

4 Lab: Modelação de interações e visualização de código

Enquadramento

Objetivos de aprendizagem



- Explicar uma colaboração entre sistemas através de diagramas de sequência.
- Usar a integração do VP como o IDE Eclipse para práticas de *round-trip engineering*.

Preparação

- Informação tutorial: [“What is Sequence Diagram?”](#)

Entrega

Cada grupo deve designar um “pivot” para o lab, que se encarrega de recolher os contributos dos colegas e fazer a entrega. Este papel é rotativo.

O exercício tem atividades para serem realizadas numa **aula** prática. O grupo deve submeter uma entrega, com as respostas aos exercícios assinalados com  

A entrega é um breve relatório, identificando o lab e os autores, e destacando (sublinhado) o aluno que foi o pivot. Anote os seus diagramas, colocando uma nota (*UML note element*) com a informação dos autores e a data de preparação.

Exercício

E4.1

Com base na [documentação disponível](#), explique, a partir da informação que se pode depreender da figura 6 no documento referido, como é que se passa a integração de sistemas quando ocorre um pagamento móvel, utilizando o Paypal.

E4.2

Um padrão recorrente na utilização de serviços acessíveis na Internet, através de interfaces programáticos, consiste na obtenção de um *token* (dadas credenciais válidas) que, depois, é passado em chamadas subsequentes para autorizar os pedidos.

Mostre a interação que ocorre quando se pretende usar o serviço de abreviação de um URL, utilizando (programaticamente) [o serviço PunyURL](#), do SAPO, considerando a modalidade de autenticação por *token*.

(Utilize a informação na documentação para criar um diagrama de sequência e os nomes dos métodos e parâmetros para usar nas mensagens.)

E4.3

Analise o projeto de código AmsLab02Clients¹ (disponível no repositório

¹ Este projeto está configurado para usar a ferramenta [Maven para gestão da build](#) (embora, para o exercício, bastará ter acesso ao código).

<https://bitbucket.org/iolvr/ams-labs>).

Modele num diagrama de sequência a informação que se pode depreender (do código) quanto à interação desencadeada pelo método `ClientMain#main()`.

Crie o modelo UML no VisualParadigm (pode começar por resolver em papel). É vantajoso criar as classes no modelo, antes de as usar como *lifelines* no diagrama de sequência.

E4.4

Nos próximos exercícios, vamos usar os diagramas da UML para criar visualizações do código em Java. Para isso, vamos usar especificamente a integração VisualParadigm / Eclipse.

Antes de avançar nos exercícios, é necessário configurar a integração do VP/Eclipse *Modelação e geração de código Java com VisualParadigm e Eclipse*:

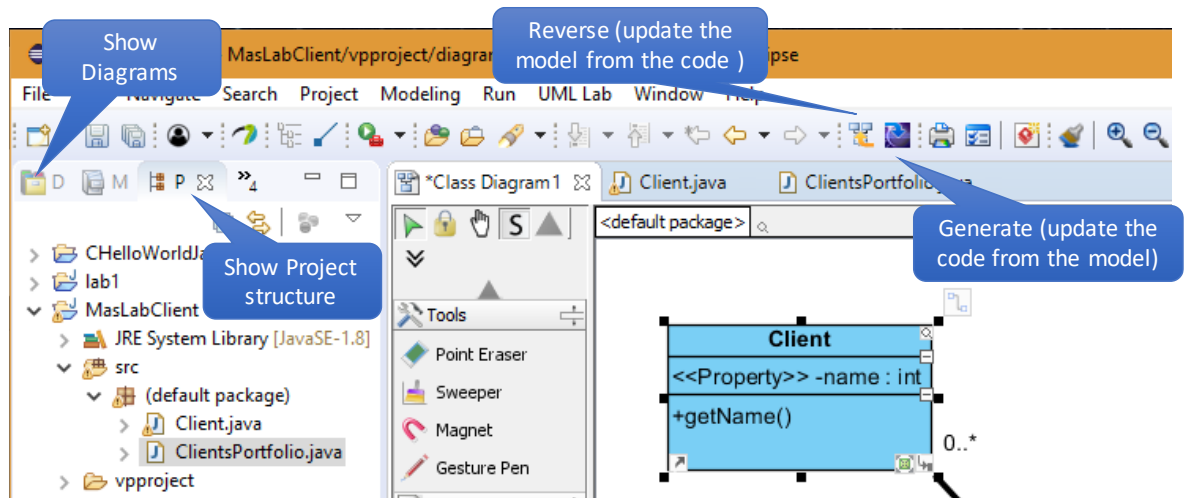
Nos seguintes pontos são referidos dois tutoriais da documentação do VisualParadigm que explicam o processo de integração. Não é para fazer tudo no primeiro; apenas o que está referido.

- a) Aceda ao [Tutorial 1](#) (*Getting Started*). Execute os passos sequencialmente até à secção “UML Modeling in Eclipse” e pare aqui (exclusive).

Note que depois de ter feito a integração (i.e., ficheiros copiados), deve fechar o VisualParadigm e trabalhar a partir do Eclipse).

Nota adicional para os utilizadores de Linux²

- b) Aceda ao [Tutorial 2](#) (*...with Round-trip Engineering*). Execute todos os passos sequencialmente (no Eclipse). Note a referência no último ponto (“3. This is the end of the tutorial...”) em que se sugere que introduza código nas classes e verifique que as alterações são refletidas no modelo.



E4.5



Veja a discussão do [padrão de desenho “Visitor”](#).

Se completou 4.4 com sucesso e tem a integração a funcionar:

- Prepare um projeto de desenvolvimento (Java) a partir do [exemplo disponível](#) e execute o método de demonstração.

² Dependendo da forma como o Eclipse foi instalado, pode obter erros de permissões na escrita de ficheiros relacionados com a integração das ferramentas. Para contornar o problema, considere utilizar uma versão do Eclipse dentro da área do utilizador. Para isso, transfira o zip com a instalação e expanda para uma subpasta da *home* do utilizador e utilize esta instância.

- Utilize a integração VP/Eclipse para criar um diagrama de classes com as classes da solução (importe o projeto para o Eclipse e arraste as classes para o diagrama).
- ➔ Acrescente ao modelo uma visualização da interação entre objectos desencadeada por `VisitorDemo#main()` mas recorrendo às funções de *reverse engineering* do VP: Menu Modeling > Instant Reverse Java to Sequence Diagram

Se não consegui completar a integração das ferramentas no passo 4.4:

- utilize [esta informação em alternativa](#).

Material suplementar

A versão Ultimate do IntelliJ IDEA, a que os alunos podem ter acesso no âmbito da [licença académica](#), inclui também um editor UML, que permite fazer o *round-trip engeneering* de forma muito prática (diagrama de classes).