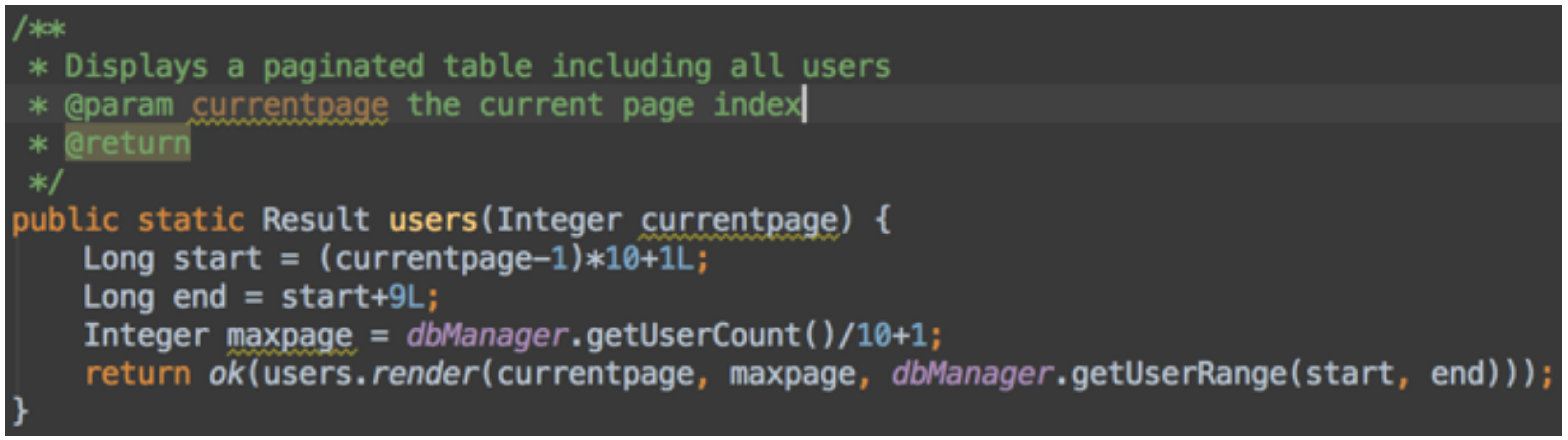
Refactoring

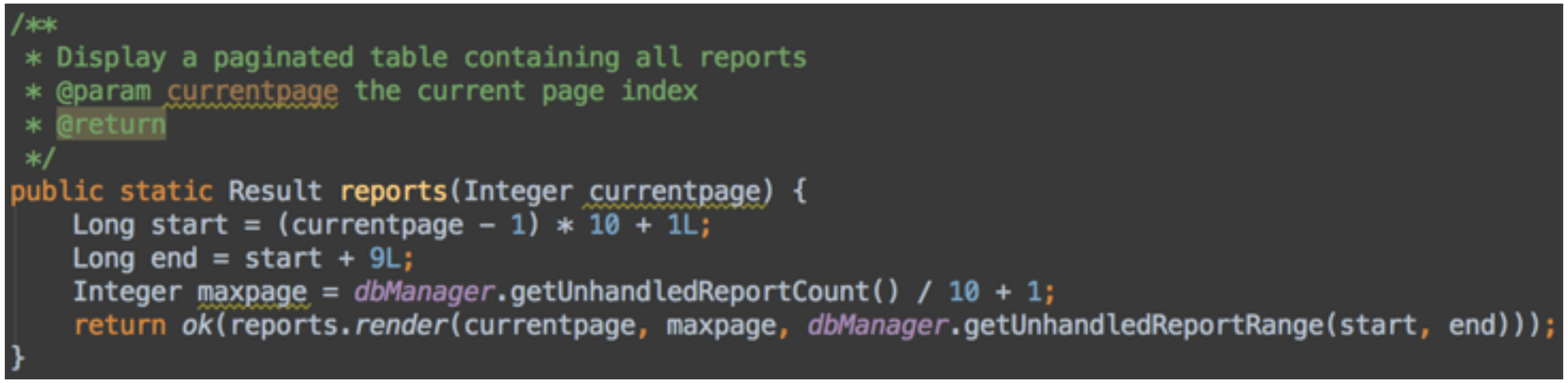
Where Am I

Fritz, Gerrier, Häusle, Zelger

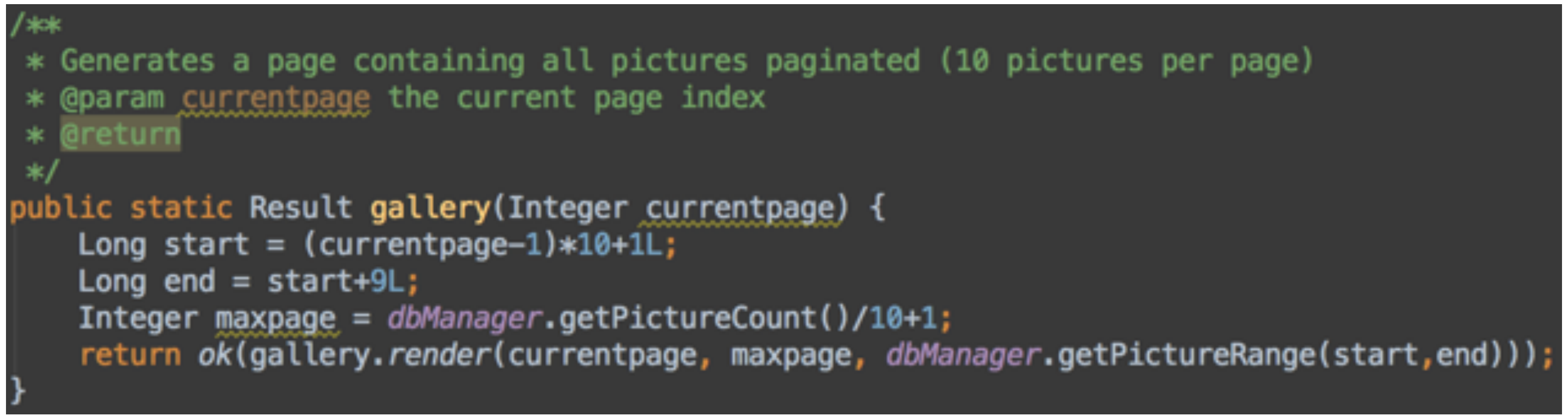
Duplicated code



```
/**
 * Displays a paginated table including all users
 * @param currentpage the current page index
 * @return
 */
public static Result users(Integer currentpage) {
    Long start = (currentpage-1)*10+1L;
    Long end = start+9L;
    Integer maxpage = dbManager.getUserCount()/10+1;
    return ok(users.render(currentpage, maxpage, dbManager.getUserRange(start, end)));
}
```

