



Programação Orientada a Objetos
2022/23
1º Trabalho Prático

Número de alunos por grupo: 2

Data limite de entrega: 28-4-2023 até às 23h55

(30/4 para Trab. Est. e Dir. Asso.)

TIU by EST

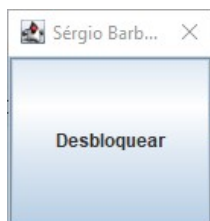
A mobilidade urbana e a redução das emissões poluentes estão na ordem do dia. Por isso, o grupo de investidores, AlbinvEST, responsável pelo sucesso de várias empresas, investiu na criação de uma empresa de aluguer de trotinetes: a TIU – Transportes Individuais Urbanos. Para o desenvolvimento do sistema informático da empresa, contataram a parceira de sempre: a EST.

Nesta primeira fase, o sistema irá simular o aluguer das trotinetes por um utente, bem como poder colocar uma trotineta em carga ou em manutenção. No futuro, mas não neste trabalho, pretende-se implementar o sistema no seu todo, pelo que as **decisões tomadas para este trabalho devem ser escaláveis e facilmente alteráveis**.

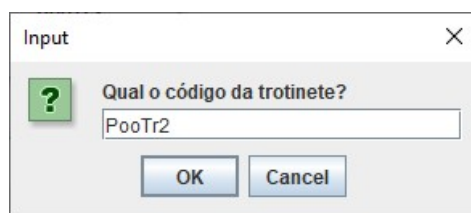
Funcionamento do sistema

O Sistema está dividido em 3 aplicações: uma que simula a aplicação de telemóvel do utente, uma que simula a trotineta e uma que é a aplicação de controlo central.

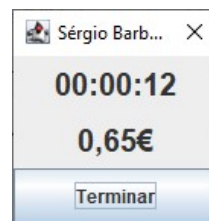
A aplicação do utente permite-lhe alugar uma trotineta e terminar o aluguer. Cada utente terá uma janela destas. Quando arranca, a aplicação apresenta um único botão que serve para desbloquear uma trotineta e começar o aluguer. No sistema real o utente usaria um QR code ou outro modo de identificar a trotineta, mas nesta versão simulada terá de introduzir o código identificador da trotineta. Assim que introduzir o código e caso a trotineta esteja disponível, a aplicação apresenta o tempo do aluguer e o custo associado. Para terminar o aluguer, basta pressionar o botão de terminar. Após isso, a aplicação volta ao estado inicial.



Janela do utente sem aluguer



Introdução do código da trotineta

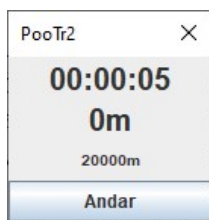


Janela durante o aluguer

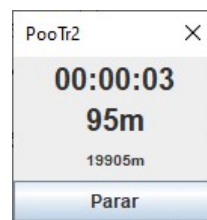
A aplicação que simula a trotineta começa por apresentar a informação de desbloqueio. Quando um utente desbloqueia a trotineta, a aplicação apresenta a informação do tempo de aluguer, a distância percorrida e a distância que tem de autonomia. Para simular o movimento da trotineta é preciso pressionar o botão “Andar”. Depois de pressionado o botão, este passa a ser o botão de “Parar”, para permitir simular a paragem da trotineta. Quando o utente termina o aluguer, a aplicação retorna ao estado de bloqueada. Cada trotineta terá uma janela destas.



Janela da trotineta bloqueada



Trotineta em aluguer (parada)



Trotineta em aluguer (em andamento)

A aplicação principal está dividida em duas partes: a de controlo das trotinetes e a de gestão de utentes. Para alternar entre as duas basta usar as etiquetas correspondentes.

Na janela de controlo das trotinetes são apresentadas duas tabelas diferentes: uma com as trotinetes que não estão a ser usadas e outra com as trotinetes a ser usadas. Como se pode ver na imagem seguinte a informação em cada uma das tabelas é diferente e autoexplicativa.

