

Сообщение об ошибках

I. Ошибка компиляции в методе *Account login (String login, String password)*

1. Код до исправления:

```
a.callLogin(serverConnection, login,
passEncoder.makeSecure(password));
```

2. Данные, на которых наблюдается некорректное поведение: —

3. Полученное и ожидаемое значения: —

4. Код после исправления:

```
try {
    a.callLogin(serverConnection, login,
passEncoder.makeSecure(password));
} catch (OperationFailedException e) {
    registerException(e);
}
```

II. Не меняется значение сессии в методе *LocalOperationResponse callLogout()*

1. Код до исправления:

```
case ServerResponse.SUCCESS:
    return LocalOperationResponse.SUCCEED_RESPONSE;
```

2. Данные, на которых наблюдается некорректное поведение: `activeSession = 1L`

3. Полученное и ожидаемое значения:

- полученное: `1L`
- ожидаемое: `null`

4. Код после исправления:

```
case ServerResponse.SUCCESS: {
    activeSession = null;
    return LocalOperationResponse.SUCCEED_RESPONSE;
}
```

III. Лишний код в методе *LocalOperationResponse deposit(double amount)*

1. Код до исправления:

```
case ServerResponse.NO_MONEY:
```

```
Object r = response.data;
if(r instanceof Double)
    return new LocalOperationResponse(LocalOperationResponse.NO_MONEY,
(Double)r);
break;
case ServerResponse.SUCCESS:
    r = response.data;
```

2. Данные, на которых наблюдается некорректное поведение: —
3. Полученное и ожидаемое значения:
 - полученное: —
 - ожидаемое: —
4. Код после исправления:

```
case ServerResponse.SUCCESS:
    Object r = response.data;
```

IV. Неверные коды ответов в методе *LocalOperationResponse callLogin(IServerConnection serverConnection, String login, String password)*

1. Код до исправления:

```
switch(response.code) {
case ServerResponse.ALREADY_LOGGED:
    return LocalOperationResponse.ACCOUNT_MANAGER_RESPONSE;
case ServerResponse.NO_USER_INCORRECT_PASSWORD:
    return LocalOperationResponse.NO_USER_INCORRECT_PASSWORD_RESPONSE;
case ServerResponse.SUCCESS:{
    Object answer = response.data;
    if(answer instanceof Long) {
        activeSession = (Long) answer;
        return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
answer);
    }
    break;
}
}
return new
LocalOperationResponse(LocalOperationResponse.INCORRECT_RESPONSE,
response);
```

2. Данные, на которых наблюдается некорректное поведение: response.code =
ServerResponse.SUCCESS и response.data = null,
response.code = ServerResponse.UNDEFINED_ERROR
3. Полученное и ожидаемое значения:

- полученное:
LocalOperationResponse.INCORRECT_RESPONSE
- ожидаемое:
LocalOperationResponse.ENCODING_ERROR,
LocalOperationResponse.UNDEFINED_ERROR

4. Код после исправления:

```
switch(response.code) {
    case ServerResponse.ALREADY_LOGGED:
        return LocalOperationResponse.ACCOUNT_MANAGER_RESPONSE;
    case ServerResponse.NO_USER_INCORRECT_PASSWORD:
        return LocalOperationResponse.NO_USER_INCORRECT_PASSWORD_RESPONSE;
    case ServerResponse.SUCCESS: {
        Object answer = response.data;
        if(answer instanceof Long) {
            activeSession = (Long) answer;
            return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
            answer);
        }
        return new
        LocalOperationResponse(LocalOperationResponse.ENCODING_ERROR,
        response);
    }
}
return new
LocalOperationResponse(LocalOperationResponse.UNDEFINED_ERROR,
response);
```

V. Неверные коды ответов в методе *LocalOperationResponse callLogout()*

1. Код до исправления:

```
if(activeSession == null)
    return LocalOperationResponse.NOT_LOGGED_RESPONSE;
ServerResponse response = serverConnection.logout(activeSession);
switch(response.code) {
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;
    case ServerResponse.SUCCESS:
        activeSession = null;
        return LocalOperationResponse.SUCCEED_RESPONSE;
}
return new
LocalOperationResponse(LocalOperationResponse.INCORRECT_RESPONSE,
response);
```

- Данные, на которых наблюдается некорректное поведение: activeSession = null, response.code = ServerResponse.UNDEFINED_ERROR
- Полученное и ожидаемое значения:

- полученное:
LocalOperationResponse.NOT_LOGGED_RESPONSE,
LocalOperationResponse.INCORRECT_RESPONSE
- ожидаемое:
LocalOperationResponse.INCORRECT_SESSION_RESPONSE,
LocalOperationResponse.UNDEFINED_ERROR

4. Код после исправления:

```
if(activeSession == null)
    return LocalOperationResponse.INCORRECT_SESSION_RESPONSE;
ServerResponse response = serverConnection.logout(activeSession);
switch(response.code) {
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;
    case ServerResponse.SUCCESS:
        activeSession = null;
        return LocalOperationResponse.SUCCEED_RESPONSE;
}
return new
LocalOperationResponse(LocalOperationResponse.UNDEFINED_ERROR,
response);
```

VI. Неверные коды ответов в методе *LocalOperationResponse withdraw(double amount)*

1. Код до исправления:

```
if(activeSession == null)
    return LocalOperationResponse.NOT_LOGGED_RESPONSE;
ServerResponse response = serverConnection.withdraw(activeSession,
amount);
switch(response.code) {
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;
    case ServerResponse.NO_MONEY:
        Object r = response.data;
        if(r instanceof Double)
            return new
LocalOperationResponse(LocalOperationResponse.NO_MONEY, (Double)r);
        break;
    case ServerResponse.SUCCESS:
        r = response.data;
        if(r instanceof Double)
            return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
(Double)r);
        break;
}
return new
LocalOperationResponse(LocalOperationResponse.INCORRECT_RESPONSE,
response);
```

2. Данные, на которых наблюдается некорректное поведение: `activeSession = null`, `response.code = ServerResponse.UNDEFINED_ERROR`
3. Полученное и ожидаемое значения:
 - полученное:
`LocalOperationResponse.NOT_LOGGED_RESPONSE`,
`LocalOperationResponse.INCORRECT_RESPONSE`
 - ожидаемое:
`LocalOperationResponse.INCORRECT_SESSION_RESPONSE`,
`LocalOperationResponse.UNDEFINED_ERROR`
4. Код после исправления:

```
if(activeSession == null)
    return LocalOperationResponse.INCORRECT_SESSION_RESPONSE;
ServerResponse response = serverConnection.withdraw(activeSession,
amount);
switch(response.code) {
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;
    case ServerResponse.NO_MONEY:
        Object r = response.data;
        if(r instanceof Double)
            return new
LocalOperationResponse(LocalOperationResponse.NO_MONEY, (Double)r);
        break;
    case ServerResponse.SUCCESS:
        r = response.data;
        if(r instanceof Double)
            return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
(Double)r);
        break;
}
return new
LocalOperationResponse(LocalOperationResponse.UNDEFINED_ERROR,
response);
```

VII. Неверные коды ответов в методе *LocalOperationResponse deposit(double amount)*

1. Код до исправления:

```
if(activeSession == null)
    return LocalOperationResponse.NOT_LOGGED_RESPONSE;
ServerResponse response = serverConnection.withdraw(activeSession,
amount);
switch(response.code) {
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;
    case ServerResponse.SUCCESS:
        r = response.data;
        if(r instanceof Double)
```

```

        return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
        (Double)r);
        break;
    }
    return new
    LocalOperationResponse(LocalOperationResponse.INCORRECT_RESPONSE,
    response);

```

2. Данные, на которых наблюдается некорректное поведение: `activeSession = null`, `response.code = ServerResponse.UNDEFINED_ERROR`
3. Полученное и ожидаемое значения:
 - полученное:
 - `LocalOperationResponse.NOT_LOGGED_RESPONSE`,
 - `LocalOperationResponse.INCORRECT_RESPONSE`
 - ожидаемое:
 - `LocalOperationResponse.INCORRECT_SESSION_RESPONSE`,
 - `LocalOperationResponse.UNDEFINED_ERROR`
4. Код после исправления:

```

if(activeSession == null)
    return LocalOperationResponse.INCORRECT_SESSION_RESPONSE;
ServerResponse response = serverConnection.withdraw(activeSession,
amount);
switch(response.code){
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;
    case ServerResponse.SUCCESS:
        r = response.data;
        if(r instanceof Double)
            return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
            (Double)r);
        break;
}
return new
LocalOperationResponse(LocalOperationResponse.UNDEFINED_ERROR,
response);

```

VIII. Неверные коды ответов в методе *LocalOperationResponse getBalance()*

1. Код до исправления:

```

if(activeSession == null)
    return LocalOperationResponse.NOT_LOGGED_RESPONSE;
ServerResponse response = serverConnection.withdraw(activeSession,
amount);
switch(response.code){
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;

