase*0) + PartPlannng("Deadlift", dl, increase*0);
String worka2 = PartPlannng("Squat", sq, increase*2) + PartPlannng("Bench Press", bp, increase*1) + PartPlannng("Pendlay Row", pr, increase*1);
String[] result1 = new String[] {worka1, "Rest and \nrecovery", workb, "Rest and \nrecovery", worka2};
String workb1 =PartPlannng("Squat", sq, increase*3) + PartPlannng("Overhead Press", ohp, increase*1) + PartPlannng("Deadlift", dl, increase*2);
String worka = PartPlannng("Squat", sq, increase*4) + PartPlannng("Bench Press", bp, increase*2) + PartPlannng("Pendlay Row", pr, increase*2);
String workb2 =PartPlannng("Squat", sq, increase*5) + PartPlannng("Overhead Press", ohp, increase*2) + PartPlannng("Deadlift", dl, increase*4);
String[] result2 = new String[] {workb1, "Rest and \nrecovery", worka, "Rest and \nrecovery", workb2};
worka1 = PartPlannng("Squat", sq, increase*6) + PartPlannng("Bench Press", bp, increase*3) + PartPlannng("Pendlay Row", pr, increase*3);
workb = PartPlannng("Squat", sq, increase*7) + PartPlannng("Overhead Press", ohp, increase*3) + PartPlannng("Deadlift", dl, increase*6);
worka2 = PartPlannng("Squat", sq, increase*8) + PartPlannng("Bench Press", bp, increase*4) + PartPlannng("Pendlay Row", pr, increase*4);
String[] result3 = new String[] {worka1, "Rest and \nrecovery", workb, "Rest and \nrecovery", worka2};
workb1 = PartPlannng("Squat", sq, increase*9) + PartPlannng("Overhead Press", ohp, increase*4) + PartPlannng("Deadlift", dl, increase*8);
worka = PartPlannng("Squat", sq, increase*10) + PartPlannng("Bench Press", bp, increase*5) + PartPlannng("Pendlay Row", pr, increase*5);
workb2 = PartPlannng("Squat", sq, increase*11) + PartPlannng("Overhead Press", ohp, increase*5) + PartPlannng("Deadlift", dl, increase*10);
String[] result4 = new String[] {workb1, "Rest and \nrecovery", worka, "Rest and \nrecovery", workb2};
worka1 = PartPlannng("Squat", sq, increase*12) + PartPlannng("Bench Press", bp, increase*6) + PartPlannng("Pendlay Row", pr, increase*6);
workb = PartPlannng("Squat", sq, increase*13) + PartPlannng("Overhead Press", ohp, increase*6) + PartPlannng("Deadlift", dl, increase*12);
worka2 = PartPlannng("Squat", sq, increase*14) + PartPlannng("Bench Press", bp, increase*7) + PartPlannng("Pendlay Row", pr, increase*7);
String[] result5 = new String[] {worka1, "Rest and \nrecovery", workb, "Rest and \nrecovery", worka2};
workb1 = PartPlannng("Squat", sq, increase*15) + PartPlannng("Overhead Press", ohp, increase*7) + PartPlannng("Deadlift", dl, increase*14);
worka = PartPlannng("Squat", sq, increase*16) + PartPlannng("Bench Press", bp, increase*8) + PartPlannng("Pendlay Row", pr, increase*8);
workb2 = PartPlannng("Squat", sq, increase*17) + PartPlannng("Overhead Press", ohp, increase*8) + PartPlannng("Deadlift", dl, increase*16);
String[] result6 = new String[] {workb1, "Rest and \nrecovery", worka, "Rest and \nrecovery", workb2};
worka1 = PartPlannng("Squat", sq, increase*18) + PartPlannng("Bench Press", bp, increase*9) + PartPlannng("Pendlay Row", pr, increase*9);
workb = PartPlannng("Squat", sq, increase*19) + PartPlannng("Overhead Press", ohp, increase*9) + PartPlannng("Deadlift", dl, increase*18);
