ESAME DI FONDAMENTI DI INFORMATICA T-2 del 12/09/2018
Proff. E. Denti – R. Calegari – G. Zannoni Tempo: 4 ore

NOME PROGETTO ECLIPSE: CognomeNome-matricola (es. RossiMario-0000123456)
NOME CARTELLA PROGETTO: CognomeNome-matricola (es. RossiMario-0000123456)
NOME ZIP DA CONSEGNARE: CognomeNome-matricola.zip (es. RossiMario-0000123456.zip)

NB: l'archivio ZIP da consegnare deve contenere l'intero progetto Eclipse

La società *Rent-a-Bike* ha richiesto lo sviluppo del software necessario a gestire il noleggio bici in diverse città.

DESCRIZIONE DEL DOMINIO DEL PROBLEMA

Il noleggio bici è una forma di mobilità ecosostenibile sempre più diffusa nelle grandi città, specialmente nelle forme che permettono la localizzazione, il prelievo e il rilascio delle bici ovunque, senza doversi recare in stazioni predefinite, grazie al GPS di cui ogni bici è dotata, unitamente a un'app per smartphone che ne consente la gestione.

La *tariffazione è a tempo*, secondo uno schema fisso, ma i cui costi variano da una città all'altra, che prevede:

- un primo periodo (solitamente di 20-30 minuti), per i noleggi brevi
- successivi periodi (solitamente di 30 minuti ciascuno) per i noleggi più lunghi
- una durata massima (solitamente 12-24 ore) o un orario massimo (solitamente mezzanotte) entro cui la bici va riconsegnata, pena il pagamento di multe salate.

Una ***tariffa*** è definita quindi per una *specifica città*, ed è caratterizzata da:

- il nome della città a cui si applica;
- il costo (in €cent) e la durata (in minuti) del primo periodo;
- il costo (in €cent) e la durata (in minuti) dei periodi successivi;
- la durata massima (in ore) OPPURE l'orario limite del noleggio.

Un noleggio che superi la durata massima o l'orario limite si dice *irregolare* e verrà sanzionato.

Un ***noleggio*** inizia quindi in un dato momento (giorno e ora) e termina in un altro momento (giorno e ora): ad esso corrisponde un ***costo in euro*** calcolato come segue:

1. si calcola la durata del noleggio
2. se ne verifica la regolarità (durata non superiore al massimo o orario limite rispettato)
3. se il noleggio è regolare, se ne calcola il costo applicando la tariffa, a scatti di durata pari ai periodi previsti in quella città (i minuti eventualmente non goduti nell'ultimo periodo vanno persi): altrimenti, si aggiunge anche la sanzione.

ESEMPI DI TARIFFE

Bologna, 59 €cent per i primi 20', poi 99 €cent ogni 30'; max 12 ore; sanzione € 7
Reggio Emilia, 30 €cent per i primi 30', poi 50 €cent ogni 30'; max entro 23:59; sanzione € 7

ESEMPI DI NOLEGGI

BO, dalle 7.20 alle 7.45 → durata 25' (noleggio regolare) → € 0.59 + 0.99 = € 1.58

BO, dalle 8.25 alle 8.44 → durata 19' (noleggio regolare) → € 0.59

RE, dalle 7.30 alle 18.40 → durata 11h10 (noleggio regolare) → 23 periodi di 30' → € 11.30

