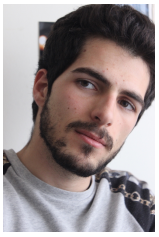
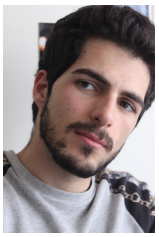
Relatório do Trabalho Prático de Programação Orientada a Objetos Grupo11 POO2017

Elísio Freitas Fernandes {55617}, Daniel Gonçalves Martins {73175}, and Nuno José Ribeiro da Silva {78879}

Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal
e-mail: {a55617, a73175, a78879}@uminho.pt


Elísio Freitas Fernandes {55617}


Daniel Gonçalves Martins {73175}


Nuno José Ribeiro da Silva {78879}

Resumo Serve o presente como relatório do projeto elaborado no âmbito da unidade curricular de Programação Orientada a Objetos. Projeto esse no qual se previa a elaboração de um programa para a empresa UMeR, o qual fosse capaz de garantir a prestação continua do seu serviço. Os requisitos de tal programa incluem a criação e manutenção da base de dados que inclui os dados dos clientes, bem como as informações dos seus colaboradores e das suas viaturas, utilizadas para a prestação do serviço em questão. Mais

ainda, está incluída a criação de uma interface de interação com os utilizadores com várias opções, como: criação de conta; posterior acesso e atualização dos dados da conta; consulta de histórico de serviços requeridos/prestados.

1 Introdução

Com o presente documento pretende-se apresentar resumidamente o trabalho prático elaborado pelo grupo, como proposta de resolução do enunciado apresentado, no seguimento da UC de programação Orientada aos Objetos. As necessidades da empresa UMeR foram abordadas e tratadas usando java, temos recorrido maioritariamente ao Atom como editor de texto e o Terminal como compilador, bem como o ***** e a consola do linux para windows. Irão ser apresentadas as classes elaboradas e a várias escolhas especificas feitas em cada uma delas, a sua hierarquia e sua razão de ser.

2 Arquitetura

Nesta proposta foram utilizados vários tipos de classes com diferentes atributos. Desde classes concretas a classes abstratas, classes com e sem implementações de interfaces e subclasses de classes abstratas.
Por questões de necessidade de gravação do estado da aplicação, todas as classes tem em comum a implementação da interface *Serializable*, oferecendo, assim, um método simples de gravação dos objetos com todos os seus estados atuais em ficheiro, para posterior consulta.

2.1 Classe UmeR

Utiliza Maps e tipos primitivos para guardar os seus dados. Possui um registo de toda a informação necessária para o bom funcionamento da empresa. Contém quatro estruturas de dados Map onde se encontram guardados 1. User como *value* e email como *key*, para guardar todos os utilizadores registados na empresa, quer clientes, quer condutores ; 2. Vehicle como *value* e licensePlate como *key*, para guardar a informação da viaturas ao serviço da empresa; 3. Trip como *value* e id de viagem como *key*, para manter um histórico de todas as viagens efetuadas através da empresa; 4. Email como *value* e licensePlate como *key*, para manter um registo do veículo associado a cada condutor. Existe ainda a variável isLogged, do tipo *boolean*, que indica se existe um utilizador "logado". Se sim, então o seu e-mail estará presente na variável loggedUserEmail, do tipo *String*. Por último, existe na classe uma variável tripNumber, do tipo *Integer*, responsável por garantir a sequencialização dos identificadores únicos da viagem, utilizados no Map das Trips.
Foram aglomerados todos os dados do tipo user, respetivamente, todos os dados de tipo Trip e veiculo, num só Map, uma vez que eram compatíveis, de maneira a ser possível efetuar operações sobre todos os utilizadores de uma só vez, bem como para facilitar a adição de novos tipo de utilizadores.

2.2 One subsection

abdc...

2.3 One more...

3 Hierarquia

Genericamente, como pode ser visto na figura ******, temos a ligação

4 Tipos de dados usados

por fotos dos dados usados e justificar o uso de maps e lists generalista - abstração

5 Manual de utilização

fotos e indicações de navegação

According to Table 1...

(a) Delay and jiiter	(b) Delay and loss
(c) Delay and throughput	(d) Jitter and loss
(e) Jitter and throughput	(f) Loss and throughput

Figura 1. Tabela exemplo.

6 Conclusions

Neste trabalho...

Referências

1. Zadeh, L.: Fuzzy sets (1965)
2. Nguyen, H., Walker, E.: First course in fuzzy logic. Boca Raton: Chapman and Hall/CRC Press (1999)