Trotinetes					
Não usadas					
Código	Autonomia	Velocidade	Carregar	Manutenção	
PooGo	30000	6	<input type="checkbox"/>	<input type="checkbox"/>	
Poo20	30000	7	<input type="checkbox"/>	<input type="checkbox"/>	

Usadas						
Código	Autonomia	Velocidade	Início	Custo	Distância	Mover
PooTr1	6620	5	28/ 3/2023 22:41:39	0.95000005	180	Parada
PooTr2	19455	5	28/ 3/2023 22:41:58	0.95000005	545	Andar

Janela de controlo das trotinetes

Uma trotinete pode ser colocada a carregar ou em manutenção (clitando no quadrado respetivo). Qualquer uma destas opções impossibilita o uso da trotinete até que seja novamente colocada em funcionamento (quer retirando de carregamento ou de manutenção). Reparar que, quando está em carregamento, a trotinete também está em manutenção.

Trotinetes					
Não usadas					
Código	Autonomia	Velocidade	Carregar	Manutenção	
PooTr1	9820	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
PooGo	30000	6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Poo20	30000	7	<input type="checkbox"/>	<input type="checkbox"/>	

Usadas						
Código	Autonomia	Velocidade	Início	Custo	Distância	Mover
PooTr2	18435	5	28/ 3/2023 22:41:58	1.40000001	1565	Parada

Janela de controlo das trotinetes (com algumas em manutenção)

Na parte de gestão dos clientes, e na sua parte esquerda, a central disponibiliza uma tabela com todos os utentes e um botão que permite criar um novo utente. Ao clicar num dos utentes da lista, aparecem, no lado direito da janela, os seus dados, nomeadamente, o nome de utilizador, o nome completo e uma tabela com os últimos alugueres realizados pelo cliente (se o cliente está a alugar, o aluguer atual NÃO aparece).

TrotinEST

Trotinetes

Utentes

Utentes

User	Nome
codeguru@guru.com	Harry Hacker
aluno1@ipcbcampus.pt	Aluno Número 1
aluno2@ipcbcampus.pt	Aluno Número 2
fsergio@ipcb.pt	Sérgio Barbosa
jojo89@g.com	João José Silva

Novo Utente

Dados do Utente

fsergio@ipcb.pt

Sérgio Barbosa

Alugueres passados

Início	Fim	Distância	Custo	Trotinete
29/ 3/2023 16:23:47	29/ 3/2023 16:26:41	845	0.95000005	PooTr2
29/ 3/2023 16:26:50	29/ 3/2023 16:31:48	1758	1.25	PooGo
29/ 3/2023 16:31:54	29/ 3/2023 16:35:13	1358	1.1	Poo20

Janela de gestão dos utentes, com um deles selecionado

Para criar um utente é só pressionar o botão “Novo Utente” e introduzir os dados respetivos.

Aspetos técnicos a ter em conta

Identificação dos utentes

Cada utente terá um user name e o nome completo. O user name é um endereço de email, e terá de ser único no sistema, pois será por este que o utilizador será identificado. O nome corresponde ao nome propriamente dito do utilizador. Para simplificar, não se usará qualquer palavra-chave para aceder ao sistema.

Tempo, velocidades e distâncias

O tempo a usar no sistema é o tempo real do computador.

Para simplificar, na simulação do movimento da trotinete usa-se uma velocidade fixa, dada em metros por segundo, sendo assim um número inteiro. Trotinetes diferentes poderão ter velocidades diferentes.

O sistema, a cada segundo, chama um método da trotinete para simular o funcionamento desta. Neste método basta somar à distância percorrida o valor da velocidade, e claro, diminuir à autonomia restante o mesmo valor.

Carregamento das trotinetes

Cada trotinete tem uma autonomia máxima (dada em metros), que representa a distância máxima que pode percorrer com a bateria completamente carregada. Tem também uma autonomia restante, que indica qual a distância que ainda pode percorrer até ficar sem bateria. Quando a autonomia restante estiver abaixo dos 500 metros, a trotinete deixa de estar disponível para novos alugueres, embora seja permitido acabar o aluguer atual.