```

```

case ServerResponse.SUCCESS:
    r = response.data;
    if(r instanceof Double)
        return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
(Double)r);
    break;
}
return new
LocalOperationResponse(LocalOperationResponse.INCORRECT_RESPONSE,
response);

```

2. Данные, на которых наблюдается некорректное поведение: `activeSession = null`, `response.code = ServerResponse.UNDEFINED_ERROR`

3. Полученное и ожидаемое значения:

- полученное:

`LocalOperationResponse.NOT_LOGGED_RESPONSE`,
`LocalOperationResponse.INCORRECT_RESPONSE`

- ожидаемое:

`LocalOperationResponse.INCORRECT_SESSION_RESPONSE`,
`LocalOperationResponse.UNDEFINED_ERROR`

4. Код после исправления:

```

if(activeSession == null)
    return LocalOperationResponse.INCORRECT_SESSION_RESPONSE;
ServerResponse response = serverConnection.withdraw(activeSession,
amount);
switch(response.code){
    case ServerResponse.NOT_LOGGED:
        return LocalOperationResponse.NOT_LOGGED_RESPONSE;
    case ServerResponse.SUCCESS:
        r = response.data;
        if(r instanceof Double)
            return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
(Double)r);
        break;
}
return new
LocalOperationResponse(LocalOperationResponse.UNDEFINED_ERROR,
response);

```

IX. Неверно написан конструктор класса *AccountManager*

1. Код до исправления:

```

public void AccountManager(IServerConnection s, IPasswordEncoder
encoder) throws OperationFailedException {
    init(s, encoder);
}

```

2. Данные, на которых наблюдается некорректное поведение: —
3. Полученное и ожидаемое значения: —
4. Код после исправления:

```
public AccountManager(IServerConnection s, IPasswordEncoder encoder)
throws OperationFailedException {
    init(s, encoder);
}
```

X. Неверно неверный модификатор доступа у метода void init()

1. Код до исправления:

```
private void init (IServerConnection s, IPasswordEncoder encoder)
throws OperationFailedException {
```

2. Данные, на которых наблюдается некорректное поведение: —
3. Полученное и ожидаемое значения: —
4. Код после исправления:

```
public void init (IServerConnection s, IPasswordEncoder encoder)
throws OperationFailedException {
```

XI. Не инициализируется поле *IPasswordEncoder passEncoder*

1. Код до исправления:

```
public void init (IServerConnection s, IPasswordEncoder encoder)
throws OperationFailedException {
    if(serverConnection!=null)
        throw new
OperationFailedException(LocalOperationResponse.ALREADY_INITIATED_RESP
ONSE);
    if(s == null || encoder == null)
        throw new
OperationFailedException(LocalOperationResponse.NULL_ARGUMENT_EXCEPTIO
N);
    serverConnection = s;
    Arrays.stream(storedAccounts).parallel().forEach(
        one -> activeAccounts.put(one, new Account())
    );
}
```

2. Данные, на которых наблюдается некорректное поведение: —
3. Полученное и ожидаемое значения: —
4. Код после исправления:


```

public void init (IServerConnection s, IPasswordEncoder encoder)
throws OperationFailedException {
    if(serverConnection!=null)
        throw new
OperationFailedException(LocalOperationResponse.ALREADY_INITIATED_RESP
ONSE);
    if(s == null || encoder == null)
        throw new
OperationFailedException(LocalOperationResponse.NULL_ARGUMENT_EXCEPTIO
N);
    serverConnection = s;
    passEncoder = encoder;
    Arrays.stream(storedAccounts).parallel().forEach(
        one -> activeAccounts.put(one, new Account())
    );
}

```

XII. Неверный возврат в методе Account login (String login, String password)

1. Код до исправления:

```

public Account login (String login, String password) {
    if(login == null || password == null) {
        registerException(new
OperationFailedException(LocalOperationResponse.NULL_ARGUMENT_EXCEPTIO
N));
        return null;
    }
    accountsLock.lock();
    Account session = activeAccounts.get(login);
    if(session != null)
        registerException(new
OperationFailedException(LocalOperationResponse.ALREADY_INITIATED_RESP
ONSE));
    Account a = new Account();
    activeAccounts.put(login, a);
    try {
        a.callLogin(serverConnection, login,
passEncoder.makeSecure(password));
    } catch (OperationFailedException e) {
        registerException(e);
    }
    accountsLock.unlock();
    return a;
}

