

# Programação Orientada por Objetos

---

## **Desenho de aplicações - Exemplo**

Prof. José Cordeiro,

Prof. Cédric Grueau,

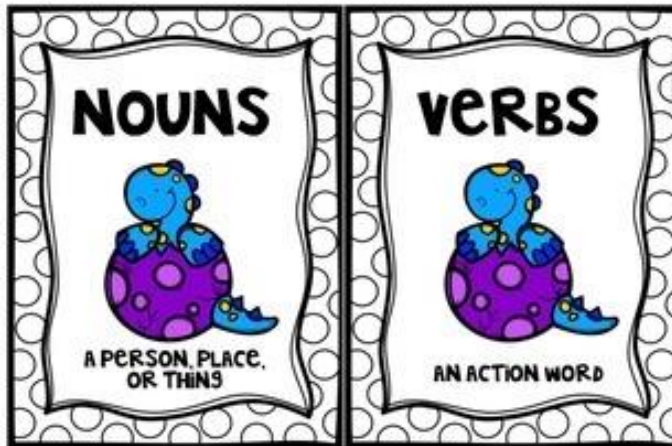
Prof. Laercio Júnior

Departamento de Sistemas e Informática

Escola Superior de Tecnologia de Setúbal – Instituto Politécnico de Setúbal

2019/2020

- Sessão 5: Exemplo Companhia de Táxis
- Sessão 6: Esboço da implementação
- Sessão 7: Desenvolvimento Iterativo



Módulo 4 – Desenho de aplicações

## SESSÃO 5 – EXEMPLO COMPANHIA DE TÁXIS

# Exemplo — Companhia de táxis

## □ Aplicação para uma companhia de táxis

- Criação de um modelo para uma companhia de táxis
- A companhia opera táxis e pequenos autocarros e planeia a sua expansão.
- Será que a expansão é rentável?
- Quantos veículos serão necessários?



# Exemplo — Companhia de táxis

## □ Descrição do problema

- A empresa opera táxis individuais e pequenos autocarros (shuttles).
- Os táxis são utilizados para o transporte de um indivíduo (ou pequeno grupo) a partir de um local para outro.
- Os autocarros são usados para recolher indivíduos em diferentes locais e levá-los para os seus vários destinos.
- Quando a empresa recebe um telefonema de um indivíduo, hotel, local de entretenimento, ou organização turística, tenta agendar um veículo para ficar com essa tarefa.
- Se não tiver veículos livres, não existe qualquer tipo de lista de espera.
- Quando um veículo chega a um local de recolha, o motorista notifica a empresa.
- Da mesma forma, quando um passageiro é deixado no seu destino, o motorista notifica a empresa.

# Exemplo — Companhia de táxis

## ☐ Descrição do problema - adendas

- Na realidade existem dois problemas adicionais que deveriam ser tidos em conta:
  - ☐ Quantos potenciais clientes se perdem por não existirem veículos disponíveis?
  - ☐ Quantos veículos ficam parados por muito tempo?
- Embora não se vá modelar este problema ficam aqui os detalhes adicionais que faltaria adicionar como exercício complementar:
  - ☐ Guardar os detalhes das viagens que não foram feitas
  - ☐ Guardar os detalhes da forma como é utilizado o tempo de cada veículo.

# Exemplo — Companhia de táxis

## □ Quais são as **classes da aplicação**?

- A **empresa** opera **táxis** individuais e pequenos **autocarros** (shuttles).
- Os **táxis** são utilizados para o transporte de um **indivíduo** (ou pequeno grupo) a partir de um **local** para outro.
- Os **autocarros** são usados para recolher **indivíduos** em diferentes **locais** e levá-los para os seus vários **destinos**.
- Quando a empresa recebe um telefonema de um **indivíduo**, **hotel**, **local de entretenimento**, ou **organização turística**, tenta agendar um **veículo** para ficar com essa **tarefa**.
- Se não tiver **veículos** livres, não existe qualquer tipo de lista de espera.
- Quando um **veículo** chega a um **local de recolha**, o **motorista** notifica a **empresa**.
- Da mesma forma, quando um **passageiro** é deixado ao seu **destino**, o **motorista** notifica a **empresa**.

