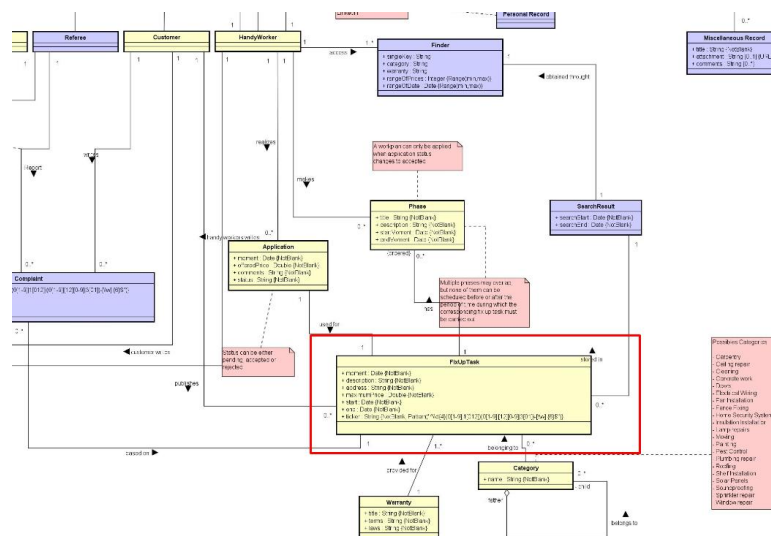


Acme Handy Worker – Json

The implemented utility is GSON's Java Library. This library lets us to parse objects in java to json data and viceversa. Thanks to "SchemaPrinter" class, the generated JSON is shown with all the necessary characters on the console.

Let's see the example we attach you on this PDF.

1. We would like to make a FixUp – Task.



2. If we have a look to the image before, we see that FixUp – Task has 3 different important relations (a part from all its predeterminate attributes). This is become into our Java Model as different attributes of different types.
 - Warranty
 - Category
 - Phases List (We must take into account that we have to make every object of the list before, so we will have to work with Phase Object)
3. Once we have this, we must implement before creating our task, every of the following items before. It means, we have to implement following the next order:



4. As a result of implementing the last objects and the attributes of the model, we have a Fix-Up Task created.

- To see how it works you have got available a JUnit4 test class in the following route: Resources/src/test/java/sample/Prueba.java. You must be careful with the console, because this shorts some words.

```
{
  "ticker":"181031-A8GT5G",
  "moment":"Oct 30, 2018",
  "description":"A FixUp description",
  "adress":"Random Street",
  "maximumPrice":200.0,
  "start":"Oct 24, 2018",
  "end":"Nov 24, 2018",
  "phases":[{
    "title":"Phase",
    "description":"A description for the phase",
    "startMoment":"Oct 20, 2018",
    "endMoment":"Oct 26, 2018",
    "number":0,"id":0,"version":0
  }],
  "warranty":{
    "title":"Test warranty",
    "terms":"Anyone read terms..",
    "id":0,
    "version":0},
  "category":{
    "name":"Tests",
    "id":0,"version":0
  },
  "id":0,
  "version":0
}
```

```
public class Prueba {

    @Test
    public void test() {
        final Category cat = new Category();
        cat.setName("Tests");
        final Warranty w = new Warranty();
        w.setTitle("Test warranty");
        w.setTerms("Anyone read terms..");
        final Phase p = new Phase();
        p.setTitle("Phase");
        p.setDescription("A description for the phase");
        p.setStartMoment(Date.valueOf("2018-10-20"));
        p.setEndMoment(Date.valueOf("2018-10-26"));
        p.setNumber(0);
        final List<Phase> ph = new ArrayList<>();
        ph.add(p);
        final Application app = new Application();
        app.setMoment(Date.valueOf("2018-10-31"));
        app.setOfferedPrice(1.0);
        app.setComments("A app comment");
        app.setStatus("ACCEPTED");
        final FixUpTask fix = new FixUpTask();
        fix.setTicker("181031-A8GT5G");
        fix.setMoment(Date.valueOf("2018-10-30"));
        fix.setDescription("A FixUp description");
        fix.setAdress("Random Street");
        fix.setMaximumPrice(200);
        fix.setStart(Date.valueOf("2018-10-24"));
        fix.setEnd(Date.valueOf("2018-11-24"));
        fix.setCategory(cat);
        fix.setWarranty(w);
        fix.setPhases(ph);

        final Gson gson = new Gson();
        final String s = gson.toJson(fix);
        SchemaPrinter.print(s);
    }
}
```

