

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	

Enunciado

Considere a existência de um sistema público de aluguer de bicicletas. O sistema é composto por vários depósitos de bicicletas espalhados pela cidade. Cada depósito contém cerca de uma dúzia de bicicletas numa área para o efeito. O depósito de bicicletas contém um determinado número de lugares devidamente numerados (um lugar para cada bicicleta), onde as bicicletas estão inicialmente colocadas com um sistema “lock/unlock”. Este sistema encontra-se conectado ao sistema principal e funciona da seguinte forma:

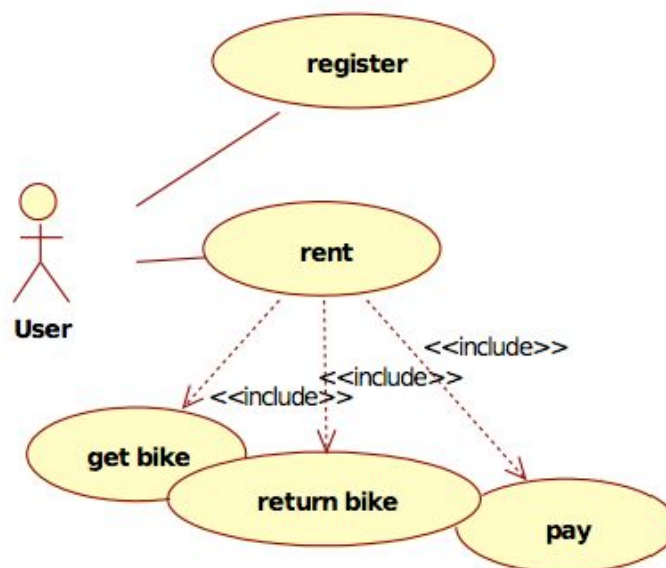
- Se um utilizador está autenticado e tem crédito na sua conta, pode alugar uma bicicleta, o sistema “liberta” a bicicleta e o utilizador pode retirar uma bicicleta, tornando-se responsável por ela até que a devolva.
- Quando o utilizador pretende devolver a bicicleta, escolhe um lugar disponível no depósito de bicicletas para o estacionamento da bicicleta. O sistema “lock/unlock” detecta a bicicleta e automaticamente e “prende” a bicicleta. A partir desse momento o utilizador deixa de ser responsável pela bicicleta.

Para que uma pessoa possa alugar bicicletas terá de se registar, fornecendo um conjunto de dados entre os quais o nome e a informação do cartão de crédito, recebendo um login (ID). Se, de seguida, o utilizador desejar alugar uma bicicleta, dirige-se a um depósito e introduz, através de um interface para o efeito (exemplo: monitor tátil), o seu login (ID). Após as respectivas validações, o sistema selecciona uma bicicleta entre as disponíveis e “liberta-a”. No momento da entrega, não há qualquer interacção com qualquer interface.

Para além das funcionalidades descritas, o sistema deve:

- Acompanhar o estado de todas as bicicletas e alugueres.
- O utilizador deverá pagar uma determinada quantia pelo aluguer, dependendo da duração do respectivo aluguer. Os utilizadores podem escolher uma bicicleta de um determinado depósito e entrega-la num outro qualquer. O sistema deverá monitorizar a manutenção das bicicletas (uma bicicleta que nunca foi alugada pode provavelmente estar avariada).
- O sistema deverá monitorar a distribuição das bicicletas pelos depósitos.

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	



O diagrama da Figura 1 apresenta os casos de uso descritos anteriormente. Os seguintes cenários descrevem com maior detalhe alguns dos casos de utilização apresentados na Figura 1:

Cenário 1: Aluguer (rent) realizado com sucesso

Pré-condição: O utilizador encontra-se registado, utilizador pode concluir as operações realizando o pagamento.