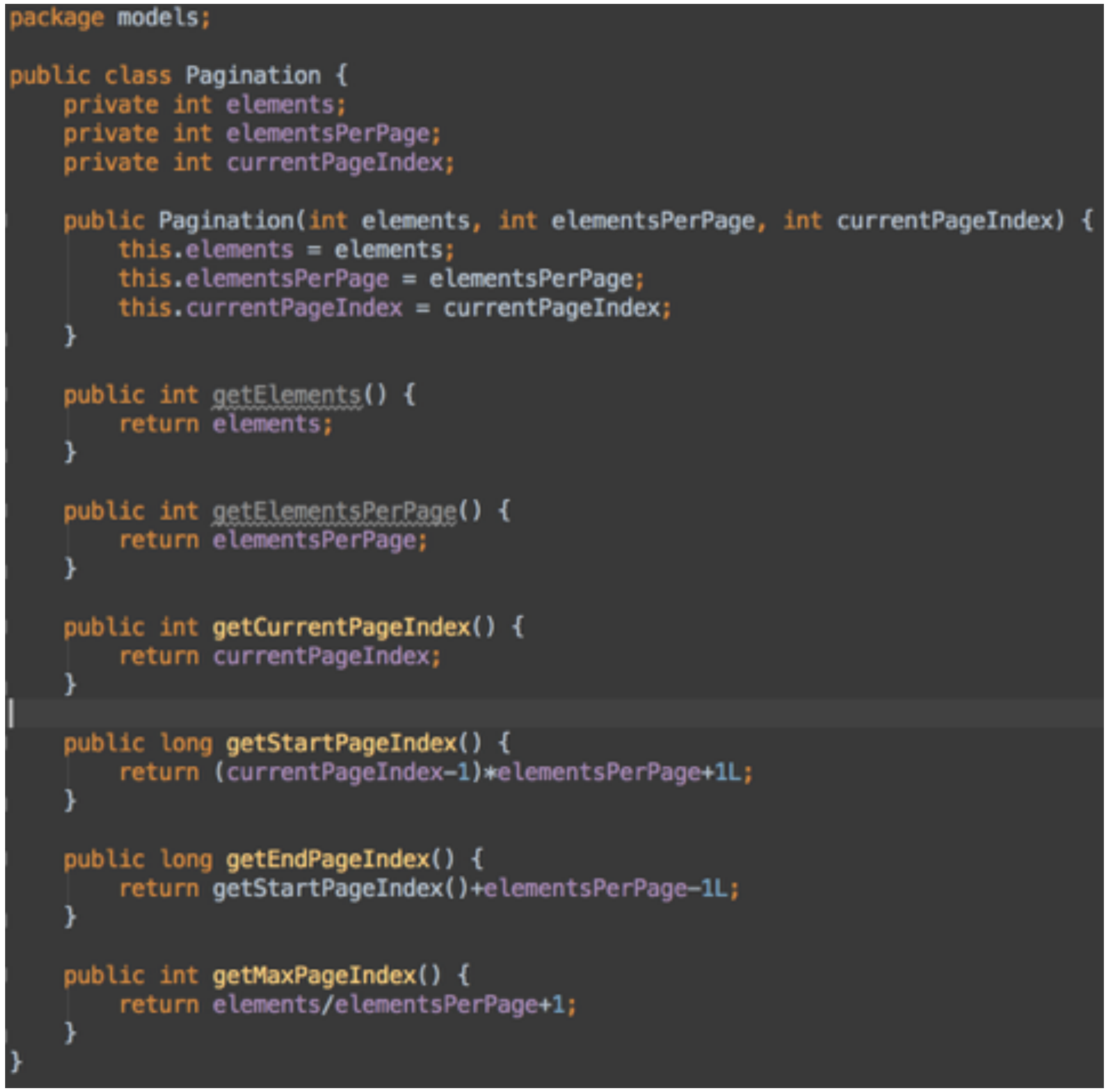


```
/**
 * Display a paginated table containing all reports
 * @param currentpage the current page index
 * @return
 */
public static Result reports(Integer currentpage) {
    Long start = (currentpage - 1) * 10 + 1L;
    Long end = start + 9L;
    Integer maxpage = dbManager.getUnhandledReportCount() / 10 + 1;
    return ok(reports.render(currentpage, maxpage, dbManager.getUnhandledReportRange(start, end)));
}
```



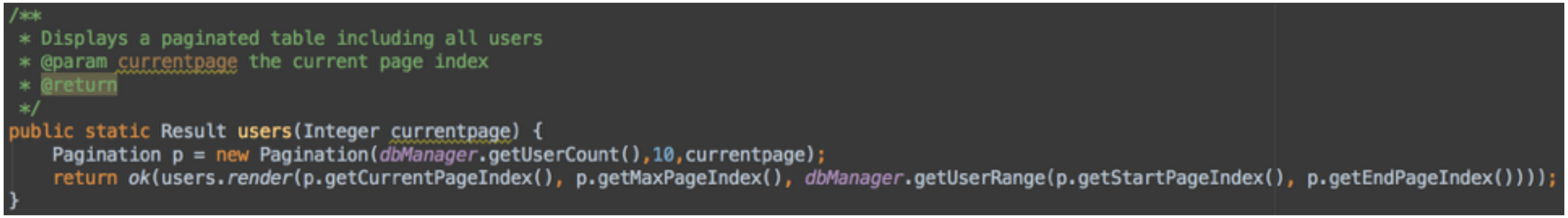
```
/**
 * Generates a page containing all pictures paginated (10 pictures per page)
 * @param currentpage the current page index
 * @return
 */
public static Result gallery(Integer currentpage) {
    Long start = (currentpage-1)*10+1L;
    Long end = start+9L;
    Integer maxpage = dbManager.getPictureCount()/10+1;
    return ok(gallery.render(currentpage, maxpage, dbManager.getPictureRange(start,end)));
}
```

Extract object (1)