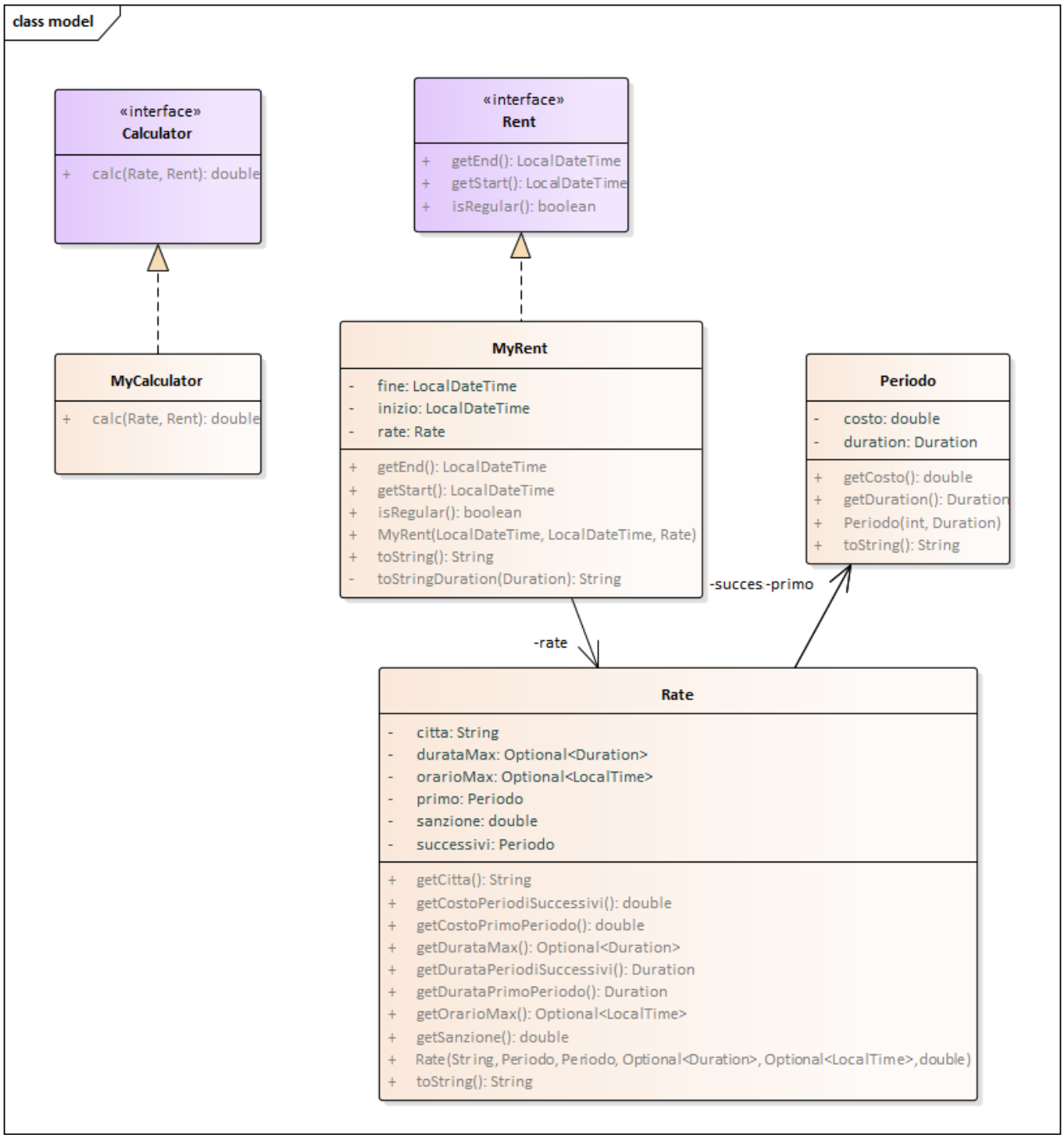
BO, dalle 7.30 alle 19.40 → durata 12h10 (noleggio irregolare) → 1 periodo di 20' + 24 periodi di 30' + multa → € 31.85

Il file di testo CityRates.txt specifica le tariffe delle varie città, nel formato dettagliato più oltre.