## □ **Nomes Singulares**: empresa, táxi, autocarro, indivíduo, local, destino, hotel, local de entretenimento, organização turística, veículo, local de recolha, motorista, passageiro.

- Nem todos vão corresponder diretamente a classes no sistema

## □ Identificar **sinónimos**

- Por exemplo: Indivíduo e passageiro significam o mesmo.

## □ Eliminar **detalhes supérfluos**

- Por exemplo: será necessária a distinção entre hotel, local de entretenimento e organização turística?

# Exemplo — Companhia de táxis

## □ Nomes e verbos

- A **empresa** opera **táxis** individuais e pequenos **autocarros** (shuttles).
- Os **táxis** são utilizados para o transporte de um **indivíduo** (ou pequeno grupo) a partir de um **local** para outro.
- Os **autocarros** são usados para recolher **indivíduos** em diferentes **locais** e levá-los para os seus vários **destinos**.
- Quando a empresa recebe um telefonema de um **indivíduo**, **hotel**, **local de entretenimento**, ou **organização turística**, tenta agendar um **veículo** para ficar com essa **tarefa**.
- Se não tiver **veículos** livres, não existe qualquer tipo de lista de espera.
- Quando um **veículo** chega a um **local de recolha**, o **motorista** notifica a **empresa**.
- Da mesma forma, quando um **passageiro** é deixado ao seu **destino**, o **motorista** notifica a **empresa**.

### Empresa

*Opera taxis.  
Recebe telefonemas.  
Agenda veículos.*

### Veículo

*Recolhe indivíduos.  
Chega a um local de recolha.  
Notifica a empresa da chegada.  
Notifica a empresa da chegada ao destino.*

### Autocarro

*Transporta um ou mais passageiros.*

### Taxi

*Transporta passageiros.*

### Local

### Passageiro

### Origem dos passageiros

*Telefona à empresa.*





Módulo 4 – Desenho de aplicações

## **SESSÃO 6 – ESBOÇO DA IMPLEMENTAÇÃO**

# Exemplo — Companhia de táxis

## □ Cenários

- Desde o pedido inicial até à largada do passageiro
- Utilizam-se as **cartas CRC** com as classes anteriormente identificadas.

### PassengerSource

Criar um passageiro.  
pedir um taxi.  
Gerar o local de recolha  
e o destino.

### Colaboradores

Passenger  
TaxiCompany

# Exemplo — Companhia de táxis

## □ Desenho da **interface das classes**

```
public class PassengerSource {  
  
    /**  
     * Have the source generate a new passenger and  
     * request a pickup from the company.  
     * @return true If the request succeeds,  
     *         false otherwise.  
     */  
    public boolean requestPickup()  
  
    /**  
     * Create a new passenger.  
     * @return The created passenger.  
     */  
    private Passenger createPassenger()  
}
```

# Exemplo — Companhia de táxis

## □ **Esboço da implementação**

- Depois da implementação das interfaces pode-se criar um esboço da implementação.
- O esboço da implementação serve para testar a adequação das interfaces.
  - É de esperar que surjam correções no design.
- Escrever alguns testes básicos que serão repetidos ao longo do desenvolvimento.

