

التقرير

1. الإضافات والصفوف الجديدة

- ❖ تمت إضافة الصف Feature الذي يعبر عن قيمة حقل المرض
- ❖ أي اصبح المريض يمثل بلائحة من التحاليل المختلفة `List<Feature<?>>`
- ❖ كما هو موضح [هنا](#)

```
/**
 *
 * @author HASASN
 */
public class Feature <T> {

    private T value;//the value of the analysis

    /*used this function to create an instance from Feautre for type(className)and set the value */
    public static Feature<?> creatInstance(String className, String value)
        throws NoSuchMethodException ,ClassNotFoundException,InstantiationException
        ,IllegalAccessException,InvocationTargetException
    {
        ...٧ lines
    }

    public Double getMeasure() {
        ...٧ lines
    }

    public Feature(String className, String value)
        throws NoSuchMethodException ,ClassNotFoundException,InstantiationException
        ,IllegalAccessException,InvocationTargetException
    {
        ...٨ lines
    }

    public Feature(T value) {
        ...٣ lines
    }

    public void setValue(T value) {
        ...٣ lines
    }

    public T getValue() {
        ...٣ lines
    }

    @Override
    public String toString() {
        ...٣ lines
    }

}

/**
 *
 * @author HASAN
 */
public class Patient {

    private ArrayList<Feature<?>> analysis;//list of analysis each one is from different type
    private String grade ;

    public Patient() {
        ...٨ lines
    }

    public Patient( ArrayList<Feature<?> > a , String g ) {
        ...١١ lines
    }

    public <T> void setAnalysis( T value ,int index) {
        analysis.add(new Feature(value));
    }

    public void setGrade(String grade) {
        ...٣ lines
    }

    public void add(Feature f ) {
        ...٤ lines
    }

    public ArrayList<Feature<?> > getAnalysis() {
        ...٣ lines
    }

    public Feature getAnalysis(int index ) {
        ...٣ lines
    }

    public String getGrade() {
        ...٣ lines
    }

    @Override
    public String toString() {
        ...٧ lines
    }

}
```

- ❖ ويحتوي الصف Feature على تابع getMeasure() الذي يقوم باعطائنا قياسا للقيمة التي يحتويها الصف وهي ستدخل في الحساب ضمن تابع قياس المسافة (اقليديةأو مينكوفسكي ...)

❖ عند قراءة البيانات من الملف سنحتاج إلى تابع أو باني لكي نقوم بتحويل القيمة النصية التي معنا الى Feature<ClassType> حيث ClassType هو اسم الصف الموافق للعمود الذي نعبئه حالياً
لذلك قمنا ببناء التابع

public static Feature<?> createInstance(String className, String value)
الذي يقوم باعطائنا الغرض المطلوب انشاؤه كما هو موضح هنا :

```
try{
    f=new Feature(className: type,value);//to create an instacne from Feature of Type (type)and Value (value)
    //this function throws exceptions
    //ClassNotFoundException,InstantiationException,IllegalAccessException,InvocationTargetException
}
```

2. التعديلات على بقية الصفوف

❖ قمنا بالتعديل على توابع المسافة لتقوم باخذه الفرق بين القياسات كما هو موضح هنا :

```
for(int i=0;i<user.getAnalysis().size();i++){
    double x=user.getAnalysis(index: i).getMeasure()-other.getAnalysis(index: i).getMeasure();
    dist+=x*x;
}
```

❖ قمنا بالتعديل على الصف DataLoader حيث بعد قراءة السطر قمنا من اجل كل حقل باستدعاء التابع CreateInstance() مع تمرير قيمة الحقل ونوعه ومن ثم اضافة الناتج الى تحاليل المريض الجديد كما هو موضح هنا

```
for(int i =0;i<data.size()-1;i++){
    String value=data.get(index: i).replaceAll(regex: "\\s", replacement:"");
    String type=s.getType(index: i);
    Feature f;
    // incase the data were boolean so replace 1 into true and so
    value=(type.equals(anObject: "java.lang.Boolean"))? (value.equals(anObject: "1")||value.equals(anObject: "true"))?"True":"False" :value;
    try{
        f=new Feature(className: type,value);//to create an instacne from Feature of Type (type)and Value (value)
        //this function throws exceptions
        //ClassNotFoundException,InstantiationException,IllegalAccessException,InvocationTargetException
    }catch(ClassNotFoundException e1 ){
        System.out.println("errors occurs :unknown Class " +e1.getMessage()+"\n");
        throw e1;
    }
    catch(InvocationTargetException e2 ){
}
```

❖ بقية الصفوف لم يطرأ عليها أي تعديل

3. معالجة الاستثناءات

❖ بعد الإضافات الجديدة ظهر معنا استثناءات جديدة بعض هذه الاستثناءات نستطيع معالجته ضمن التابع نفسه وبعضه لا

1. ClassNotFoundException يحدث عندما لا يوجد نمط موافق لما هو موجود ضمن الملف وهذا الاستثناء نستطيع معالجته بوجهتي نظر
1 لانستطيع معالجته ولذلك لانستطيع اكمال البرنامج فنوقفه لاننا لانعلم ماهو الصف المطلوب

2 نقوم بوضع الصف هو String وهذا مااعتمدناه

2. InvocationTargetException يحدث عندما يتم استدعاء تابعيفة أو بناء جملة بشكل غير صحيح ويؤدي ذلك إلى حدوث استثناء داخلي. و يتم تمرير الاستثناء الداخلي إلى InvocationTargetException ويتم إرجاعه إلى المستخدم كاستثناء خارجي