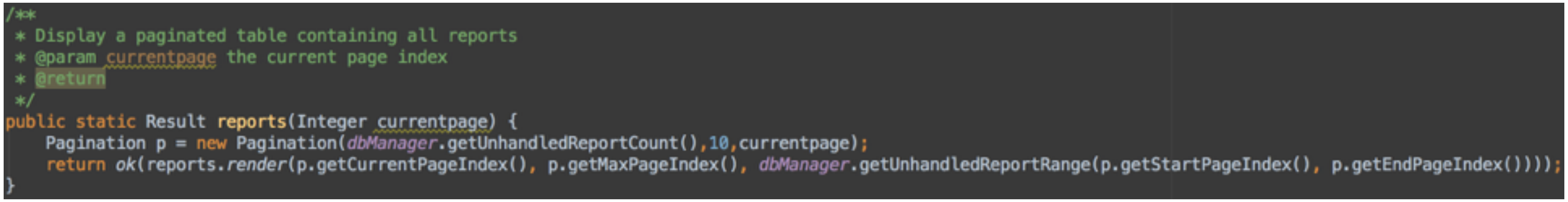


```java
package models;

public class Pagination {
    private int elements;
    private int elementsPerPage;
    private int currentPageIndex;

    public Pagination(int elements, int elementsPerPage, int currentPageIndex) {
        this.elements = elements;
        this.elementsPerPage = elementsPerPage;
        this.currentPageIndex = currentPageIndex;
    }

    public int getElements() {
        return elements;
    }

    public int getElementsPerPage() {
        return elementsPerPage;
    }

    public int getCurrentPageIndex() {
        return currentPageIndex;
    }

    public long getStartPageIndex() {
        return (currentPageIndex-1)*elementsPerPage+1L;
    }

    public long getEndPageIndex() {
        return getStartPageIndex()+elementsPerPage-1L;
    }

    public int getMaxPageIndex() {
        return elements/elementsPerPage+1;
    }
}
```

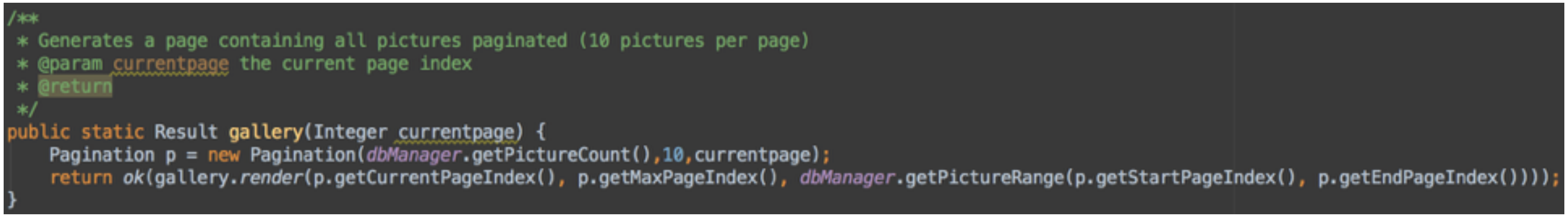
Extract object (2)



```
/**
 * Displays a paginated table including all users
 * @param currentpage the current page index
 * @return
 */
public static Result users(Integer currentpage) {
    Pagination p = new Pagination(dbManager.getUserCount(),10,currentpage);
    return ok(users.render(p.getCurrentPageIndex(), p.getMaxPageIndex(), dbManager.getUserRange(p.getStartPageIndex(), p.getEndPageIndex())));
}
```

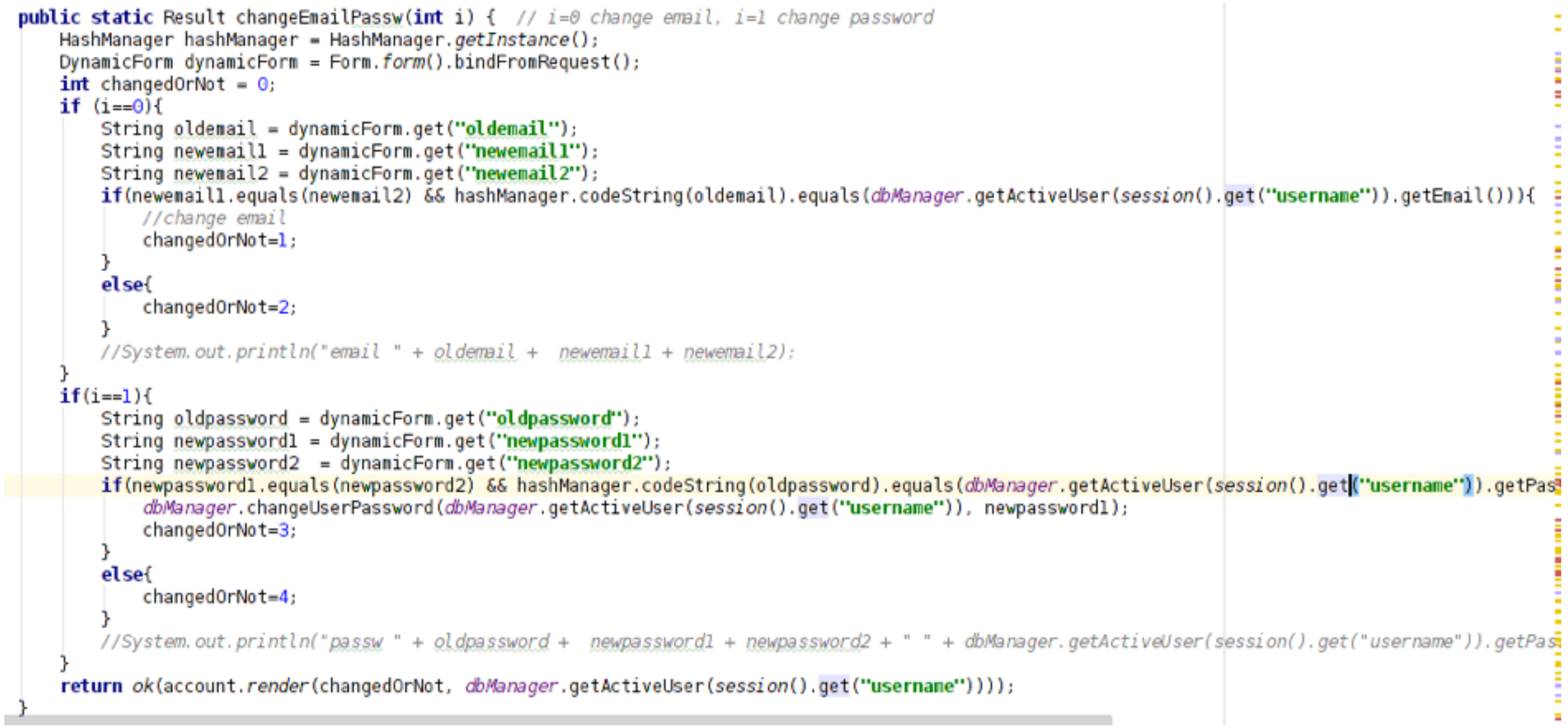


```
/**
 * Display a paginated table containing all reports
 * @param currentpage the current page index
 * @return
 */
public static Result reports(Integer currentpage) {
    Pagination p = new Pagination(dbManager.getUnhandledReportCount(),10,currentpage);
    return ok(reports.render(p.getCurrentPageIndex(), p.getMaxPageIndex(), dbManager.getUnhandledReportRange(p.getStartPageIndex(), p.getEndPageIndex())));
}
```