# Exemplo — Companhia de táxis

## □ Esboço da implementação – Exemplo da classe **Vehicle**

```
public abstract class Vehicle {

    private TaxiCompany company;
    // Where the vehicle is.
    private Location location;
    // Where the vehicle is headed.
    private Location targetLocation;

    /**
     * Constructor of class Vehicle
     * @param company The taxi company. Must not be null.
     * @param location The vehicle's starting point. Must not be null.
     * @throws NullPointerException If company or location is null.
     */
    public Vehicle(TaxiCompany company, Location location) {
        if(company == null) {
            throw new NullPointerException("company");
        }
        if(location == null) {
            throw new NullPointerException("location");
        }
        this.company = company;
        this.location = location;
        targetLocation = null;
    }
    // Continua...
```

# Exemplo — Companhia de táxis

## □ Esboço da implementação

```
/**
 * Notify the company of our arrival at a pickup location.
 */
public void notifyPickupArrival() {
    company.arrivedAtPickup(this);
}

/**
 * Notify the company of our arrival at a
 * passenger's destination.
 * @param passenger The passenger who has arrived.
 */
public void notifyPassengerArrival(Passenger passenger) {
    company.arrivedAtDestination(this, passenger);
}

/**
 * Receive a pickup location.
 * How this is handled depends on the type of vehicle.
 * @param location The pickup location.
 */
public abstract void setPickupLocation(Location location);
```

# Exemplo — Companhia de táxis

## □ Esboço da implementação

```
/**
 * Receive a passenger.
 * How this is handled depends on the type of vehicle.
 * @param passenger The passenger being picked up.
 */
public abstract void pickup(Passenger passenger);

/**
 * Is the vehicle free?
 * @return Whether or not this vehicle is free.
 */
public abstract boolean isFree();

/**
 * Offload any passengers whose destination is the
 * current location.
 */
public abstract void offloadPassenger();

/**
 * Get the location.
 * @return Where this vehicle is currently located.
 */
public Location getLocation() {
    return location;
}
```

# Exemplo — Companhia de táxis

## □ Esboço da implementação

```
/**
 * Set the current location.
 * @param location Where it is. Must not be null.
 * @throws NullPointerException If location is null.
 */
public void setLocation(Location location) {
    if(location != null) {
        this.location = location;
    }
    else {
        throw new NullPointerException();
    }
}

/**
 * Get the target location.
 * @return Where this vehicle is currently headed, or null
 *         if it is idle.
 */
public Location getTargetLocation() {
    return targetLocation;
}
```



## □ Esboço da implementação

```
/**
 * Set the required target location.
 * @param location Where to go. Must not be null.
 * @throws NullPointerException If location is null.
 */
public void setTargetLocation(Location location)
{
    if(location != null) {
        targetLocation = location;
    }
    else {
        throw new NullPointerException();
    }
}

/**
 * Clear the target location.
 */
public void clearTargetLocation()
{
    targetLocation = null;
}
}
```

## □ Testes – exemplo classe Taxi

```
public class TaxiTest {
    private Taxi taxi;

    /**
     * Default constructor for test class TaxiTest
     */
    public TaxiTest() {
    }

    /**
     * Create a taxi.
     */
    @Before
    public void setUp() {
        TaxiCompany company = new TaxiCompany();
        Location taxiLocation = new Location();
        taxi = new Taxi(company, taxiLocation);
    }

    /**
     * Tears down the test fixture.
     */
    @After
    public void tearDown() {
    }
    // Continua...
```

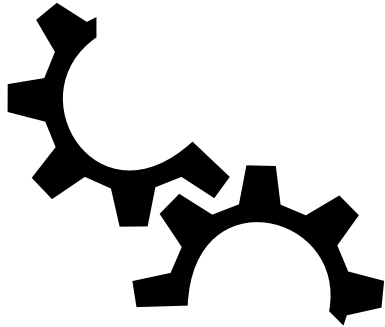
# Exemplo – Companhia de táxis

## □ Testes – exemplo classe Taxi

```
/** Test creation and the initial state of a taxi.
 */
@Test
public void testCreation() {
    assertEquals(true, taxi.isFree());
}

/** Test that a taxi is no longer free after it has picked up a passenger.
 */
@Test
public void testPickup() {
    Location pickup = new Location();
    Location destination = new Location();
    Passenger passenger = new Passenger(pickup, destination);
    taxi.pickup(passenger);
    assertEquals(false, taxi.isFree());
}

/** Test that a taxi becomes free again after offloading a passenger.
 */
@Test
public void testOffload() {
    testPickup();
    taxi.offloadPassenger();
    assertEquals(true, taxi.isFree());
}
} // Fim classe TaxiTest
```



