Bugs (problemas):

1/2. Strings and Boxed types should be compared using "equals ()"

PROBLEMA

**public** **boolean** galdutaMarkatu(Quote quo) {

**if**(kuota.getQuestion().getQuestionNumber()==quo.getQuestion().getQuestionNumber() && quo.getQuoteNumber()!= kuota.getQuoteNumber()) {

**this**.egoera="galduta";

**return** **true**;

}

**return** **false**;

}

SOLUCION

**public** **boolean** galdutaMarkatu(Quote quo) {

**if**(kuota.getQuestion().getQuestionNumber().equals(quo.getQuestion().getQuestionNumber()) && !(quo.getQuoteNumber().equals(kuota.getQuoteNumber()))) {

**this**.egoera="galduta";

**return** **true**;

}

**return** **false**;

}

3. “equals(Object obj)" should test argument type

PROBLEMA

**public** **boolean** equals(Object o) {

Apustua a = (Apustua) o;

**if**(a==**null**) {

**return** **false**;

}

**return** **this**.getApostuaNumber().equals(a.getApostuaNumber());

}

SOLUCION

**public** **boolean** equals(Object o) {

**if**(o==**null**)

**return** **false**;

**if** (**this**.getClass() != o.getClass())

**return** **false**;

Apustua a = (Apustua) o;

**return** **this**.getApostuaNumber().equals(a.getApostuaNumber());

}

4. "equals(Object obj)" and "hashCode()" should be overridden in pairs

PROBLEMA

@Override

**public** **boolean** equals(Object o) {

**if**(o==**null**)

**return** **false**;

**if** (**this**.getClass() != o.getClass())

**return** **false**;

Apustua a = (Apustua) o;

**return** **this**.getApostuaNumber().equals(a.getApostuaNumber());

}

SOLUCION

@Override

**public** **boolean** equals(Object o) {

**if**(o==**null**)

**return** **false**;

**if** (**this**.getClass() != o.getClass())

**return** **false**;

Apustua a = (Apustua) o;

**return** **this**.getApostuaNumber().equals(a.getApostuaNumber());

}

@Override

**public** **int** hashCode() {

**return** **this**.hashCode();

}

Code Smells: (no hay blocker)

1. Define a constant instead of duplicating this literal "Etiquetas" 3 times. [+3 locations] (critical)

PROBLEMA

**this**.setTitle(ResourceBundle.*getBundle*("Etiquetas").getString("ApustuBerdinak"));

lblApustuak = **new** JLabel(ResourceBundle.*getBundle*("Etiquetas").getString("ErabApustuak"));

btnClose = **new** JButton(ResourceBundle.*getBundle*("Etiquetas").getString("Close"));

SOLUCION

**private** **static** **final** String ***e*** = "Etiquetas";

**this**.setTitle(ResourceBundle.*getBundle*(e).getString("ApustuBerdinak"));

lblApustuak = **new** JLabel(ResourceBundle.*getBundle*(e).getString("ErabApustuak"));

btnClose = **new** JButton(ResourceBundle.*getBundle*(e).getString("Close"));

2. Replace the synchronized class "Vector" by an unsynchronized one such as "ArrayList" or "LinkedList". (major)

PROBLEMA

**private** Vector<Apustua> apustuak = **new** Vector<Apustua>();

**public** Vector<Apustua> getApustuak() {

**return** apustuak;

}

**public** **void** setApustuak(Vector<Apustua> apustuak) {

**this**.apustuak = apustuak;

}

SOLUCION

**import** java.util.ArrayList;

**private** ArrayList<Apustua> apustuak = **new** ArrayList<Apustua>();

**public** ArrayList<Apustua> getApustuak() {

**return** apustuak;

}

**public** **void** setApustuak(ArrayList<Apustua> apustuak) {

**this**.apustuak = apustuak;

}

3. Remove this unused method parameter "me". (major)

PROBLEMA

**public** ApustuBerdinakGUI(Registered copia, Registered me) {…}

JFrame a = **new** ApustuBerdinakGUI((Registered) list.getSelectedValue(), user);

SOLUCION

**public** ApustuBerdinakGUI(Registered copia) {…}

JFrame a = **new** ApustuBerdinakGUI((Registered) list.getSelectedValue());

4. Use isEmpty() to check whether the collection is empty or not. (minor)

PROBLEMA

**if** (events.size() == 0) {

btnApustuaGehitu.setEnabled(**false**);}

SOLUCION

**if** (events.isEmpty()) {

btnApustuaGehitu.setEnabled(**false**);}

5. Complete the task associated to this TODO comment. (info)

PROBLEMA

addWindowListener(**new** WindowAdapter() {

@Override

**public** **void** windowClosing(WindowEvent e) {

**try** {

//if (ConfigXML.getInstance().isBusinessLogicLocal()) facade.close();

} **catch** (Exception e1) {

// **TODO** Auto-generated catch block System.***out***.println("Error: "+e1.toString()+" , probably problems with Business Logic or Database");

}

System.*exit*(1);

}

});

SOLUCION

Es un comentario de posible mejora, por si quisiéramos implementar más código.