worka2 = PartPlannng("Squat", sq, increase*20) + PartPlannng("Bench Press", bp, increase*10) + PartPlannng("Pendlay Row", pr, increase*10);
String[] result7 = new String[] {worka1, "Rest and \nrecovery", workb, "Rest and \nrecovery", worka2};
workb1 = PartPlannng("Squat", sq, increase*21) + PartPlannng("Overhead Press", ohp, increase*10) + PartPlannng("Deadlift", dl, increase*20);
worka = PartPlannng("Squat", sq, increase*22) + PartPlannng("Bench Press", bp, increase*11) + PartPlannng("Pendlay Row", pr, increase*11);
workb2 = PartPlannng("Squat", sq, increase*23) + PartPlannng("Overhead Press", ohp, increase*11) + PartPlannng("Deadlift", dl, increase*22);
String[] result8 = new String[] {workb1, "Rest and \nrecovery", worka, "Rest and \nrecovery", workb2};
worka1 = PartPlannng("Squat", sq, increase*24) + PartPlannng("Bench Press", bp, increase*12) + PartPlannng("Pendlay Row", pr, increase*12);
workb = PartPlannng("Squat", sq, increase*25) + PartPlannng("Overhead Press", ohp, increase*12) + PartPlannng("Deadlift", dl, increase*24);
worka2 = PartPlannng("Squat", sq, increase*26) + PartPlannng("Bench Press", bp, increase*13) + PartPlannng("Pendlay Row", pr, increase*13);
String[] result9 = new String[] {worka1, "Rest and \nrecovery", workb, "Rest and \nrecovery", worka2};
workb1 = PartPlannng("Squat", sq, increase*27) + PartPlannng("Overhead Press", ohp, increase*13) + PartPlannng("Deadlift", dl, increase*26);
worka = PartPlannng("Squat", sq, increase*28) + PartPlannng("Bench Press", bp, increase*14) + PartPlannng("Pendlay Row", pr, increase*14);
workb2 = PartPlannng("Squat", sq, increase*29) + PartPlannng("Overhead Press", ohp, increase*14) + PartPlannng("Deadlift", dl, increase*28);
String[] result10 = new String[] {workb1, "Rest and \nrecovery", worka, "Rest and \nrecovery", workb2};
worka1 = PartPlannng("Squat", sq, increase*30) + PartPlannng("Bench Press", bp, increase*15) + PartPlannng("Pendlay Row", pr, increase*15);
workb = PartPlannng("Squat", sq, increase*31) + PartPlannng("Overhead Press", ohp, increase*15) + PartPlannng("Deadlift", dl, increase*30);
worka2 = PartPlannng("Squat", sq, increase*32) + PartPlannng("Bench Press", bp, increase*16) + PartPlannng("Pendlay Row", pr, increase*16);
String[] result11 = new String[] {worka1, "Rest and \nrecovery", workb, "Rest and \nrecovery", worka2};
workb1 = PartPlannng("Squat", sq, increase*33) + PartPlannng("Overhead Press", ohp, increase*16) + PartPlannng("Deadlift", dl, increase*32);
worka = PartPlannng("Squat", sq, increase*34) + PartPlannng("Bench Press", bp, increase*17) + PartPlannng("Pendlay Row", pr, increase*17);
workb2 = PartPlannng("Squat", sq, increase*35) + PartPlannng("Overhead Press", ohp, increase*17) + PartPlannng("Deadlift", dl, increase*34);
String[] result12 = new String[] {workb1, "Rest and \nrecovery", worka, "Rest and \nrecovery", workb2};
monthSet.add(result1);monthSet.add(result2);monthSet.add(result3);monthSet.add(result4);
monthSet.add(result5);monthSet.add(result6);monthSet.add(result7);monthSet.add(result8);
monthSet.add(result9);monthSet.add(result10);monthSet.add(result11);monthSet.add(result12);
return monthSet;
}
}
```



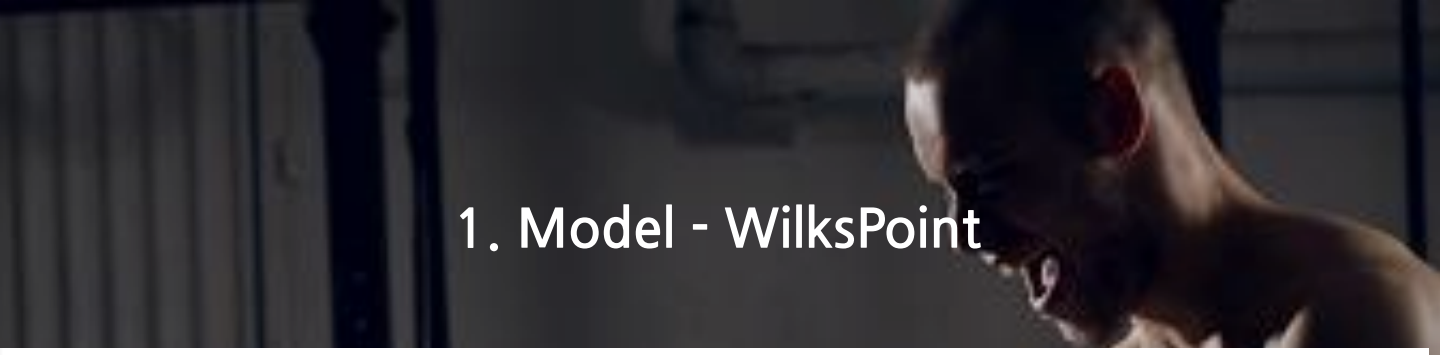
1. Model - Wendler