Módulo 4 – Desenho de aplicações

## **SESSÃO 7 – DESENVOLVIMENTO ITERATIVO**

# Exemplo — Companhia de táxis

- ❑ Desenvolvimento **Iterativo**
- ❑ Ir dando pequenos passos até à conclusão da aplicação.
- ❑ Marcar o fim de cada passo com um período de testes:
  - Testes regressivos.
  - Correção antecipada de erros.
  - Rever as decisões de design anteriores se for necessário.
  - Tratar os erros encontrados como sucessos.

# Exemplo — Companhia de táxis

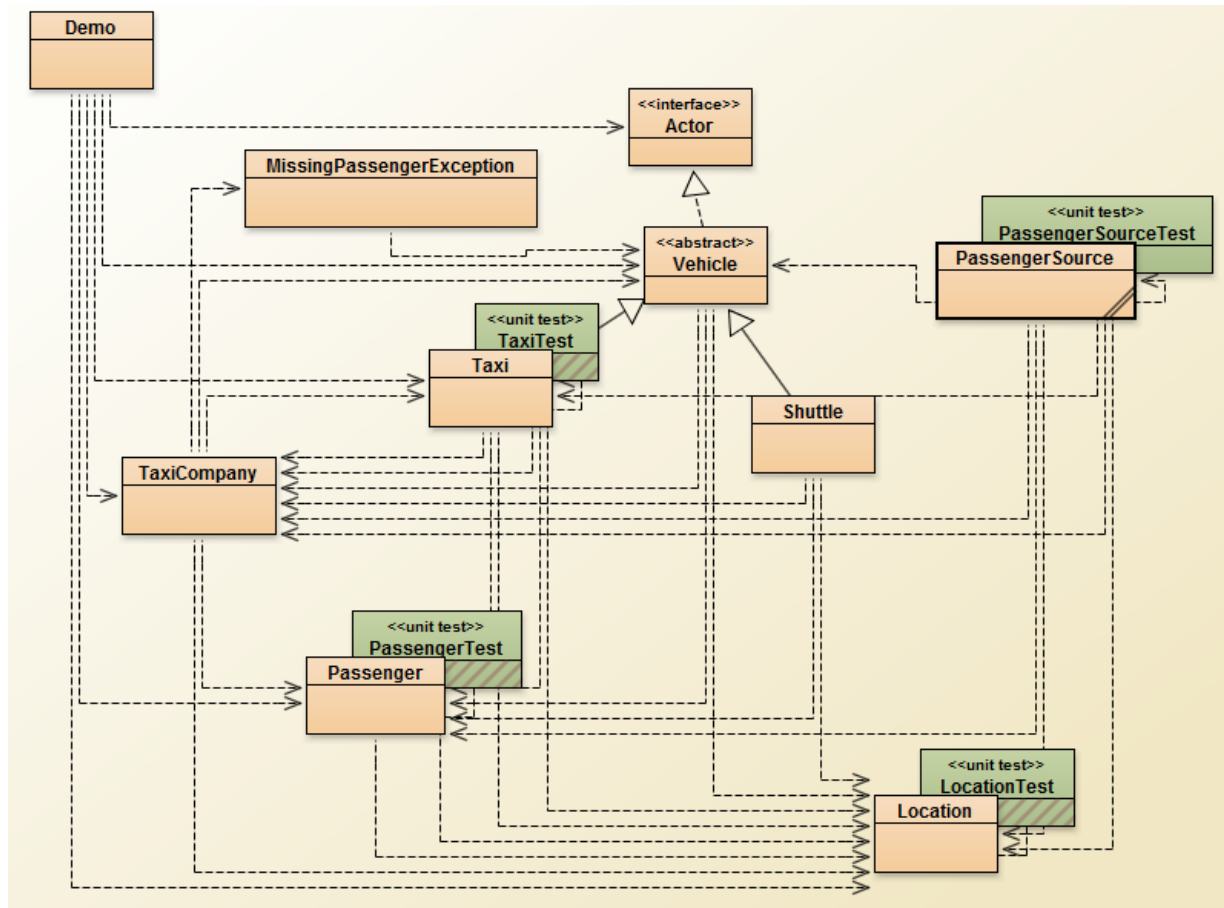
## □ Desenvolvimento **Iterativo**

### ■ Exemplo dos pequenos passos até à conclusão da aplicação.

- Possibilitar um único passageiro ser recolhido e levado ao seu destino por um único táxi.
- Providenciar vários táxis para recolha de passageiros independentes que são transportados aos seus destinos simultaneamente.
- Assegurar que são registados os detalhes dos telefonemas para os quais não existiam táxis disponíveis.
- Permitir que um único autocarro recolha diferentes passageiros transportando-os para os seus vários destinos.
- Criar a interface que visualiza todos os veículos e passageiros durante a simulação.
- Assegurar que os táxis e os autocarros operam em simultâneo.
- Assegurar as restantes funcionalidades, incluindo todos os dados estatísticos.

# Exemplo – Companhia de táxis

- Desenvolvimento **Iterativo** – primeira etapa
  - Possibilitar um único passageiro ser recolhido e levado ao seu destino por um único táxi.



# Exemplo – Companhia de táxis

## □ Primeira etapa – exemplo classe Taxi

```
/**
 * A taxi is able to carry a single passenger.
 *
 * @author David J. Barnes and Michael Kölling
 * @version 2011.07.31
 */
public class Taxi extends Vehicle {
    private Passenger passenger;

    /**
     * Constructor for objects of class Taxi
     * @param company The taxi company. Must not be null.
     * @param location The vehicle's starting point. Must not be null.
     * @throws NullPointerException If company or location is null.
     */
    public Taxi(TaxiCompany company, Location location) {
        super(company, location);
    }

    // continua...