Parte 1 (punti: 19)

Dati (namespace rentabike.model) *(punti: 9)*

Il modello dei dati deve essere organizzato secondo il diagramma UML di seguito riportato:



SEMANTICA:

a) la classe ***Periodo*** (fornita) rappresenta un periodo, caratterizzato da costo (cent) e durata (in minuti).

b) la classe ***Rate*** (fornita) espone i metodi che caratterizzano le proprietà di una tariffa: nome città, proprietà del primo periodo (costo e durata), proprietà dei periodi successivi (costo e durata), eventuale durata max e orario di rientro limite (argomenti ***Optional***), ammontare della sanzione.

c) l'interfaccia ***Rent*** (fornita) rappresenta un noleggio con le sue caratteristiche (inizio, fine, tariffa applicabile). Il metodo ***isIrregular*** consente di discriminare i noleggi regolari da quelli irregolari, mentre gli accessor recuperano il momento iniziale e finale del noleggio.

d) la classe **MyRent (da realizzare)** implementa ***Rent*** opportunamente: il costruttore riceve i dati nel formato specificato dal diagramma UML.

e) l'interfaccia ***Calculator*** (fornita) dichiara il metodo ***calc*** che effettua il calcolo del costo di un noleggio (***Rent***) ricevuto come secondo argomento sulla base della ***Rate*** ricevuta come primo argomento;

f) la classe **MyCalculator (da realizzare)** concretizza ***Calculator*** implementando ***calc*** coerentemente a quanto specificato nel Dominio del problema.

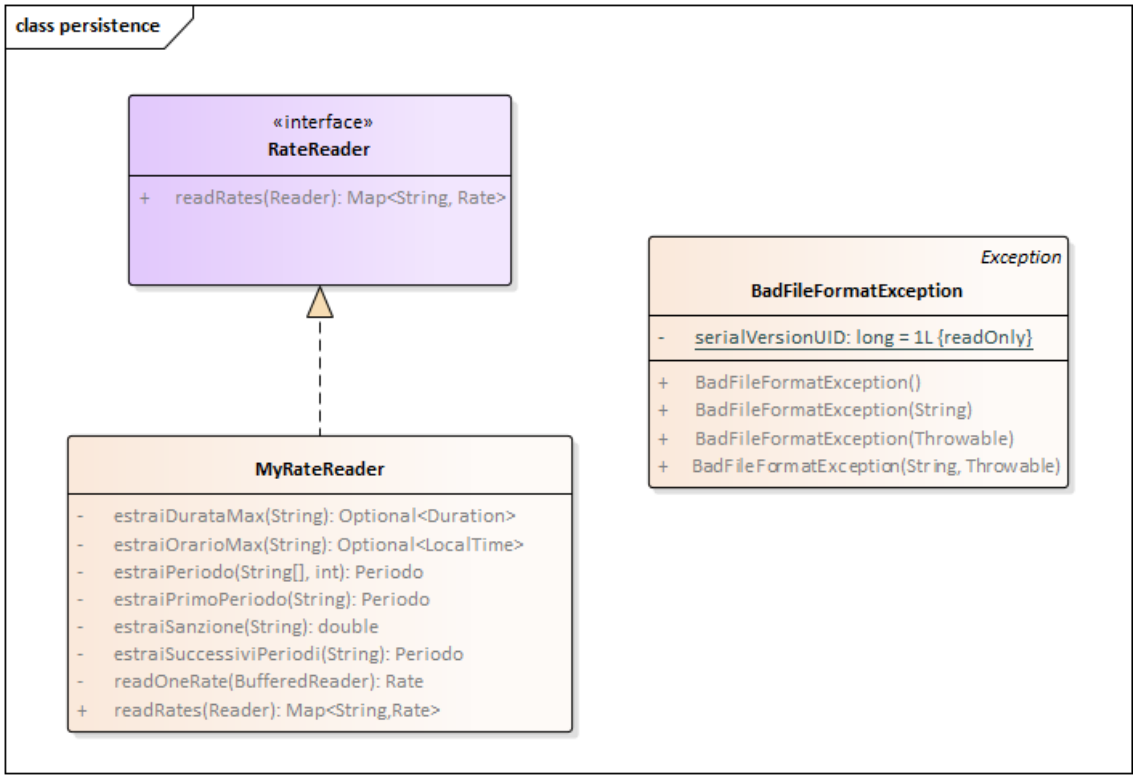
Persistenza (namespace rentabike.persistence) *(punti: 10)*