```
public class Wendler {
    private double[] intensityCalc(double rmWeight) {       // use to intensityForWaves
        RepMaxCalc rmc = new RepMaxCalc();
        double[] tmp = {0, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.4, 0.5, 0.6};
        double[] result = new double[tmp.length];
        for(int i = 0 ; i < result.length ; i++) {
            result[i] = (rmc.Mround(((rmWeight*0.9) * tmp[i]), 2.5));
        }
        return result;
    }
    private String[] waveA(Member member, double increase) {
        double sq = member.getSq1rm();
        double bp = member.getBp1rm();
        double dl = member.getDl1rm();
        double ohp = member.getOhp1rm();
        double[] tmp = intensityCalc(sq);
        String sq1 = "Squat\n"+(tmp[1]+increase*2) + "kg X5\n"
                + (tmp[3]+increase*2) + "kg X5\n"
                + (tmp[5]+increase*2) + "kg X5+\n";
        tmp = intensityCalc(bp);
        String bp1 = "Bench Press\n"+(tmp[1]+increase) + "kg X5\n"
                + (tmp[3]+increase) + "kg X5\n"
                + (tmp[5]+increase) + "kg X5+\n";
        tmp = intensityCalc(dl);
        String dl1 = "Deadlift\n"+(tmp[1]+increase) + "kg X5\n"
                + (tmp[3]+increase) + "kg X5\n"
                + (tmp[5]+increase) + "kg X5+\n";
        tmp = intensityCalc(ohp);
        String ohp1 = "Overhead Press\n"+(tmp[1]+increase) + "kg X5\n"
                + (tmp[3]+increase) + "kg X5\n"
                + (tmp[5]+increase) + "kg X5+\n";
        String[] result = {sq1,ohp1,"Rest and \nrecovery",dl1,bp1};
        return result;
    }
    private String[] waveB(Member member, double increase) {
    private String[] waveC(Member member, double increase) {
    private String[] waveD(Member member, double increase) {
```



1. Model - Wendler