```



# Exemplo — Companhia de táxis

## □ Primeira etapa – exemplo classe Taxi

```
/**
 * Carry out a taxi's actions.
 */
public void act() {
    Location target = getTargetLocation();
    if(target != null) {
        // Find where to move to next.
        Location next = getLocation().nextLocation(target);
        setLocation(next);
        if(next.equals(target)) {
            if(passenger != null) {
                notifyPassengerArrival(passenger);
                offloadPassenger();
            }
            else {
                notifyPickupArrival();
            }
        }
    }
    else {
        incrementIdleCount();
    }
}
```

# Exemplo — Companhia de táxis

## □ Primeira etapa – exemplo classe Taxi

```
/**
 * Is the taxi free?
 * @return Whether or not this taxi is free.
 */
public boolean isFree() {
    return getTargetLocation() == null && passenger == null;
}

/**
 * Receive a pickup location. This becomes the target location.
 * @param location The pickup location.
 */
public void setPickupLocation(Location location) {
    setTargetLocation(location);
}

/**
 * Receive a passenger. Set their destination as the target location.
 * @param passenger The passenger.
 */
public void pickup(Passenger passenger) {
    this.passenger = passenger;
    setTargetLocation(passenger.getDestination());
}
```

# Exemplo — Companhia de táxis

## □ Primeira etapa – exemplo classe **Taxi**

```
/**
 * Offload the passenger.
 */
public void offloadPassenger()
{
    passenger = null;
    clearTargetLocation();
}

/**
 * Return details of the taxi, such as where it is.
 * @return A string representation of the taxi.
 */
public String toString()
{
    return "Taxi at " + getLocation();
}
}
```

# Exemplo — Companhia de táxis

## ❑ Testar a primeira etapa – exemplo classe TaxiTest

```
/**
 * The test class TaxiTest.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class TaxiTest {
    private Taxi taxi;
    private Passenger passenger;

    /**
     * Default constructor for test class TaxiTest
     */
    public TaxiTest() {
    }

    // Continua...
```