Come anticipato, il file CityRates.txt specifica le tariffe delle varie città, una per riga. Ogni riga contiene una serie di elementi separati da virgole, ulteriormente organizzati al proprio interno in sequenze di elementi separate da spazi. Più precisamente, nell'ordine si ha:

- nome della città (può contenere spazi)
- il costo e la durata del periodo iniziale
  - costo del periodo iniziale (valore intero seguito dalla parola chiave "**`cent`**")
  - la parola chiave "**`per`**"
  - un valore intero
  - la parola chiave "**`minuti`**"
- il costo e la durata dei periodi successivi
  - la parola chiave "**`poi`**"
  - costo (in €cent) del periodo (valore intero seguito dalla parola chiave "**`cent`**")
  - la parola chiave "**`per`**"
  - un valore intero
  - la parola chiave "**`minuti`**"
- la durata massima o l'orario di rientro massimo
  - la parola chiave "**`max`**"
  - **o** un numero intero, seguito dalla parola chiave "**`ore`**"
    oppure la parola chiave "**`entro`**" seguito da un orario nel formato classico "hh:mm"
- la sanzione
  - la parola chiave "**`sanzione`**"
  - costo (in €) della multa (valore reale seguito dalla parola chiave "**`euro`**")

ESEMPIO DEL FILE CityRates.txt

```
Bologna, 59 cent per 20 minuti, poi 99 cent per 30 minuti, max 12 ore, sanzione 7.50 euro
Reggio Emilia, 30 cent per 30 minuti, poi 50 cent per 30 minuti, max entro 23:59, sanzione 7 euro
```



L'interfaccia ***RateReader*** (fornita) dichiara il metodo ***readRates*** che, dato un ***Reader***, legge le tariffe dal file e le restituisce sotto forma di mappa ***Map<String, Rate>*** indicizzata per nome città.

La classe *MyRateReader* (da realizzare) implementa tale interfaccia effettuando i necessari controlli sul formato del file, lanciando ***BadFileFormatException*** (fornita) in caso di errori di formato con messaggio dettagliato sulla specifica parola chiave mancante o elemento errato, o propagando ***IOException*** in caso di errori di lettura con specifico messaggio d'errore.

Parte 2 (punti: 11)