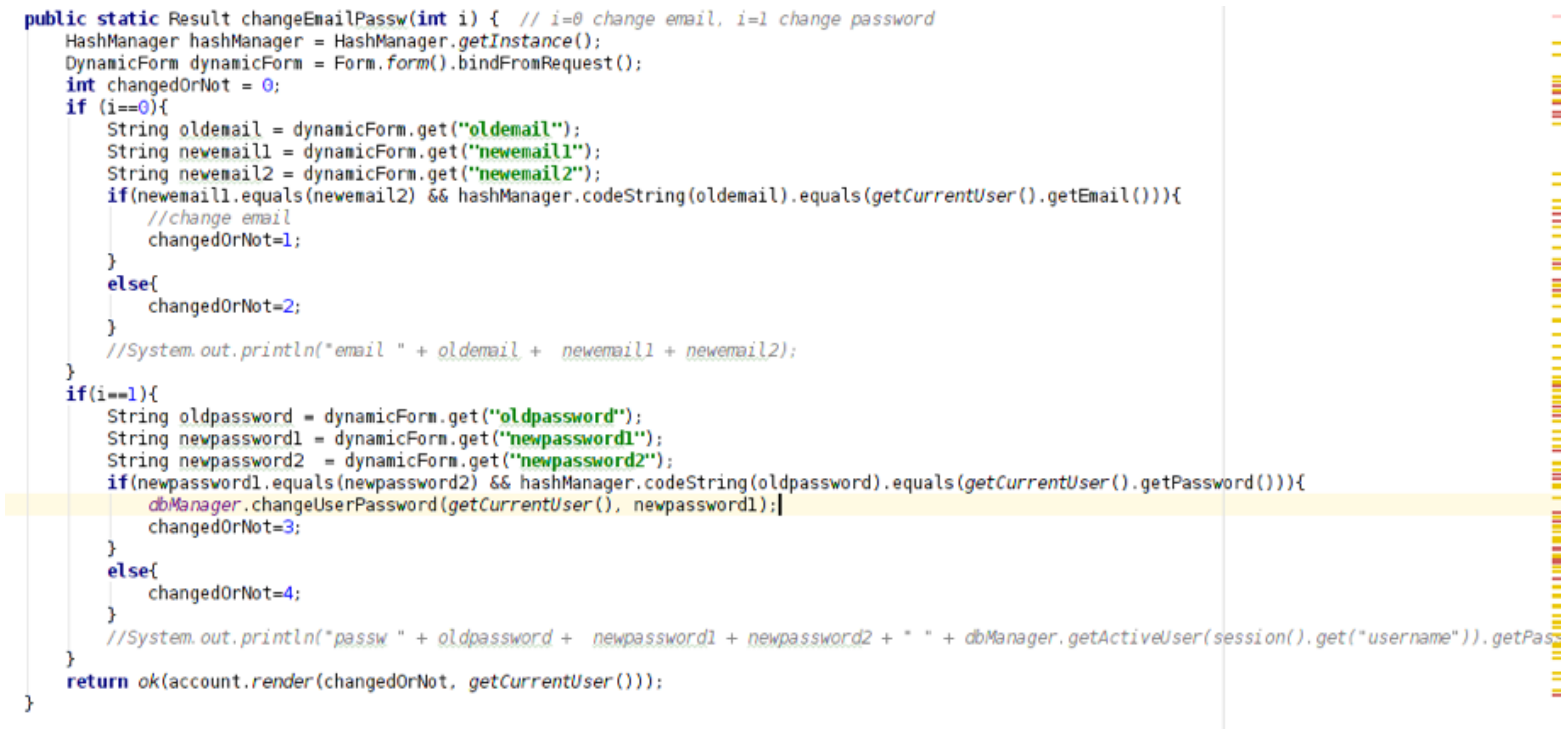
```
/**
 * Generates a page containing all pictures paginated (10 pictures per page)
 * @param currentpage the current page index
 * @return
 */
public static Result gallery(Integer currentpage) {
    Pagination p = new Pagination(dbManager.getPictureCount(),10,currentpage);
    return ok(gallery.render(p.getCurrentPageIndex(), p.getMaxPageIndex(), dbManager.getPictureRange(p.getStartPageIndex(), p.getEndPageIndex())));
}
```

Duplicated code

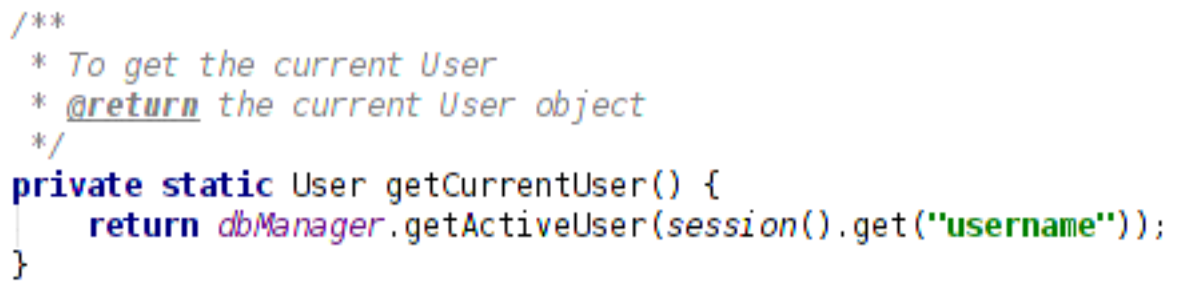


```java
public static Result changeEmailPassw(int i) {  // i=0 change email, i=1 change password
    HashManager hashManager = HashManager.getInstance();
    DynamicForm dynamicForm = Form.form().bindFromRequest();
    int changedOrNot = 0;
    if (i==0){
        String oldemail = dynamicForm.get("oldemail");
        String newemail1 = dynamicForm.get("newemail1");
        String newemail2 = dynamicForm.get("newemail2");
        if(newemail1.equals(newemail2) && hashManager.codeString(oldemail).equals(dbManager.getActiveUser(session().get("username")).getEmail())){
            //change email
            changedOrNot=1;
        }
        else{
            changedOrNot=2;
        }
        //System.out.println("email " + oldemail +  newemail1 + newemail2);
    }
    if(i==1){
        String oldpassword = dynamicForm.get("oldpassword");
        String newpassword1 = dynamicForm.get("newpassword1");
        String newpassword2  = dynamicForm.get("newpassword2");
        if(newpassword1.equals(newpassword2) && hashManager.codeString(oldpassword).equals(dbManager.getActiveUser(session().get("username")).getPas
            dbManager.changeUserPassword(dbManager.getActiveUser(session().get("username")), newpassword1);
            changedOrNot=3;
        }
        else{
            changedOrNot=4;
        }
        //System.out.println("passw " + oldpassword +  newpassword1 + newpassword2 + " " + dbManager.getActiveUser(session().get("username")).getPas
    }
    return ok(account.render(changedOrNot, dbManager.getActiveUser(session().get("username"))));
}
```