Para simular o carregamento, usa-se uma velocidade fixa de carregamento (será a mesma velocidade para todas as trotinetes) de 400 metros por segundo. Isto significa que, no método que simula o funcionamento da trotinete, se deve aumentar a autonomia restante de 400 metros, enquanto estiver em carregamento e até atingir a autonomia máxima.

Custo do aluguer

O utente paga a utilização da trotinete em blocos de um minuto. O preço por minuto é de 0.15€. O minuto paga-se logo no início, mesmo que o utente não usufrua de todo o minuto. Por exemplo, se o utente andar 9 minutos e 30 segundos, paga 10 minutos de utilização, dando um total de 1.5€.

Ao preço do aluguer, acresce uma taxa de desbloqueio de 0.50€. No exemplo anterior, o utente pagaria o total de 2€ pelo aluguer de 9 minutos e 30 segundos (1.5€ do tempo e 0.5€ da taxa de desbloqueio).

Unidades a usar

Os preços serão sempre em euros e devem ser armazenados em float.

A velocidade da trotinete é em metros por segundo e é um inteiro.

As distâncias e autonomias são em metros e são inteiros.

A velocidade de carregamento é de 400 metros por segundo.

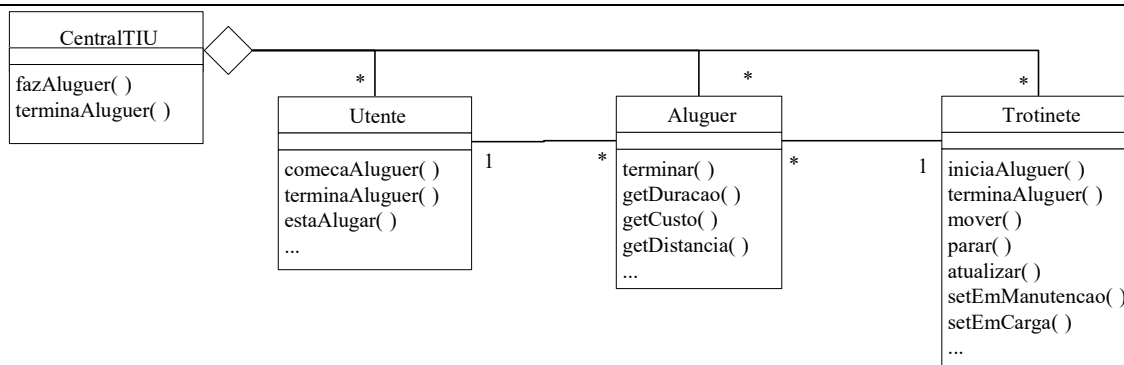
Os tempos dos alugueres devem ser do tipo `LocalDateTime`.

Manutenção e evolução

Devem tomar as melhores decisões para evitar alterações em várias partes do código caso haja alterações em alguns parâmetros. Exemplos de parâmetros que podem mudar é o preço do aluguer por minuto, a taxa de desbloqueio e a velocidade de carregamento.

Diagrama de classes

O diagrama de classes, incompleto, para este trabalho é o seguinte:



Código fornecido

Para este trabalho já são disponibilizadas várias classes que, contudo, estão incompletas. Há muitas variáveis a serem declaradas e métodos a serem implementados. Todos os métodos apresentados no código DEVEM ser implementados, mesmo que durante a implementação do sistema não venham a ser necessários. **Além dos métodos mencionados poderão ter de desenvolver novos métodos. Além dos campos indicados podem ser necessários novos campos.** Quer os métodos quer os campos devem ser colocados na classe certa.

