5CCS2OSD Coursework, 2017

Counting 15% of module marks

October 12, 2017

Divided into 3 stages:

1. Requirements analysis + initial specification (class diagram, use case diagram)
2. Design (detailed steps for use cases, architecture diagram, detailed definitions of operations)
3. Implementation and testing.

Teams will be selected randomly.

Teams should have leader/coordinator + allocate tasks to members.

You can use a UML/drawing tool of your choice (list on slide 117 of Part 2 notes).

Deadline: November 30th, 4pm.

Submission by team leader/one representative on Keats.

1 Investment analysis system

- Coursework concerns management + analysis of investments.
- An investor may purchase *bonds*. A bond has a *term* (number of years to expiry), *coupon* (percentage of investment, paid to investor at regular intervals), *frequency* of payments. Bonds have names, eg. "UK government bond" and purchase date.
- Assume *frequency* = 1 per year, investment = 100.
- Investors receive back invested sum at term, + final coupon payment.

Eg., invest £100 in bond b paying 10% coupon annually for 5 years. 4 payments of £10, one of £110.

System enables new bonds to be defined + added to investor's portfolio.

- Bonds have a *price*: investor pays this with sum invested
- System should compute and display the *payout* of all bonds: sum of payments. £150 in example.

- System should compute + display bond *value*s: sum of *discounted* payments, using inflation rate r:

  X after N years of inflation r is $\frac{X}{(1+r)^N}$.

  Eg., £10 1 year in future, with 5% inflation, is worth 10/1.05 today = £9.52.

System computes *discount* for each payment, and hence *value* of bond. In example, value is £121.65 for r = 0.05.

- For each bond, system should calculate + show its *Macaulay duration* for rate r:

  $$duration = (\Sigma_{p:payments} \frac{p.time * coupon}{(1+r)^{p.time}} + \frac{term*100}{(1+r)^{term}})/value$$

  Where $p.time$ is time of p, in years from start of bond.

  In example, 4.25 years for r = 0.05

- For each bond, system should compute + show its *internal rate of return*: the r such that

  $$price = \Sigma_{p:payments} \frac{coupon}{(1+r)^{p.time}} + \frac{100}{(1+r)^{term}}$$

  Ie., $price = value(r)$.

This final use case is quite challenging to implement.

1.1 Coursework: First stage

- Carry out requirements elicitation and document system functionalities as use cases. Point out any ambiguous or incomplete requirements.
- Draw initial class diagram of system data including classes, attributes and associations, but not details of operations.

1.2 Second stage: Design

- Draw an architecture diagram for the system
- Write pseudocode activities for use cases + operations, defining steps of processing.
- Eg.:

```
query payout() : double
post:
  result = payments->collect(amount)->sum()
```

1.3 Coursework: Implementation

- Write implementations for your classes, use cases + operations in Java or another language.
- You may find it helpful to use the UML-RSDS code generator, or other UML tool with code generation capabilities.
- Write test cases and check your system using these.

1.4 Coursework: Final report

- No more than 15 pages, in PDF format.
- Named by number of group, eg: Team15OSD.pdf
- Describe team members roles + contributions to project.
- Include class diagram, use case diagram, operation + use case pseudocode, architecture diagram.
- Include test cases and results.
- Include a code listing of implemented classes and use case code.

Generally, team members will get the same mark, but students who have not participated will get a 0 mark.