Extract method

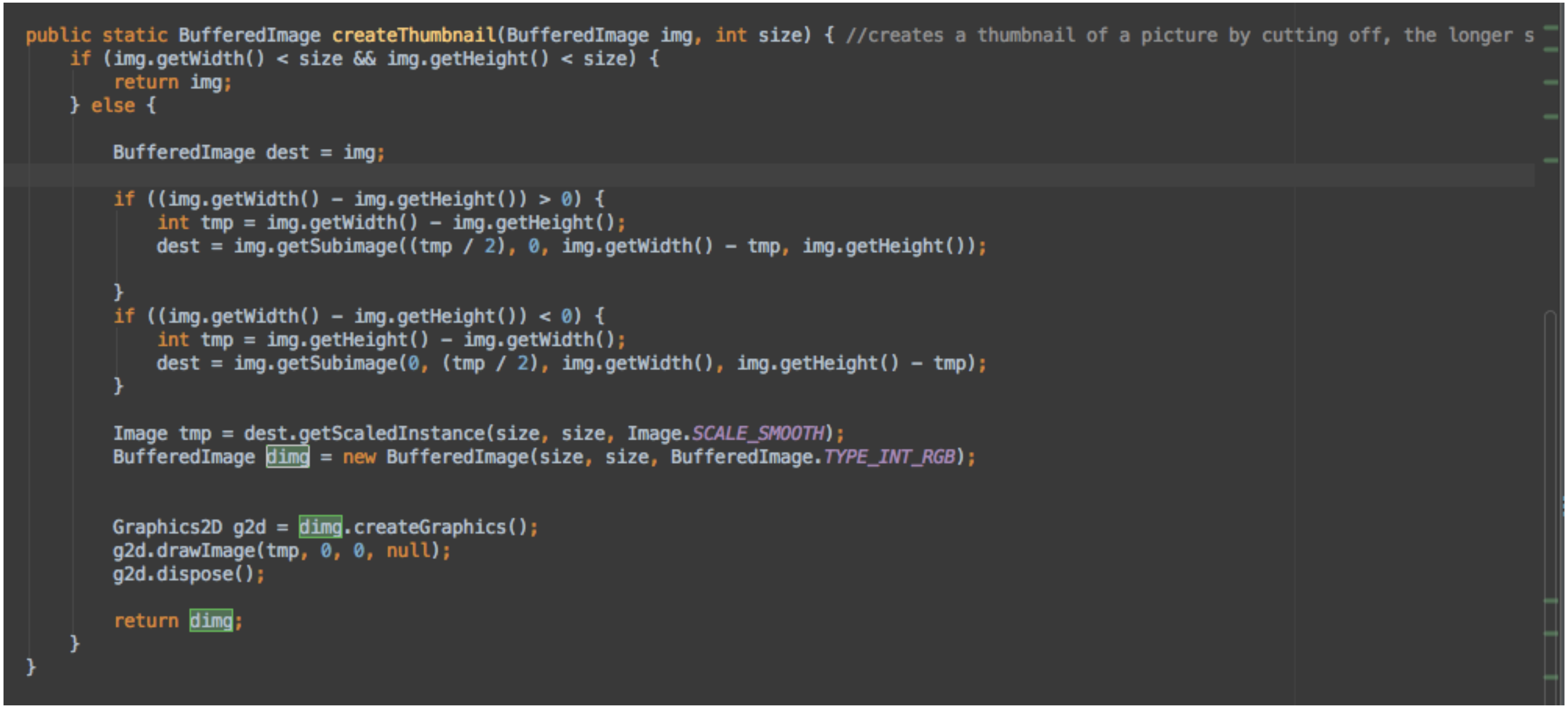


```java
public static Result changeEmailPassw(int i) {  // i=0 change email, i=1 change password
    HashManager hashManager = HashManager.getInstance();
    DynamicForm dynamicForm = Form.form().bindFromRequest();
    int changedOrNot = 0;
    if (i==0){
        String oldemail = dynamicForm.get("oldemail");
        String newemail1 = dynamicForm.get("newemail1");
        String newemail2 = dynamicForm.get("newemail2");
        if(newemail1.equals(newemail2) && hashManager.codeString(oldemail).equals(getCurrentUser().getEmail())){
            //change email
            changedOrNot=1;
        }
        else{
            changedOrNot=2;
        }
        //System.out.println("email " + oldemail +  newemail1 + newemail2);
    }
    if(i==1){
        String oldpassword = dynamicForm.get("oldpassword");
        String newpassword1 = dynamicForm.get("newpassword1");
        String newpassword2  = dynamicForm.get("newpassword2");
        if(newpassword1.equals(newpassword2) && hashManager.codeString(oldpassword).equals(getCurrentUser().getPassword())){
            dbManager.changeUserPassword(getCurrentUser(), newpassword1);
            changedOrNot=3;
        }
        else{
            changedOrNot=4;
        }
        //System.out.println("passw " + oldpassword +  newpassword1 + newpassword2 + " " + dbManager.getActiveUser(session().get("username")).getPas
    }
    return ok(account.render(changedOrNot, getCurrentUser()));
}
```