O grupo responsável pelo desenvolvimento das interfaces já cumpriu todas as suas tarefas. O grupo do sistema (você) é que ainda não cumpriu com a sua parte e terá de o fazer até à data de entrega. Nas classes das interfaces estão assinalados (com `// TODO`) os locais onde terão de acrescentar o vosso código, de modo a pôr todo o sistema a funcionar. Podem acrescentar código em mais locais que não os assinalados, se a vossa solução assim o exigir.

Classes a desenvolver

Estas são as classes que se devem desenvolver. Algumas classes têm, no código fornecido, métodos declarados que terão de ser implementados. Podem ter de implementar mais métodos além dos que estão indicados no código. Todas elas deverão ter os construtores, getters e setters relevantes.

Utente

Esta classe representa um utente. Deve ter o user name (email), nome, se tem algum aluguer no momento e uma lista dos alugueres todos que já realizou.

Trotinete

Esta classe representa uma trotinete. Deve ter o código identificativo, se tem algum aluguer no momento, uma lista de alugueres que já realizou, a autonomia, a autonomia restante, a velocidade de deslocamento, a distância total percorrida, e a distância percorrida no aluguer atual.

Aluguer

Esta classe representa um aluguer. Deve ter o utente e a trotinete envolvidos no aluguer, bem como a data e hora de início do aluguer. Caso já tenha terminado, deve ter ainda a indicação da data e hora de fim, a distância percorrida e o custo associado.

CentralTIU

Classe que representa o sistema. Deve armazenar todos os utentes e todas as trotinetes.

Arranque

Esta classe tem o método main e é ela que inicia os vários componentes do sistema e define a configuração de teste.

JanelaXXX

Classes que representam as várias interfaces da aplicação. Nesta classe devem ser implementados os blocos de código indicados nos TODO.

Configurações de teste

No código da classe `Arranque` deve ser colocado o código de teste. **O software deve estar preparado para qualquer configuração, não deve ficar limitado à configuração de teste.** Para facilitar os testes, devem-se criar os seguintes elementos.

Utentes a usar

User Name	fsergio@ipcb.pt	jojo89@g.com	codeguru@guru.com	aluno@ipcbcampus.pt*
Nome	Sérgio Barbosa	João José Silva	Harry Hacker	Nome do Aluno*

** repetir a informação dos alunos para cada elemento do grupo.*

Trotinetes a usar

Código	Autonomia	Velocidade	Autonomia restante (no arranque)
PooTr1	20000	5	400
PooTr2	20000	5	20000
PooGo	30000	6	8000
Poo20	30000	7	30000

Objetivos

O objetivo deste trabalho é implementar o sistema tal como está descrito, dando-se especial atenção à correta implementação da arquitetura do sistema.

Sugestão dos passos a seguir	Critérios de avaliação	Conceitos a utilizar
Estudar as teóricas e as fichas práticas	Estrutura de packages	Análise de problemas
Identificar as classes presentes, as relações entre elas e a sua interface;	Estrutura de classes	Declaração de classes
Implementar a classe <code>Utente</code>	Interface das classes	Relações entre classes
Implementar a classe <code>Trotinete</code>	Implementação das classes	
Implementar a classe <code>Aluguer</code>	Interligação entre os objectos	
Implementar a classe <code>CentralTIU</code>	Funcionamento do sistema	
Ajustar a interação com o utilizador	Relatório final	
	Defesa do trabalho	

Notas:

Todas as classes devem pertencer a um package que não o por defeito.

A otimização de algumas das operações será beneficiada. Por outro lado, soluções pouco eficazes (desperdício de memória exagerado, código redundante, etc) serão penalizadas.

O código deve seguir as regras de codificação estabelecidas na disciplina.

O que se deve entregar:

- As regras para entrega serão anunciadas mais tarde.

Dúvidas:

Eventuais dúvidas podem ser esclarecidas junto dos docentes da disciplina. Pode também ser usada a plataforma de e-learning que terá um fórum onde poderão colocar dúvidas.

Os trabalhos ficarão sujeitos a uma defesa na aula prática/teórica da última semana de aulas.

Trabalhos idênticos serão anulados.