TP – Agglomerations

En este TP se analizará un conjunto de aglomeraciones de la aplicación SportsTracker (http://www.saring.de/sportstracker/). La versión a analizar será la 5.7, la cual puede descargarse de https://sourceforge.net/projects/sportstracker/files/SportsTracker/SportsTracker%205.7.0/

Para compilarla, se necesita (como mínimo):

- Java SE Development Kit (JDK) 7
- Eclipse Juno (>4.0)
- Maven 3.0.3
- Groovy-Eclipse 2.7.0 (https://github.com/groovy/groovy-eclipse)
    - Groovy-Eclipse Feature
    - Groovy Compiler 2.0 Feature
    - Groovy-Eclipse M2E integration

El resultado del análisis del TP deberá entregarse en un informe antes del 10/11/17.

1. Por cada una de las siguientes 5 aglomeraciones:
    a. Analice el código fuente relacionado (esto es, los distintos code smells que forman parte de la aglomeración en cuestión y las clases Java afectadas)
    b. Identifique cuáles son los **problemas** que Ud. considera están asociados a los code smells de la aglomeración (por ej. tangling, scattering, duplicación de código, alto acoplamiento, baja cohesión, concentración de funcionalidad, ciclos, poca abstracción, etc.). Para cada problema, provea una breve explicación de su lógica o razonamiento para reconocerlo como un problema. De ser posible, ejemplifique con casos concretos del código para dichos problemas.
    c. En base al conjunto de problemas anterior, analice la probabilidad de que la aglomeración "como un todo" pueda generar un problema de mantenimiento critico o importante para el sistema (ya sea en lo inmediato, o en un futuro). Exprese su opinión utilizando una escala del 1 y 5 (1=improbable; 5=muy probable). Justifique su respuesta

Para el análisis, puede ayudarse con la siguiente tabla:

Aglomeración	**Code smells relacionados**	**Problemas de diseño identificados**	**Análisis global de la aglomeración (Escala: 1-5)**
#1			
#2			
#3			
#4			
#5			

2. Una vez que haya analizado todas las aglomeraciones, provea un orden para las mismas, desde la que considera más crítica para el sistema a la que considera menos critica.

Aglomeraciones:

-
    a.
    b.
    c.
    d.
    e.
    f.
    g.
    h.
    i.
    j.
    k.
    l.
    m.
    n.
    o.
    p.
    q.
    r.
    s.
    t.
    u.
    v.
- OptionsDialog (Intra-class)
    a. OptionsDialog.ok (Feature Envy)
    b. OptionsDialog.setInitialValues (Feature Envy)
    c. OptionsDialog.setInitialValues (Intensive Coupling)
- de.saring.sportstracker.gui.views.listview (Intra-component)
    a. BaseListView (Refuse Parent Bequest)
    b. BaseListView.createPopupMenu (Feature Envy)
    c. BaseListView.getTable (Shotgun Surgery)
    d. ExerciseListView (Refuse Parent Bequest)
    e. ExerciseListView.addColumnByIndex (Feature Envy)
    f. ExerciseListView.print (Feature Envy)
    g. ExerciseListView.updateView (Feature Envy)
    h. NoteListView.print (Feature Envy)
    i. WeoghtListView.print (Feature Envy)
- Exercise (Intra-class)
    a. Exercise.getAvgSpeed (Shotgun Surgery)
    b. Exercise.getDistance (Shotgun Surgery)

    c. Exercise.getDuration (Shotgun Surgery)
    d. Exercise.getEquipment (Shotgun Surgery)
    e. Exercise.getSportSubType (Shotgun Surgery)
    f. Exercise.getSportType (Shotgun Surgery)
- de.saring.sportstracker.gui.dialogs (Intra-component)
    a. ExerciseDialog.ok (intensive Coupling)
    b. FilterDialog.ok (intensive Coupling)
    c. OptionsDialog.ok (Feature Envy)
    d. OptionsDialog.setInitialValues (Feature Envy)
    e. OptionsDialog.setInitialValues (Intensive Coupling)
    f. OverviewDialog (Brain Class)
    g. OverviewDialog.addEquipmentTimeSeries (Dispersed Coupling)
    h. OverviewDialog.addExerciseTimeSeries (Brain Method)
    i. OverviewDialog.addExerciseTimeSeries (Dispersed Coupling)
    j. OverviewDialog.addSportSubTypeTimeSeries (Dispersed Coupling)
    k. OverviewDialog.createExerciseFilterForTimeStep (Feature Envy)
    l. OverviewDialog.mergeExerciseFilterIfEnabled (Feature Envy)
    m. OverviewDialog.mergeExerciseFilterIfEnabled (Intensive Coupling)
    n. OverviewDialog.updateDiagram (Brain Method)
    o. SportTypeDialog.editEquipment (Feature Envy)
    p. SportTypeDialog.editSportSubType (Feature Envy)
    q. SportTypeDialog.updateEquipmentList (Feature Envy)
    r. SportTypeDialog.updateSportSubtypeList (Feature Envy)
    s. SportTypeListDialog.deleteSportType (Feature Envy)
    t. StatisticDialog.changerFilter (Feature Envy)
    u. StatisticDialog.setFilterValues (Feature Envy)
    v. StatisticResultsDialog.setStatisticResults (Feature Envy)
- de.saring.sportstracker.data (Intra component)
    a. Equipment.getName (Shotgun Surgery)
    b. Equipment.setName (Shotgun Surgery)
    c. Exercise.getAvgSpeed (Shotgun Surgery)
    d. Exercise.getDistance (Shotgun Surgery)
    e. Exercise.getDuration (Shotgun Surgery)
    f. Exercise.getEquipment (Shotgun Surgery)
    g. Exercise.getSportSubType (Shotgun Surgery)
    h. Exercise.getSportType (Shotgun Surgery)
    i. ExerciseList.getExercisesForFilter (Feature Envy)
    j. ExerciseList.getExercisesForFilter (Intensive Coupling)
    k. ExerciseList.updateSportTypes (Feature Envy)
    l. ExerciseList.updateSportTypes (Intensive Coupling)
    m. SportSubType.getName (Shotgun Surgery)
    n. SportSubType.setName (Shotgun Surgery)
    o. SportType.getColor (Shotgun Surgery)
    p. SportType.getEquipmentList (Shotgun Surgery)
    q. SportType.getName (Shotgun Surgery)
    r. SportType.getSportSubTypeList (Shotgun Surgery)
    s. SportType.isRecordDistance (Shotgun Surgery)

t. SportType.setName (Shotgun Surgery)
u. Weight.getValue (Shotgun Surgery)

-
  a.
  b.
  c.
  d.
  e.
  f.