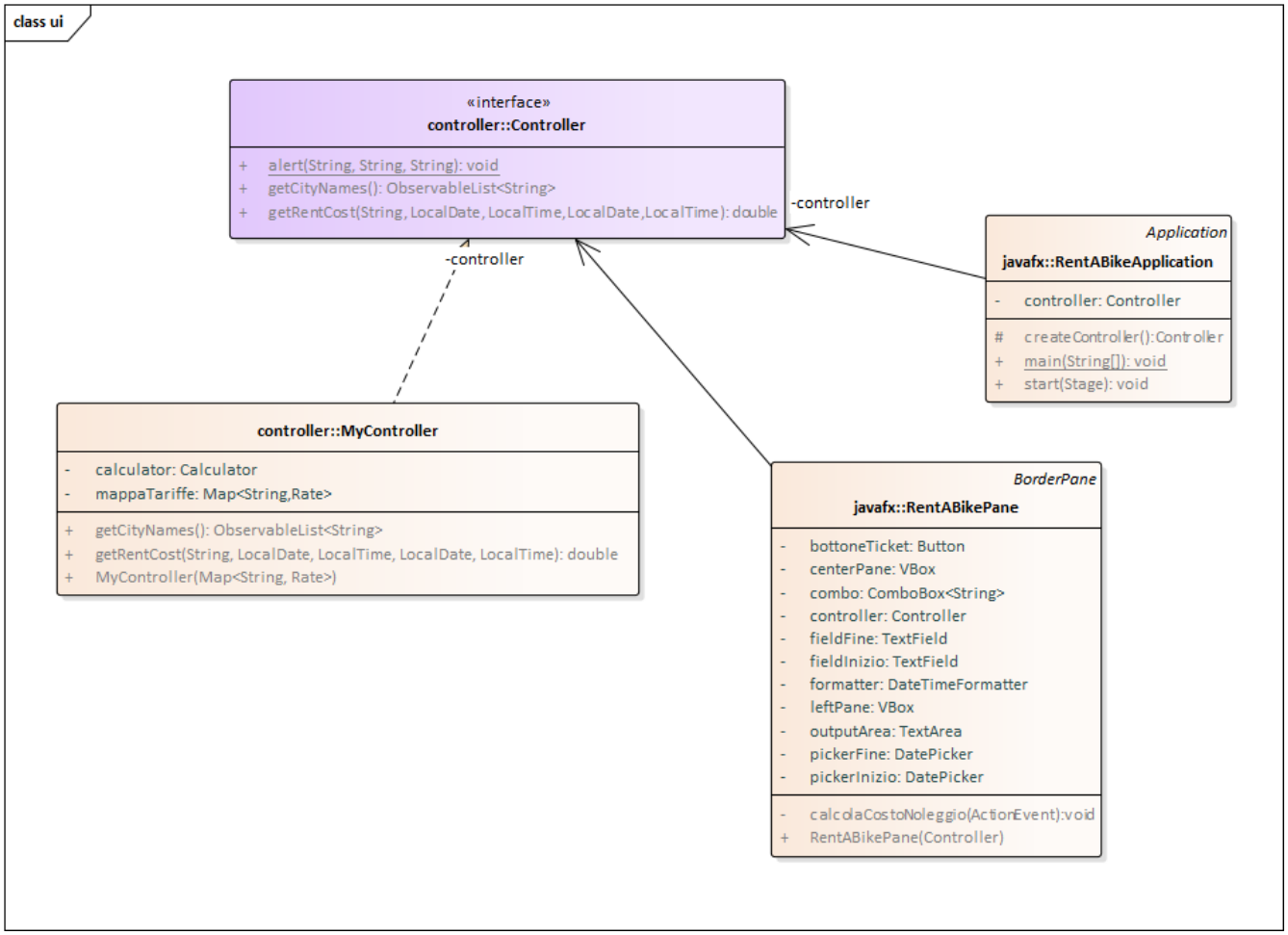
Controller (namespace rentabike.ui.controller) *(punti: 3)*

L'interfaccia ***Controller*** (fornita) dichiara il metodo ***getCityNames***, che restituisce la *lista osservabile* delle città per cui sono definite le tariffe, e il metodo ***getRentCost*** che calcola il costo del noleggio. Quest'ultimo riceve come argomenti:

- il nome della città;
- la data (un ***LocalDate***) e l'ora (un ***LocalTime***) di inizio sosta
- la data (un ***LocalDate***) e l'ora (un ***LocalTime***) di fine sosta

È fornito inoltre il metodo statico ***alert*** per far comparire una finestra di dialogo che segnali errori: i tre argomenti rappresentano il titolo della finestra, l'header e il testo del messaggio (Figg. 3 e 4).

La **classe *MyController* (da realizzare)** deve implementare ***Controller*** senza effettuare verifiche sugli argomenti, che si suppongono validati dal chiamante.



Interfaccia utente (namespace rentabike.ui.javafx) *(punti: 8)*

L'interfaccia utente deve essere simile (non necessariamente identica) all'esempio mostrato nelle figure seguenti.