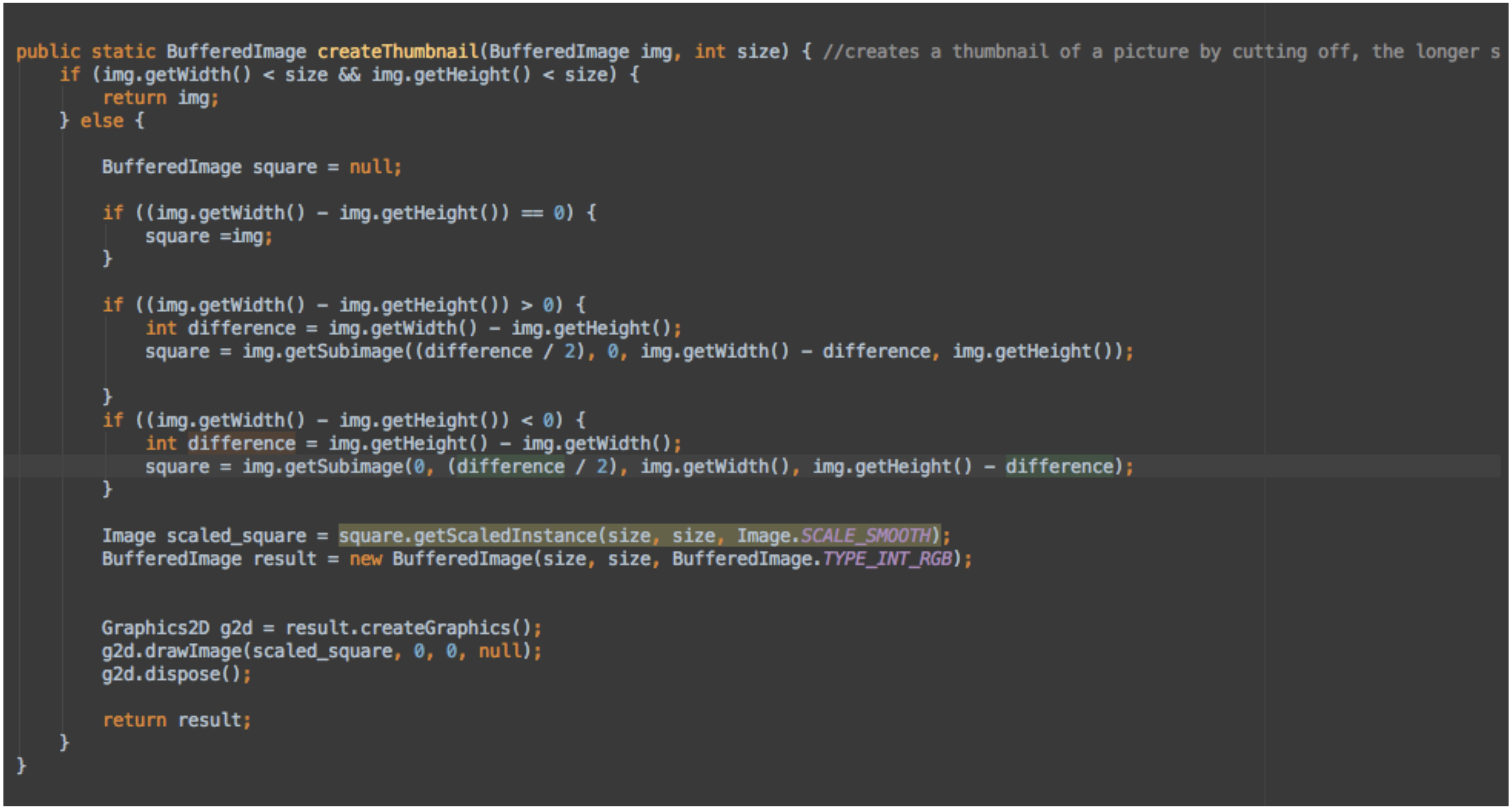
```java
/**
 * To get the current User
 * @return the current User object
 */
private static User getCurrentUser() {
    return dbManager.getActiveUser(session().get("username"));
}
```

Complicated code



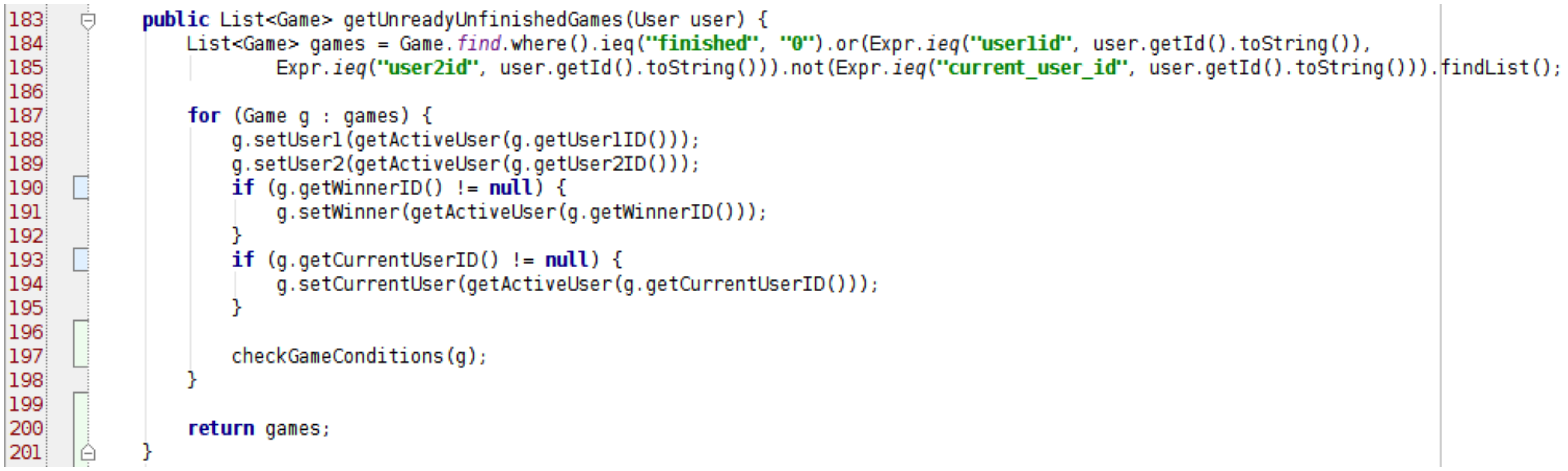
```java
public static BufferedImage createThumbnail(BufferedImage img, int size) { //creates a thumbnail of a picture by cutting off, the longer s
    if (img.getWidth() < size && img.getHeight() < size) {
        return img;
    } else {

        BufferedImage dest = img;

        if ((img.getWidth() - img.getHeight()) > 0) {
            int tmp = img.getWidth() - img.getHeight();
            dest = img.getSubimage((tmp / 2), 0, img.getWidth() - tmp, img.getHeight());

        }
        if ((img.getWidth() - img.getHeight()) < 0) {
            int tmp = img.getHeight() - img.getWidth();
            dest = img.getSubimage(0, (tmp / 2), img.getWidth(), img.getHeight() - tmp);
        }

        Image tmp = dest.getScaledInstance(size, size, Image.SCALE_SMOOTH);
        BufferedImage dimg = new BufferedImage(size, size, BufferedImage.TYPE_INT_RGB);


        Graphics2D g2d = dimg.createGraphics();
        g2d.drawImage(tmp, 0, 0, null);
        g2d.dispose();

        return dimg;
    }
}
```