# Exemplo — Companhia de táxis

## ❑ Testar a primeira etapa – exemplo classe TaxiTest

```
/**
 * Create a taxi.
 *
 * Called before every test case method.
 */
@Before
public void setUp() {
    TaxiCompany company = new TaxiCompany();
    // Starting position for the taxi.
    Location taxiLocation = new Location(0, 0);
    // Locations for the passenger.
    Location pickup = new Location(1, 2);
    Location destination = new Location(5, 6);

    passenger = new Passenger(pickup, destination);
    taxi = new Taxi(company, taxiLocation);
}

/**
 * Tears down the test fixture.
 *
 * Called after every test case method.
 */
@After
public void tearDown() {
}
```

# Exemplo — Companhia de táxis

## ❑ Testar a primeira etapa – exemplo classe TaxiTest

```
/**
 * Test creation and the initial state of a taxi.
 */
@Test
public void testCreation() {
    assertEquals(true, taxi.isFree());
}

/**
 * Test that a taxi is no longer free after it has picked up a passenger.
 */
@Test
public void testPickup() {
    taxi.pickup(passenger);
    assertEquals(false, taxi.isFree());
}

/**
 * Test that a taxi becomes free again after offloading a passenger.
 */
public void testOffload() {
    taxi.pickup(passenger);
    assertEquals(false, taxi.isFree());
    taxi.offloadPassenger();
    assertEquals(true, taxi.isFree());
}
```

# Exemplo — Companhia de táxis

## ❑ Testar a primeira etapa – exemplo classe TaxiTest

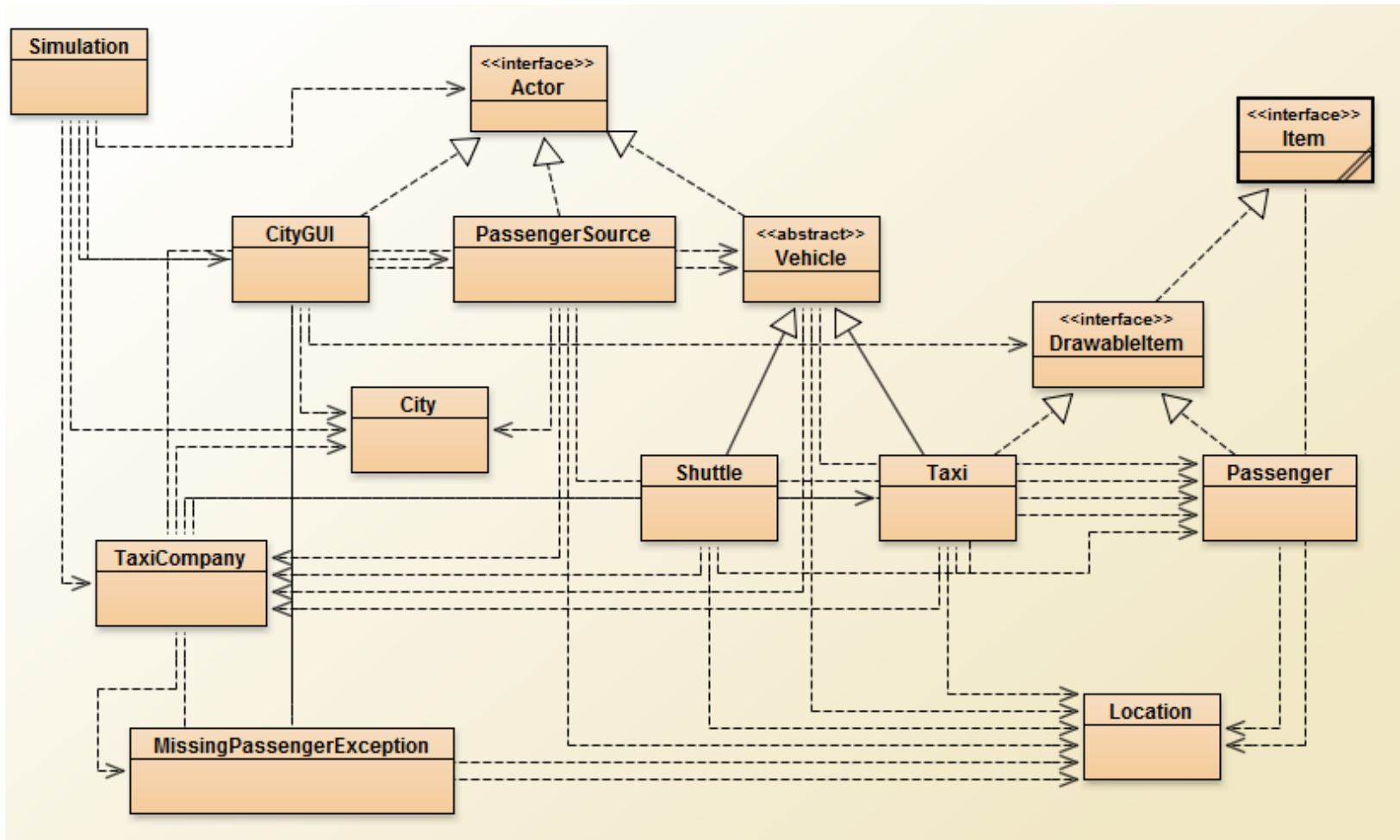
```
/**
 * Test that a taxi picks up and delivers a passenger within
 * a reasonable number of steps.
 */
public void testDelivery()
{
    Location pickupLocation = passenger.getPickupLocation();
    Location destination = passenger.getDestination();
    // The number of steps expected is the sum of the taxi's
    // distance to the passenger and the distance from there
    // to the destination.
    int stepsExpected = taxi.getLocation().distance(pickupLocation) +
        pickupLocation.distance(destination);

    taxi.pickup(passenger);

    int steps = 0;
    while(!taxi.isFree() && steps < stepsExpected) {
        taxi.act();
        steps++;
    }
    assertEquals(steps, stepsExpected);
    assertEquals(taxi.isFree(), true);
}
} // Fim da classe TaxiTest
```