Pós-condição: O utilizador recebe a bicicleta e o sistema regista a operação

Etapa	Descrição
1	Utilizador insere a sua identificação na interface de utilizador
2	O sistema verifica a identificação do utilizador de forma a verificar se o utilizador se encontra registado
3	O sistema verifica se cliente pode realizar o pagamento (se tem pelo menos o valor mínimo de 1 euro).
4	O sistema seleciona uma bicicleta disponível numa determinada posição e informa o utilizador
5	O sistema regista o aluguer da bicicleta para o utilizador

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	

Cenário 2: Entrega de aluguer (return bike)

Pré-condição: O utilizador encontra-se registado

Pós-condição: A bicicleta é entregue e realizado o pagamento

Etapa	Descrição
1	O Utilizador entrega a bicicleta
2	O sistema identifica o utilizador, calcula o valor a pagar e desconta esse valor na conta do cliente
3	O sistema informa o utilizador do seu saldo

A dedução de casos de teste e respetiva implementação deverá ser baseada na especificação de cada um dos seguintes métodos:

Especificação para o método: getBicycle

requer: IDUser > 0, IDDeposit > 0, starttime >= 0

garante:

Se IDUser não existe, então

uma exceção específica é emitida

Senão

Se o depósito não existir, então

retorna -1

Senão

Se crédito do utilizador < 1, então

retorna -1

Senão

Se não existirem bicicletas disponíveis ou o utilizador tenha um aluguer ativo, então

retorna -1

Senão

a bicicleta é associada (sinalizando o starttime) ao cliente, o lugar é libertado e é retornado o identificador da bicicleta que será utilizada.

Especificação para o método: returnBicycle

requer: IDUser >= 0, IDDeposit > 0, endtime >= 0

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	

garante:

*Se IDUser OU IDDeposit não existe OU a bicicleta não está associada a aluguer ativo, então
 retorna -1
 Se não existirem lugares de entrega livres, então
 retorna -1
 Senão
 calcula o pagamento através do método de pagamento (para o caso de uso: pay): bicycleRentalFee e retorna o saldo atual do cliente*

Especificação para o método: bicycleRentalFee

requer: rentalProgram >=0, starttime >= 0, endtime >= 0, startTime <= endtime, nRentals (número de alugueres realizados) >=0

garante:

*Se rentalProgram = 1, então
 retorna (endtime - starttime) * rentalFee
 Senão
 Se rentalProgram = 2 E resto da divisão inteira do nRentals por 10 é !=0, então
 Se (endtime - starttime) <= 10, então
 retorna rentalFee * (endtime - starttime)
 Senão
 retorna 10*rentalFee + ((endtime - starttime)-10)* rentalFee
 /2
 Senão
 retorna 0*

Especificação para o método: verifyCredit

requer: IDUser >= 0

garante:

*Se o utilizador com IDUser não existir OU o valor do crédito < 1, então
 retorna falso
 Senão
 retorna verdadeiro*

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	

especificação para o método: addCredit

requer: IDUser >= 0, amount > 0

garante:

Que ao valor do crédito do cliente (caso exista) é adicionado o valor associado ao conteúdo da variável amount

especificação para o método: registerUser

requer: IDUser >= 0, name com valor não nulo, rentalProgram: 1 ou 2

garante:

Se utilizador já existir, então
uma exceção específica é emitida
Senão
utilizador é adicionado

Não é necessário testar os métodos que não estão especificados.

Os requisitos do problema apresentado encontram-se implementados na biblioteca: BikeRentalSystem.

Especifique casos de teste utilizando as técnicas “Equivalence Class Partitioning” e “Boundary Value Analysis” no nível de testes unitários (especifique casos de teste para cada método, mas exclua o método principal e métodos construtores). Especifique *test inputs*, *execution conditions* e *expected outputs*, assegurando que os casos de teste cobrem *Valid equivalence classes* e *Invalid equivalence classes*. Implemente os casos de teste e execute-os utilizando a framework: JUnit 5¹.

Deve elaborar um relatório de testes. Registe os resultados dos testes adequadamente no seu relatório de teste. Lembre-se de especificar o test case ID, objeto testado, descrição, entradas, saídas esperadas e outras informações que considere úteis. Para além do relatório de testes, o código usado para implementar cada um dos testes deverá devidamente documentado de forma a identificar cada caso de teste.

O relatório (deve basear-se no documento: *IEEE Standard for Software Unit Testing* disponível na plataforma moodle) deverá ser entregue através da plataforma moodle, enquanto que o código desenvolvido deverá ser disponibilizado através de um repositório GIT (que deverá ser devidamente identificado no relatório) até dia **18 de Novembro pelas 23:55**. O repositório GIT entregue deverá refletir as contribuições de cada membro do grupo, assim como a evolução do projeto. Por precaução deverá também entregar o relatório na plataforma moodle.