Code simplification



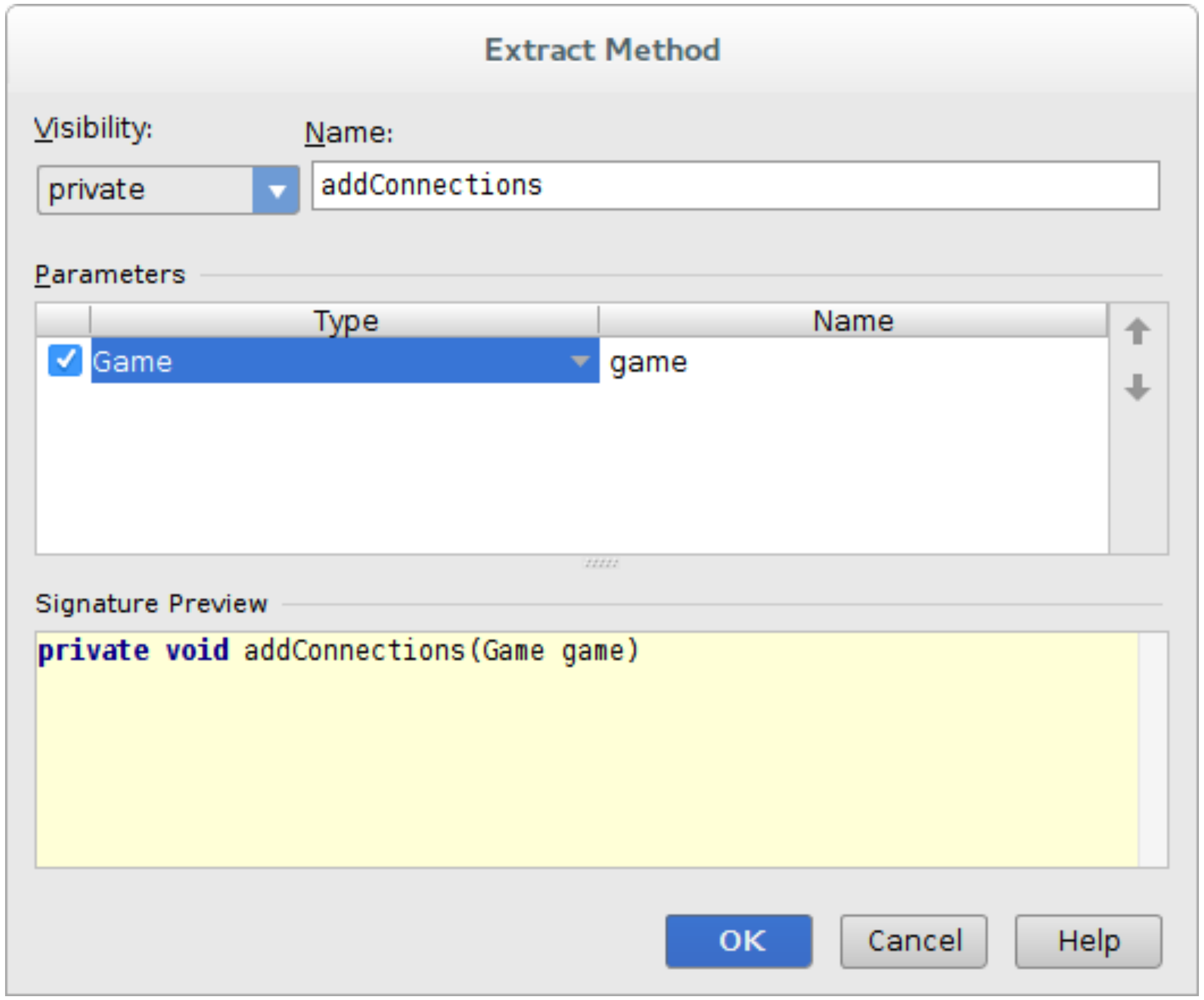
```
public static BufferedImage createThumbnail(BufferedImage img, int size) { //creates a thumbnail of a picture by cutting off, the longer s
    if (img.getWidth() < size && img.getHeight() < size) {
        return img;
    } else {

        BufferedImage square = null;

        if ((img.getWidth() - img.getHeight()) == 0) {
            square =img;
        }

        if ((img.getWidth() - img.getHeight()) > 0) {
            int difference = img.getWidth() - img.getHeight();
            square = img.getSubimage((difference / 2), 0, img.getWidth() - difference, img.getHeight());

        }
        if ((img.getWidth() - img.getHeight()) < 0) {
            int difference = img.getHeight() - img.getWidth();
            square = img.getSubimage(0, (difference / 2), img.getWidth(), img.getHeight() - difference);
        }

        Image scaled_square = square.getScaledInstance(size, size, Image.SCALE_SMOOTH);
        BufferedImage result = new BufferedImage(size, size, BufferedImage.TYPE_INT_RGB);


        Graphics2D g2d = result.createGraphics();
        g2d.drawImage(scaled_square, 0, 0, null);
        g2d.dispose();

        return result;
    }
}
```

Duplicated code



```java
public List<Game> getUnreadyUnfinishedGames(User user) {
    List<Game> games = Game.find.where().ieq("finished", "0").or(Expr.ieq("user1id", user.getId().toString()),
            Expr.ieq("user2id", user.getId().toString())).not(Expr.ieq("current_user_id", user.getId().toString())).findList();

    for (Game g : games) {
        g.setUser1(getActiveUser(g.getUser1ID()));
        g.setUser2(getActiveUser(g.getUser2ID()));
        if (g.getWinnerID() != null) {
            g.setWinner(getActiveUser(g.getWinnerID()));
        }
        if (g.getCurrentUserID() != null) {
            g.setCurrentUser(getActiveUser(g.getCurrentUserID()));
        }

        checkGameConditions(g);
    }

    return games;
}
```

Extract method (1)



