**Design Patterns**

בנוסף לתבנית הארכיטקטונית שהצגנו בקובץ הקודם (MVC), בפרויקט אנחנו משתמשים בשתי תבניות עיצוב נוספות:

**Factory Pattern** - כל ה-JDialogs שמוצגים בפרוייקט יורשים מה-API swing.Jdialog. בנוסף, בפרויקט שלנו, כל jdialog מממש את הממשק FootballScoutingJDialogs. כדי ליצור jdialog חדש, נפנה אל המתודה הסטטית CreateJDialog שנמצאת במחלקה JDialogFactory.

בהתאם לפרמטרים שהמתודה מקבלת, נוצר אובייקט חדש מטיפוס שממומש ע"י FootballScoutingJDialogs והוא הערך המוחזר.

כמובן שכדי להשתמש ב-jdialog שקיבלנו, נצטרך לעשות down cast לטיפוס הספציפי.

**Singleton Pattern –** המחלקה Database מחזיקה את ה-state הנוכחי של האפליקציה (= מערך השחקנים).

לכן, לא יהיה יותר ממופע אחד של המחלקה בפרוייקט (לא נרצה ליצור מצב שיוכלו להיות שני מופעים של המחלקה שמחזיקים מערכים שונים של שחקנים). לכן, מחלקה זו משתמשת ב-Singleton. כשנרצה ליצור מופע של המחלקה נפנה למתודה הסטטית getInstance שנמצאת במחלקה Database. היא תבדוק האם כבר קיים אובייקט מסוג Database בפרויקט. אם קיים, היא תחזיר אותו. אם לא – היא תיצור אחד חדש (ע"י הקונסטרקטור שלה שמוגדר כ-private), תשמור אותו במשתנה סטטי ותחזיר אותו. כך למעשה נמנע מצב בו יהיו שני מופעים שונים ל-state של האפליקציה.