```java
public ArrayList<String[]> createRoutine(Member member, double increase) {
    ArrayList<String[]> monthSet = new ArrayList<String[]>();
    String[] waveA, waveB, waveC, waveD;
    for(int i = 0 ; i < 3 ; i++) {
        waveA = waveA(member, increase*i);
        waveB = waveB(member, increase*i);
        waveC = waveC(member, increase*i);
        waveD = waveD(member, increase*i);
        monthSet.add(waveA);
        monthSet.add(waveB);
        monthSet.add(waveC);
        monthSet.add(waveD);
    }
    return monthSet;
}
```



1. Model - WilksPoint

```java
public class WilksPoint {
    private double maleA = -216.0475144;
    private double maleB = 16.2606339;
    private double maleC = -0.002388645;
    private double maleD = -0.00113732;
    private double maleE = 7.01863E-06;
    private double maleF = -1.291E-08;
    private double femaleA = 594.31747775582;
    private double femaleB = -27.23842536447;
    private double femaleC = 0.82112226871;
    private double femaleD = -0.00930733913;
    private double femaleE = 4.731582E-05;
    private double femaleF = -9.054E-08;
    public double Coeff(Member member){
        DecimalFormat df = new DecimalFormat("0.##");
        double x = member.getWeight();
        double total = member.getBp1rm() + member.getDl1rm() + member.getSq1rm();
        if(member.getGender().equals("male")) {
            String tmp = df.format(
                total / (maleA + (maleB*x) + (maleC*x*x) + (maleD*x*x*x) +
                        (maleE*x*x*x*x) + (maleF*x*x*x*x*x*x))/2*1000
            );
            double result = Double.parseDouble(tmp);
            return result;
        }
        else {
            String tmp = df.format(
                total / (femaleA + (femaleB*x) + (femaleC*x*x) + (femaleD*x*x*x) +
                        (femaleE*x*x*x*x) + (femaleF*x*x*x*x*x*x))/2*1000
            );
            double result = Double.parseDouble(tmp);
            return result;
        }
    }
}
```

1. Model - WilksRank

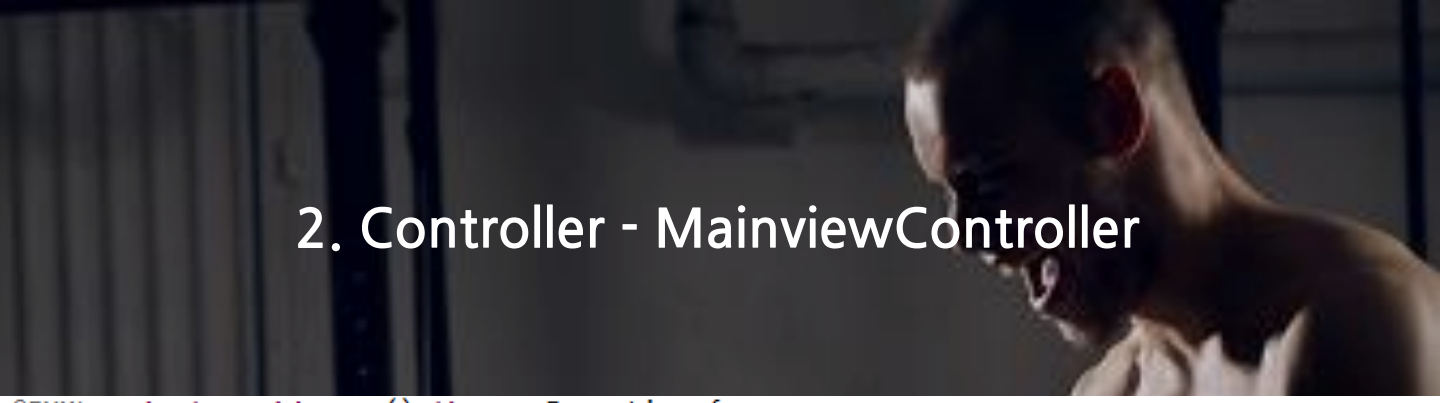
```
public class WilksRank {
    WilksPoint wp = new WilksPoint();
    final private double[][] rankArr = new double [][]{
            {52,116,193,227,321,416},
            {56,116,193,230,320,415},
            {60,117,195,231,321,414},
            {67,118,197,236,326,416},
            {75,119,198,236,326,415},
            {82,120,201,239,329,418},
            {90,121,201,241,329,416},
            {100,121,202,243,330,415},
            {110,123,204,242,329,412},
            {125,122,203,241,326,408},
            {145,121,202,240,324,405},
            {146,124,206,245,330,413}
    //  {WeightClass, Beginner, Novice, Intermediate, Advanced, Elite}
    };
    private double[] rankCatch(double weight) {
        double [] result = rankArr[0];
        for(int i = 0 ; i < rankArr.length ; i++) {
            if(weight >= rankArr[i][0]) result = rankArr[i];
        }
        return result;
    }
    public String weightClass(Member member) {
        String result = "52+";
        for(int i = 0 ; i < rankArr.length ; i++) {
            if(member.getWeight() >= rankArr[i][0]) result = rankArr[i][0] + "+";
        }
        return result;
    }
    public String getRank(Member member) {                    // 수정필요
        double wilkspoint = wp.Coeff(member);
        double [] standard = rankCatch(member.getWeight());
        String rank = "Beginner";
        if(wilkspoint > standard[2]) rank = "Novice";
        if(wilkspoint > standard[3]) rank = "Intermediate";
        if(wilkspoint > standard[4]) rank = "Advanced";
        if(wilkspoint > standard[5]) rank = "Elite";
        return rank;
    }
}
```

2. Controller - MainviewController