La classe *BikeRentPane* (da realizzare), che estende ***BorderPane***, deve prevedere:

- in alto, una combo box per la scelta della città, con a fianco, un pulsante a sfondo rosso "*Calcola costo*" per calcolare il costo del noleggio;
- a sinistra, due ***DatePicker*** e due campi di testo, per inserire data e orario di inizio e fine noleggio;
- a destra, un'area di testo in cui mostrare il costo del noleggio, compreso dell'eventuale sanzione. **Il costo deve essere formattato in Euro tramite utilizzo esplicito del Locale.ITALY** (NB: Il simbolo della valuta può precedere o seguire l'importo, secondo le convenzioni della versione Java utilizzata).

È altresì compito del pannello effettuare un'accurata verifica dell'input prima di invocare il controller, in particolare:

- verificando che la città selezionata non sia nulla;
- verificando che la data di fine noleggio non sia antecedente la data di inizio noleggio;
- verificando che l'orario di fine noleggio non sia antecedente l'orario di inizio noleggio.

In tali casi l'applicazione deve mostrare opportuni dialoghi, tramite il metodo statico ***Controller.alert*** (Figg. 3 e 4).

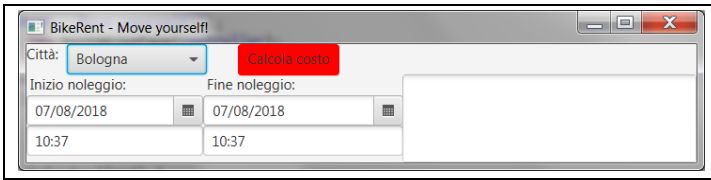
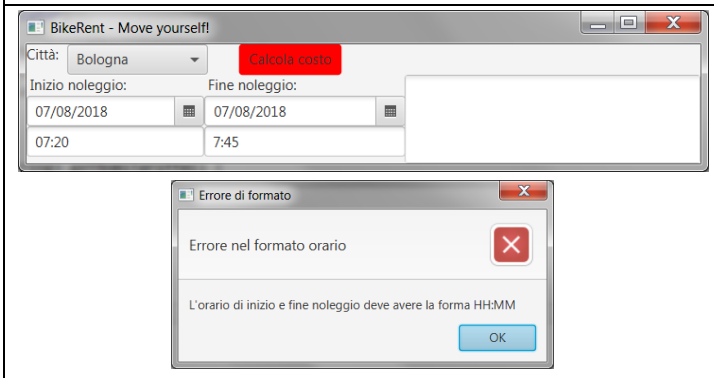


Figura 1

Figura 2



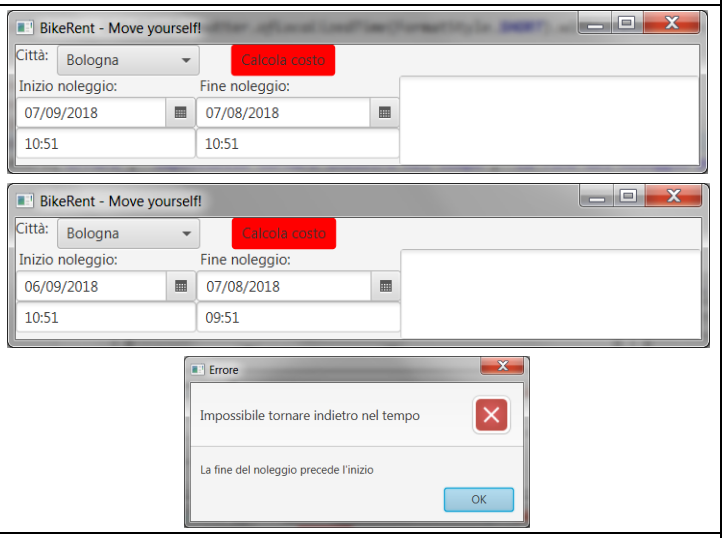


Figura 3

Figura 4