O trabalho deverá ser realizado por grupos de **3 elementos**. A submissão deverá ser realizada na plataforma moodle até dia **13 de Novembro**.

¹ <https://junit.org/junit5/>

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	

Anexo 1

```

public interface AdministratorInterface {
    /**
     * Registo de utilizador no sistema Bicycle Rentals
     *
     * @param IDUser - id do utilizado
     * @param name - nome do utilizador
     * @param rentalProgram - tarifário de aluguer 1 ou 2
     * @throws UserAlreadyExists - Se utilizador com o mesmo id já existir no sistema
     a excepção é lançada
     */
    public void registerUser(int IDUser, String name, int rentalProgram) throws
    UserAlreadyExists;

    /**
     * Adição de depósitos e locks ao sistema Bicycle Rentals.
     * Para a sua criação apenas é necessário indicar um id não existente no sistema.
     *
     * @param idDeposit - id de depósito existente ou depósito a criar
     * @param idLock - id de cadeado existente ou a criar
     */
    public void addLock(int idDeposit, int idLock);

    /**
     * Adição de depósitos, locks e bicicletas ao sistema Bicycle Rentals.
     * Para a sua criação apenas é necessário indicar um id não existente no sistema.
     *
     * @param idDeposit - id de depósito existente ou depósito a criar
     * @param idLock - id de cadeado existente ou a criar
     * @param idBike - id de bicicleta existente ou a criar
     */
    public void addBicycle(int idDeposit, int idLock, int idBike);

    /**
     * Adição de crédito à conta de um utilizador
     *
     * @param idUser - id do utilizador
     * @param amount - quantia a adicionar
     */
    public void addCredit(int idUser, int amount);

    /**
     * @return lista de utilizadores no sistema Bicycle Rentals
     */
    public List<User> getUsers();

    /**
     * @return lista de depósitos no sistema Bicycle Rentals
     */
    public List<Deposit> getDeposits();

    /**
     * @return lista de bicicletas no sistema Bicycle Rentals
     */
    public List<Bike> getBikes();
}

```

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	

Anexo 2

```

public interface UserInterface {

    /**
     * Cálculo do custo do aluguer com base no tarifário de aluguer.
     * O tempo é medido em unidades de tempo representados por números inteiros
     *
     * @param rentalProgram tarifário de aluguer, 1 ou 2
     * @param initTime - início do aluguer
     * @param endTime - fim do aluguer
     * @param nRentals - número de alugueres já feitos pelo utilizador
     * @return valor a pagar pelo aluguer com as características definidas
     */
    public int bicycleRentalFee(int rentalProgram, int initTime, int endTime, int
nRentals);

    /**
     * Retirar uma bicicleta do depósito e iniciar o aluguer.
     * Requer que o utilizador tenha crédito positivo
     *
     * @param IDDeposit - id do depósito
     * @param IDUser - id do utilizador
     * @param startTime - unidade de tempo em que a bicicleta é levantada
     * @return id da bicicleta levantada
     * @throws UserDoesNotExists - exceção se o utilizador não existir
     */
    public int getBicycle(int IDDeposit, int IDUser, int startTime) throws
UserDoesNotExists;

    /**
     * Faz o retorno da bicicleta ao depósito e termina o aluguer.
     *
     * @param IDDeposit - id do depósito
     * @param IDUser - id do utilizador
     * @param endTime - unidade de tempo em que a bicicleta é devolvida
     * @return id do lock onde a bicicleta foi devolvida
     */
    public int returnBicycle(int IDDeposit, int IDUser, int endTime);

    /**
     * Verifica se o saldo da conta de um utilizador é positivo
     *
     * @param IDUser - id do utilizador
     * @return booleano indicando se existe crédito positivo na conta do utilizador
     */
    public boolean verifyCredit(int IDUser);
}

```

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova	Ano letivo 2018/2019	Data
	Curso Licenciatura Engenharia Informática	Hora	
	Unidade Curricular Engenharia de Software 2	Duração	

Anexo 3

```
public class BikeRentalSystem implements AdministratorInterface, UserInterface {

    public BikeRentalSystem(int rentalFee) {
        this.rentalFee = rentalFee;

        this.bikes = new ArrayList<>();
        this.deposits = new ArrayList<>();
        this.users = new ArrayList<>();
    }

    ...

}
```