Visibility: private

Name: addConnections

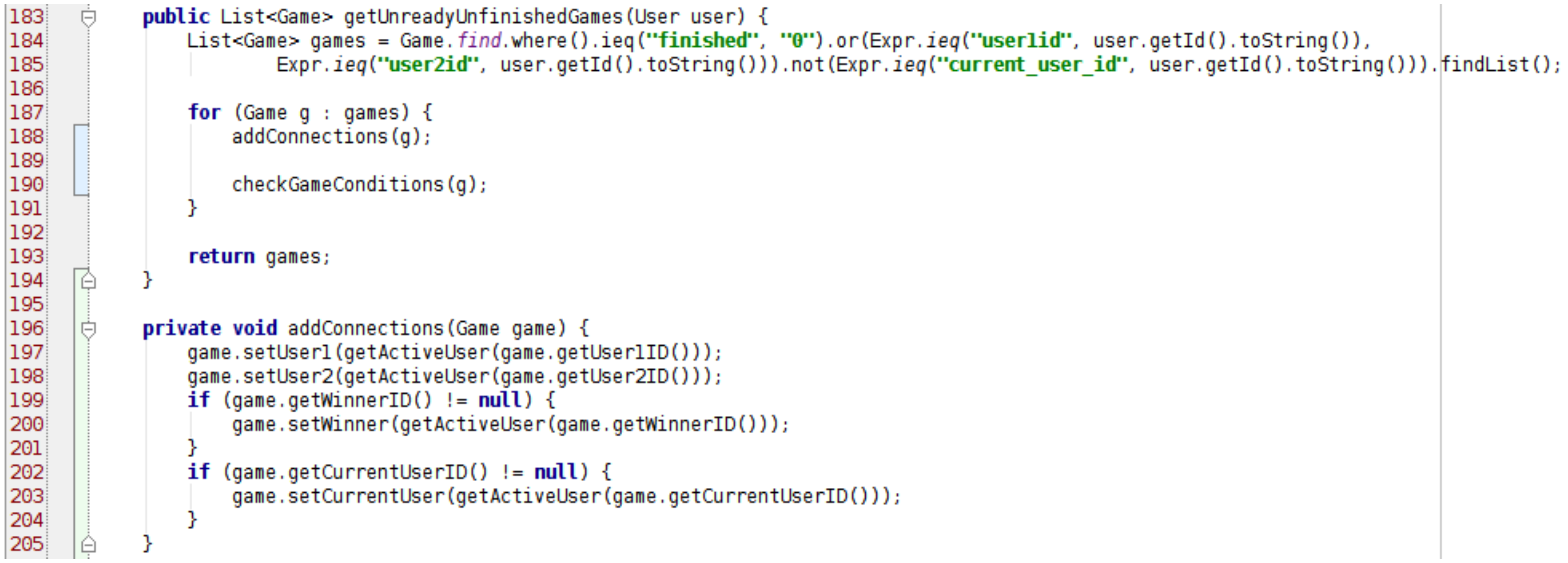
Parameters

	Type	Name
✓	Game	game

Signature Preview

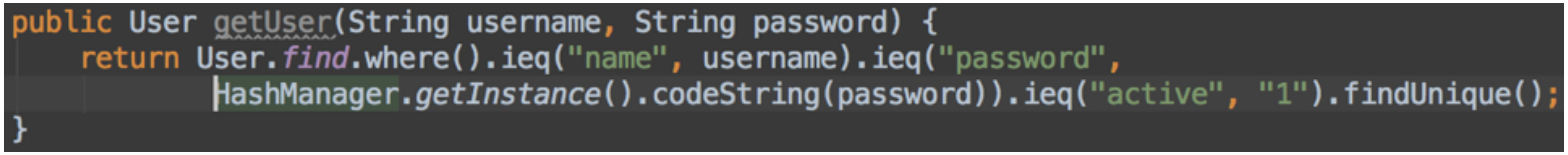
```
private void addConnections(Game game)
```

Extract method (2)

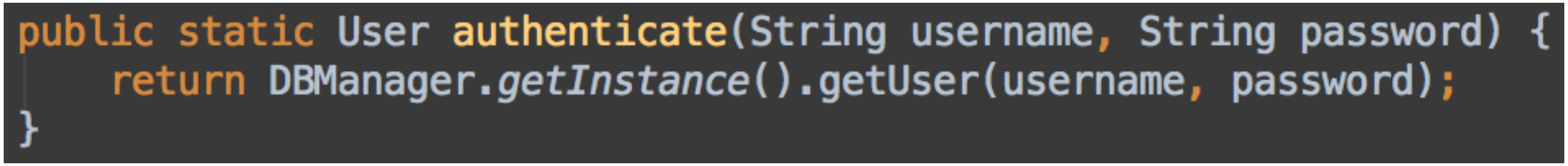


```java
public List<Game> getUnreadyUnfinishedGames(User user) {
    List<Game> games = Game.find.where().ieq("finished", "0").or(Expr.ieq("userlid", user.getId().toString()),
            Expr.ieq("user2id", user.getId().toString())).not(Expr.ieq("current_user_id", user.getId().toString())).findList();

    for (Game g : games) {
        addConnections(g);

        checkGameConditions(g);
    }

    return games;
}

private void addConnections(Game game) {
    game.setUser1(getActiveUser(game.getUser1ID()));
    game.setUser2(getActiveUser(game.getUser2ID()));
    if (game.getWinnerID() != null) {
        game.setWinner(getActiveUser(game.getWinnerID()));
    }
    if (game.getCurrentUserID() != null) {
        game.setCurrentUser(getActiveUser(game.getCurrentUserID()));
    }
}
```

Invalid method name

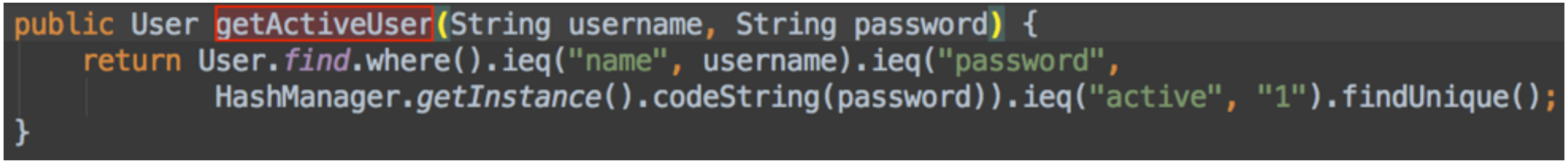


```java
public User getUser(String username, String password) {
    return User.find.where().ieq("name", username).ieq("password",
            HashManager.getInstance().codeString(password)).ieq("active", "1").findUnique();
}
```

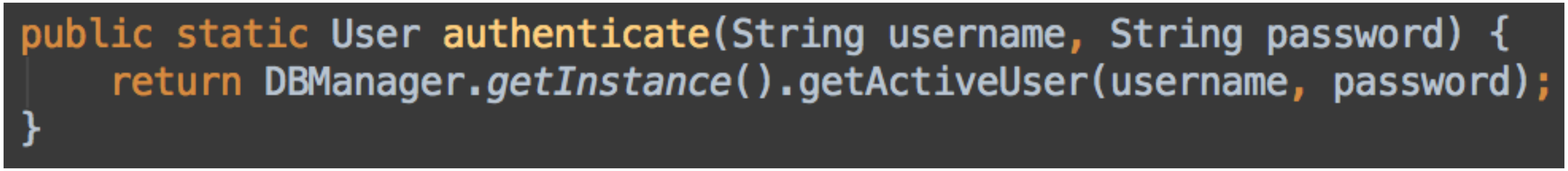


```java
public static User authenticate(String username, String password) {
    return DBManager.getInstance().getUser(username, password);
}
```

Method renaming



```java
public User getActiveUser(String username, String password) {
    return User.find.where().ieq("name", username).ieq("password",
            HashManager.getInstance().codeString(password)).ieq("active", "1").findUnique();
}
```



```java
public static User authenticate(String username, String password) {
    return DBManager.getInstance().getActiveUser(username, password);
}
```

Miscellaneous

- Durch die Entwicklung haben sich viele Unschönheiten (doppelter Code, falsche Benennung) eingeschlichen
- IDE hat Refactoring gut unterstützt
- Code hat nach Refactoring immer noch funktioniert

Where Am I

Vielen Dank für die Aufmerksamkeit