```
@Override
public void initialize(URL location, ResourceBundle resources) {
    if (fio.oblist == null) {
        fio.list = new ArrayList<Member>();
        fio.oblist = FXCollections.observableArrayList();
    }
    try {
        fio.loaddata();
        fio.initModel();
    } catch (Exception e) {}
```

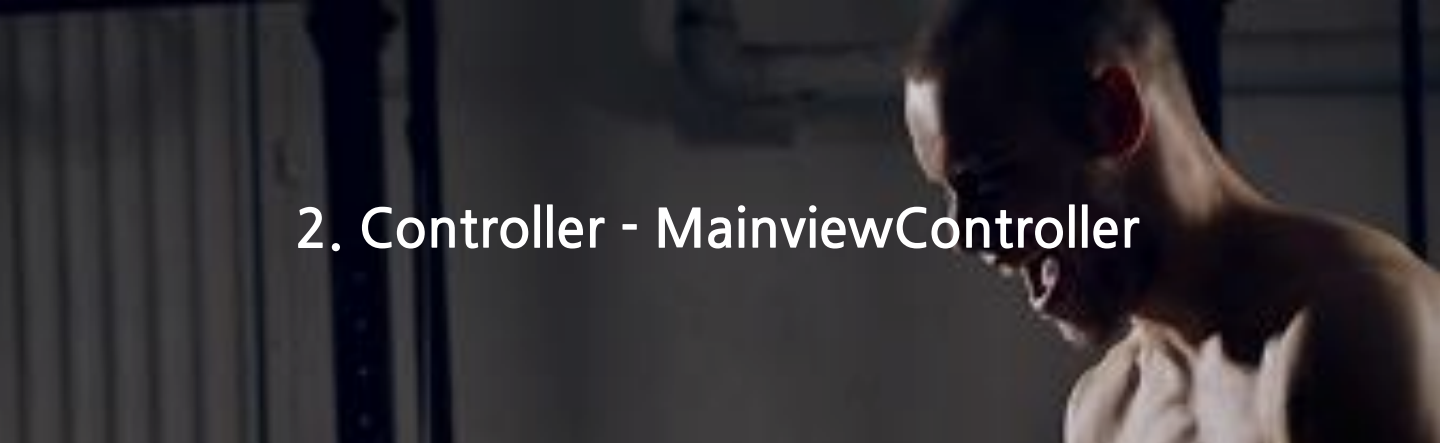
```
tableColumn.setCellValueFactory(v->v.getValue().getNameSSP());
table.setItems(fio.oblist);
table.getSelectionModel().selectedItemProperty().addListener((ob, ov, nv) -> {
    member = nv;        //      for Member Edit and Workout Scheduling
```

```
//editField init
Bindings.bindBidirectional(editName.textProperty(), name.textProperty());
```



2. Controller - MainviewController

```
@FXML private void open() throws Exception {
    try {
        FileChooser fc = new FileChooser();
        File file = fc.showOpenDialog(new Stage());
        fio.loaddata(file);
        filePath = file.getAbsolutePath().replace("\\", "\\\\");
        fio.initModel();
        fio.savedata();
        Stage now = (Stage) table.getScene().getWindow();
        FXMLLoader loader = new FXMLLoader(getClass().getResource("mainview.fxml"));
        now.setResizable(false);
        now.setScene(new Scene(loader.load()));
    } catch (Exception e) {}
}
@FXML private void save() throws Exception {
    FileChooser fc = new FileChooser();
    File file = fc.showSaveDialog(new Stage());
    fio.savedata(file);
}
@FXML private void quit() throws Exception {    System.exit(0); }
@FXML private void delbtn() {
    try {
        Alert alert = new Alert(AlertType.CONFIRMATION);
        alert.setTitle("Delete Member");
        alert.setHeaderText("Delete this Member : "+member.getName());
        alert.setContentText("Are you sure you want to delete this member?");
        Optional<ButtonType> result = alert.showAndWait();
        if (result.get() == ButtonType.OK){
            fio.list.remove(table.getSelectionModel().getSelectedIndex());
            fio.savedata();
            Stage now = (Stage) table.getScene().getWindow();
            FXMLLoader loader = new FXMLLoader(getClass().getResource("mainview.fxml"));
            now.setResizable(false);
            now.setScene(new Scene(loader.load()));
        } else {}
    } catch (Exception e) {}
}
```



2. Controller - MainviewController

```
@FXML private void editbtn() {
    setHideandShow("H", tableBtnStack.getChildren().get(0));
    setHideandShow("S", tableBtnStack.getChildren().get(1),
            infoStack.getChildren().get(1), strStack.getChildren().get(1));
}
@FXML private void modificationbtn() {
    modification();
}
private void modification() {
    try {
        member.setName(editName.getText());
        member.setAge(editAge.getText());
        member.setGender(editGender.getText());
        member.setAddress(editAddress.getText());
        member.setEmail(editEmail.getText());
        member.setPhonenumber(editPhonenumber.getText());
        member.setWeight(Double.parseDouble(editbodyweight.getText()));
        member.setBp1rm(Double.parseDouble(editbp.getText()));
        member.setSq1rm(Double.parseDouble(editsq.getText()));
        member.setDl1rm(Double.parseDouble(editdl.getText()));
        member.setOhp1rm(Double.parseDouble(editohp.getText()));
        member.setPr1rm(Double.parseDouble(editpr.getText()));
        setHideandShow("H", tableBtnStack.getChildren().get(0));
        setHideandShow("S", tableBtnStack.getChildren().get(1),
                infoStack.getChildren().get(1), strStack.getChildren().get(1));
        fio.savedata();
        Stage now = (Stage) table.getScene().getWindow();
        FXMLLoader loader = new FXMLLoader(getClass().getResource("mainview.fxml"));
        now.setResizable(false);
        now.setScene(new Scene(loader.load()));
    } catch (Exception e) {}
}
```

2. Controller - MainviewController