# Exemplo — Companhia de táxis

- Desenvolvimento **Iterativo** — outra etapa mais tarde

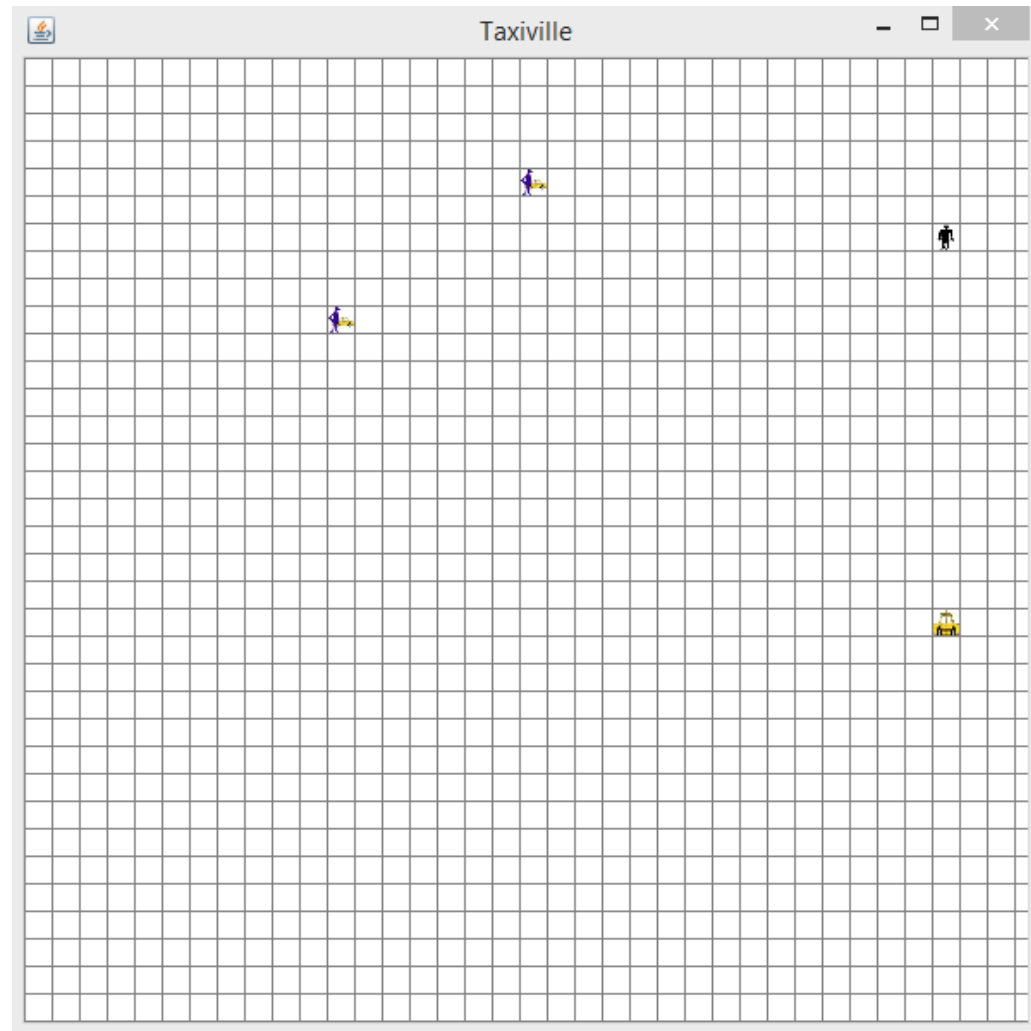




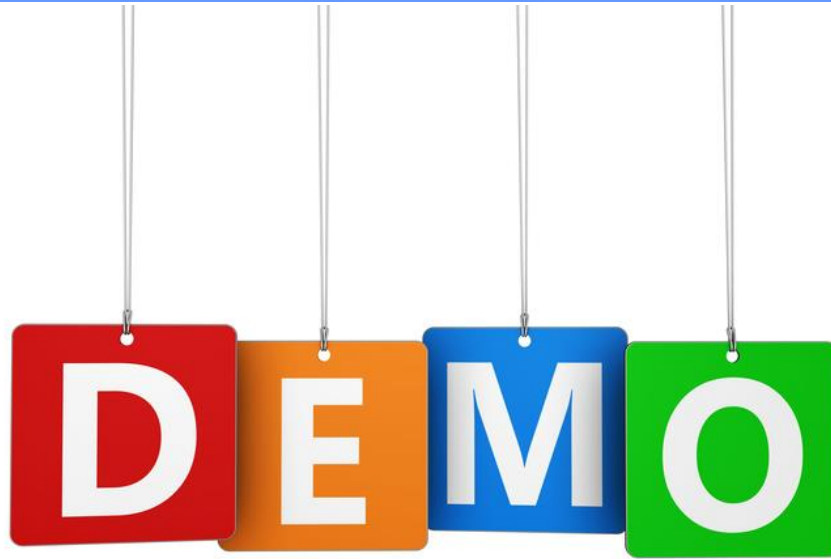
# Exemplo — Companhia de táxis

□ Desenvolvimento **Iterativo** — outra etapa mais tarde

■ Interface gráfica



# Exemplo — Simulação da Operação de uma Companhia de Táxis



## ☐ Revisão

- Para a construção de software robusto é necessário seguir processos bem pensados:
  - ☐ Analisar cuidadosamente.
  - ☐ Especificar claramente.
  - ☐ Projetar completamente.
  - ☐ Implementar e testar incrementalmente.
  - ☐ Rever, rever e aprender. Ninguém é perfeito!

# Bibliografia

- Objects First with Java (6th Edition), David Barnes & Michael Kölling, Pearson Education Limited, 2016
  - Capítulo 16