```

2. Данные, на которых наблюдается некорректное поведение: passEncoder.makeSecure(password) threw OperationFailedException; session != null; первая успешная авторизация

3. Полученное и ожидаемое значения:

- полученное: not null; not null; OperationFailedException и null
- ожидаемое: null; null; not null

4. Код после исправления:

```

public Account login (String login, String password) {
    if(login == null || password == null) {
        registerException(new
OperationFailedException(LocalOperationResponse.NULL_ARGUMENT_EXCEPTION));
        return null;
    }
    accountsLock.lock();
    Account session = activeAccounts.get(login);
    if(session != null) {
        registerException(new
OperationFailedException(LocalOperationResponse.ALREADY_INITIATED_RESPONSE));
        return null;
    }
    Account a = new Account();
    try {
        a.callLogin(serverConnection, login,
passEncoder.makeSecure(password));
    } catch (OperationFailedException e) {
        registerException(e);
        return null;
    }
    activeAccounts.put(login, a);
    accountsLock.unlock();
    return a;
}

```

XIII. Не устанавливается логин в методе
*LocalOperationResponse callLogin(IServerConnection
serverConnection, String login, String password)*

1. Код до исправления:

```

Object answer = response.data;
if(answer instanceof Long) {
    activeSession = (Long) answer;
    return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
answer);
}

```

2. Данные, на которых наблюдается некорректное поведение: login = "login"

3. Полученное и ожидаемое значения:

- полученное: null
- ожидаемое: "login"

4. Код после исправления:

```

if(answer instanceof Long) {
    activeSession = (Long) answer;
    this.login = login;
    return new LocalOperationResponse(LocalOperationResponse.SUCCEED,
answer);
}

```

XIV. Не обрабатываются все ошибки в методе *logout(Account account)*

1. Код до исправления:

```
b.callLogout();
```

2. Данные, на которых наблюдается некорректное поведение: *b.callLogout()* return *LocalOperationResponse.UNDEFINED_ERROR*
3. Полученное и ожидаемое значения:
 - полученное: —
 - ожидаемое: *LocalOperationResponse.UNDEFINED_ERROR*
4. Код после исправления:

```
LocalOperationResponse response = b.callLogout();  
if (response.code != LocalOperationResponse.SUCCEED) {  
    registerException(new OperationFailedException(response));  
}
```

XV. Всегда удаляется сессия в методе *logout(Account account)*

1. Код до исправления:

```
Account b = activeAccounts.remove(account.getLogin());  
if (b == null) {  
    registerException(new  
OperationFailedException(LocalOperationResponse.INCORRECT_SESSION_RESP  
ONSE));  
    return;  
}  
LocalOperationResponse response = b.callLogout();  
if (response.code != LocalOperationResponse.SUCCEED) {  
    registerException(new OperationFailedException(response));  
}
```

2. Данные, на которых наблюдается некорректное поведение: *b.callLogout()* return *LocalOperationResponse.UNDEFINED_ERROR*
3. Полученное и ожидаемое значения:
 - полученное: сессию удалили
 - ожидаемое: —
4. Код после исправления:

```
Account b = activeAccounts.get(account.getLogin());  
if (b == null) {
```

```

        registerException(new
OperationFailedException(LocalOperationResponse.INCORRECT_SESSION_RESP
ONSE));
        return;
    }
    LocalOperationResponse response = b.callLogout();
    if (response.code != LocalOperationResponse.SUCCEED) {
        registerException(new OperationFailedException(response));
        return;
    }
    activeAccounts.remove(b.getLogin());

```

XVI. Неполностью обрабатываются ошибки в методе *Account login(String login, String password)*

1. Код до исправления:

```

try {
    a.callLogin(serverConnection, login,
passEncoder.makeSecure(password));
} catch (OperationFailedException e) {
    registerException(e);
    return null;
}

```

2. Данные, на которых наблюдается некорректное поведение: `serverConnection.login(login, password)` `return`

`ServerResponse.NO_USER_INCORRECT_PASSWORD`

3. Полученное и ожидаемое значения:

- полученное: логин сохранен
- ожидаемое: логин не сохранен

4. Код после исправления:

```

try {
    LocalOperationResponse response = a.callLogin(serverConnection,
login, passEncoder.makeSecure(password));
    if (response.code != LocalOperationResponse.SUCCEED) {
        registerException(new OperationFailedException(response));
        return null;
    }
} catch (OperationFailedException e) {
    registerException(e);
    return null;
}

```