```
@FXML private void newbtn() {
    try {
        Stage now = (Stage) ap.getScene().getWindow();
        now.close();
        FXMLLoader loader = new FXMLLoader(getClass().getResource("joinview.fxml"));
        Parent root = loader.load();
        JoinviewController jcon = loader.getController();
        jcon.maincon = this;
        Stage stage = new Stage();
        stage.setScene(new Scene(root));
        stage.showAndWait();
        FXMLLoader loader2 = new FXMLLoader(getClass().getResource("mainview.fxml"));
        Stage next = new Stage();
        next.setScene(new Scene(loader2.load()));
        next.show();
    } catch (Exception e) {}
}
@FXML private void onermbtn() {
    try {
        ap.setDisable(true);
        Stage newStage = new Stage();
        newStage.setAlwaysOnTop(true);
        Parent root = FXMLLoader.load(getClass().getResource("calcview.fxml"));
        newStage.setScene(new Scene(root));
        newStage.showAndWait();
        ap.setDisable(false);
    } catch (Exception e) {}
}
@FXML private void routinebtn() throws IOException {
    routinebtn.setDisable(true);
    ap.setDisable(true);
    FXMLLoader loader = new FXMLLoader(getClass().getResource("routineview.fxml"));
    Parent root = loader.load();
    RoutineviewController rvc = loader.getController();
    rvc.setMaincon(this);
    Stage stage = new Stage();
    stage.setAlwaysOnTop(true);
    stage.setScene(new Scene(root));
    stage.showAndWait();
    routinebtn.setDisable(false);
    Stage now = (Stage) ap.getScene().getWindow();
    Parent root2 = FXMLLoader.load(getClass().getResource("mainview.fxml"));
    now.setScene(new Scene(root2));
}
```

2. Controller - MainviewController

```
@FXML private void scheduleRemovebtn() {
    member.setStrongliftRoutine(null);
    member.setWendlerRoutine(null);
    fio.savedata();
    try {
        Stage now = (Stage) table.getScene().getWindow();
        FXMLLoader loader = new FXMLLoader(getClass().getResource("mainview.fxml"));
        now.setResizable(false);
        now.setScene(new Scene(loader.load()));
    } catch (Exception e) {}
}
```

2. Controller - MainviewController

```
private void routineDisplay(ArrayList<String[]> routine) {
    month1.getChildren().clear();
    month2.getChildren().clear();
    month3.getChildren().clear();
    String[] day = {"Mon", "Tues", "Wed", "Thur", "Fri"};
    for(int i = 0 ; i < routine.size() ; i++) {
        for(int j = 0 ; j < routine.get(i).length ; j++) {
            if(i > 7) {
                month3.add(new Label(routine.get(i)[j]), j+1, i-7);
                Label label = new Label("Week"+(i+1));
                label.setPrefSize(100, 100);
                label.setFont(new Font(20));
                label.setAlignment(Pos.CENTER);
                label.setTextAlignment(TextAlignment.CENTER);
                month3.add(label, 0, i-7);
                Label label2 = new Label(day[j]);
                label2.setPrefSize(100, 100);
                label2.setFont(new Font(24));
                label2.setAlignment(Pos.CENTER);
                month3.add(label2, j+1, 0);
            }
            else if(i > 3) {
                month2.add(new Label(routine.get(i)[j]), j+1, i-3);
                Label label = new Label("Week"+(i+1));
                label.setPrefSize(100, 100);
                label.setFont(new Font(20));
                label.setAlignment(Pos.CENTER);
                label.setTextAlignment(TextAlignment.CENTER);
                month2.add(label, 0, i-3);
                Label label2 = new Label(day[j]);
                label2.setPrefSize(100, 100);
                label2.setFont(new Font(24));
                label2.setAlignment(Pos.CENTER);
                month2.add(label2, j+1, 0);
            }
            else {
                month1.add(new Label(routine.get(i)[j]), j+1, i+1);
                Label label = new Label("Week"+(i+1));
                label.setPrefSize(100, 100);
                label.setFont(new Font(20));
                label.setAlignment(Pos.CENTER);
                label.setTextAlignment(TextAlignment.CENTER);
                month1.add(label, 0, i+1);
                Label label2 = new Label(day[j]);
                label2.setPrefSize(100, 100);
                label2.setFont(new Font(24));
                label2.setAlignment(Pos.CENTER);
                month1.add(label2, j+1, 0);
            }
        }
    }
}
```

2. Controller - MainviewController

```
private void setHideandShow(String hs, Node...no) {
    if(hs.equals("H")) {
        for(Node n : no) {
            n.setOpacity(0);
            n.setDisable(true);
        }
    }
    else if(hs.equals("S")) {
        for(Node n : no) {
            n.setOpacity(1);
            n.setDisable(false);
        }
    }
}
private EventHandler<KeyEvent> searchEvent = (value)->{
    if(value.getCode().equals(KeyCode.ESCAPE))  {
        TextField tf = (TextField) value.getTarget();
        tf.setText("");
    }
    if(value.getCode().equals(KeyCode.ENTER))   {
        String searchname = search.getText();
        for(int i = 0 ; i < table.getItems().size() ; i++) {
            if(table.getItems().get(i).getName().equals(searchname)) {
                table.getFocusModel().focus(i);
                table.requestFocus();
            }
        }
    }
};
private EventHandler<KeyEvent> editKeyEvent = (value)->{
    if(value.getCode().equals(KeyCode.ESCAPE))  {
        TextField tf = (TextField) value.getTarget();
        tf.setText("");
    }
    if(value.getCode().equals(KeyCode.ENTER))   {
        modification();
    }
};
```

2. Controller - JoinviewController

```java
@Override
public void initialize(URL location, ResourceBundle resources) {
    checkMale.setToggleGroup(group);
    checkFemale.setToggleGroup(group);
    newName.addEventFilter(KeyEvent.KEY_PRESSED, newKeyEvent);
    newAge.addEventFilter(KeyEvent.KEY_PRESSED, newKeyEvent);
    newAddress.addEventFilter(KeyEvent.KEY_PRESSED, newKeyEvent);
    newPhonenumber.addEventFilter(KeyEvent.KEY_PRESSED, newKeyEvent);
    newEmail.addEventFilter(KeyEvent.KEY_PRESSED, newKeyEvent);
}
@FXML private void create() {
    if (newName.getText().contentEquals("")) {
        nameWarning.setText("* Essential information");
        newName.requestFocus();
    }
    else if (newAge.getText().contentEquals("")) {
        ageWarning.setText("* Essential information");
        newAge.requestFocus();
    }
    else if (checkMale.isSelected()==false && checkFemale.isSelected()==false) {
        genderWarning.setText("* Essential information");
        checkMale.requestFocus();
    }
    else if (newAddress.getText().contentEquals("")) {
        addressWarning.setText("* Essential information");
        newAddress.requestFocus();
    }
    else if (newPhonenumber.getText().contentEquals("")) {
        phonenumberWarning.setText("* Essential information");
        newPhonenumber.requestFocus();
    }
    else if (newEmail.getText().contentEquals("")) {
        emailWarning.setText("* Essential information");
        newEmail.requestFocus();
    }
    else {
        try {
            String name = newName.getText();
            String age = newAge.getText();
            String gender = null;
            if(checkMale.isSelected()) {gender = "Male";}
            else {gender = "Female";}
            String address = newAddress.getText();
            String phonenumber = newPhonenumber.getText();
            String email = newEmail.getText();
            maincon.fio.list.add(new Member(name, age, gender, address, phonenumber, email, 0, 0, 0, 0, 0, 0));
            maincon.fio.savedata();
            Stage now = (Stage) create.getScene().getWindow();
            now.close();
        } catch (Exception e) {}
    }
}
```

2. Controller - JoinviewController

```
@FXML private void cancel() {
    try {
        Stage now = (Stage) create.getScene().getWindow();
        now.close();
    } catch (Exception e) {}
}
private EventHandler<KeyEvent> newKeyEvent = (value)->{
    if(value.getCode().equals(KeyCode.ESCAPE))  {
        TextField tf = (TextField) value.getTarget();
        tf.setText("");
    }
    if(value.getCode().equals(KeyCode.ENTER))   {
        create();
    }
};
```

2. Controller - CalcviewController

```
@Override
public void initialize(URL location, ResourceBundle resources) {
    RepMaxCalc rmc = new RepMaxCalc();
    weightSpinner.setValueFactory(weightlist);
    weightSpinner.setEditable(true);
    repsChoiceBox.setItems(repslist);
    repsChoiceBox.setValue(1);
    weightSpinner.valueProperty().addListener((obs, ov, nv)->{
        ArrayList<String[]> rm = rmc.rmCalc(nv, Double.parseDouble(repsChoiceBox.getValue()+""));
        if(repsChoiceBox.getValue()==1) {
            gp.getChildren().clear();
            for(int i = 0 ; i < rm.size() ; i++) {
                for(int j = 0 ; j < rm.get(i).length ; j++) {
                    if(i == 1 && j==0) {
                        gp.add(new Label("1RM"), j, i);
                    }else if(i == 1 && j!=0) {gp.add(new Label(nv+""), j, i);
                    }else{gp.add(new Label(rm.get(i)[j]), j, i);}
                }
            }
        } else {
            gp.getChildren().clear();
            for(int i = 0 ; i < rm.size() ; i++) {
                for(int j = 0 ; j < rm.get(i).length ; j++) {
                    gp.add(new Label(rm.get(i)[j]), j, i);
                }
            }
        }
    });
    repsChoiceBox.valueProperty().addListener((obs, ov, nv)->{
        ArrayList<String[]> rm = rmc.rmCalc(Double.parseDouble(weightSpinner.getValue()+""), nv);
        if(nv==1) {
            gp.getChildren().clear();
            for(int i = 0 ; i < rm.size() ; i++) {
                for(int j = 0 ; j < rm.get(i).length ; j++) {
                    if(i == 1 && j==0) {
                        gp.add(new Label("1RM"), j, i);
                    }else if(i == 1 && j!=0) {gp.add(new Label(weightSpinner.getValue()+""), j, i);
                    }else{gp.add(new Label(rm.get(i)[j]), j, i);}
                }
            }
        } else {
            gp.getChildren().clear();
            for(int i = 0 ; i < rm.size() ; i++) {
                for(int j = 0 ; j < rm.get(i).length ; j++) {
                    gp.add(new Label(rm.get(i)[j]), j, i);
                }
            }
        }
    });
}
```

2. Controller - RoutineviewController

```
public void initialize(URL location, ResourceBundle resources) {
    panelist.add(firstpane);
    panelist.add(secondpane);
    panelist.add(thirdpane);
}
@FXML void wendlerbtn() {
    Wendler wd = new Wendler();
    try {
        if(maincon.member.getWendlerRoutine()==null&&maincon.member.getStrongliftRoutine()==null) {
            maincon.member.setWendlerRoutine(wd.createRoutine(maincon.member, 2.5));
            maincon.fio.savedata();
        }
        Stage now = (Stage)wendlerbtn.getScene().getWindow();
        now.close();
    } catch (Exception e) {}
}
@FXML void strongliftbtn() {
    StrongLift sl = new StrongLift();
    try {
        if(maincon.member.getWendlerRoutine()==null&&maincon.member.getStrongliftRoutine()==null) {
            maincon.member.setStrongliftRoutine(sl.createRoutine(maincon.member, 2.5));
            maincon.fio.savedata();
        }
        Stage now = (Stage)strongliftbtn.getScene().getWindow();
        now.close();
    } catch (Exception e) {}
}
@FXML private void wendlerhyperlink() throws MalformedURLException, IOException, URISyntaxException {
    Desktop.getDesktop().browse(new URL("https://crossfitoutbreak.com/wendler-531-strength-program/").toURI());
}
@FXML private void stronglifthyperlink() throws MalformedURLException, IOException, URISyntaxException {
    Desktop.getDesktop().browse(new URL("https://www.transparentlabs.com/blogs/all/35395395-building-mass-with-5x5-training").toURI());
}
@FXML private void nextbtn() {
    if(stackintex != 2) {
        stackintex++;
        for(int i = 0 ; i < panelist.size() ; i++) {
            if(stackintex == i) {
                panelist.get(i).setOpacity(1);
            }
            else {
                panelist.get(i).setOpacity(0);
            }
        }
    }
}
@FXML private void prevbtn() {
    if(stackintex != 0) {
        stackintex--;
        for(int i = 0 ; i < panelist.size() ; i++) {
            if(stackintex == i) {
                panelist.get(i).setOpacity(1);
            }
            else {
                panelist.get(i).setOpacity(0);
            }
        }
    }
}
```

2. Controller - WebviewController

```java
public class WebviewController implements Initializable {
    @FXML WebView webview;
    WebEngine engine;
    @Override
    public void initialize(URL location, ResourceBundle resources) {
        this.engine  = webview.getEngine();
        engine.load("https://en.wikipedia.org/wiki/One-repetition_maximum");
    }
}
```