

Prova pratica del 17 giugno 2020

Ricordarsi di onsegnare il codice sorgente di tutte le classi e della classe main salvando il progetto Code::Blocks in Z:

Simulazione Sistema di consegna cibo

Si scriva un programma per simulare un sistema per la consegna del cibo, o **FoodDelivery**, in un periodo di emergenza sanitaria come quella che stiamo vivendo. Un negozio di FoodDelivery è caratterizzato da un nome, una posizione (caratterizzata da coordinate (x,y) in chilometri su una ipotetica mappa), un prezzo medio e un cibo tipico (memorizzato come stringa, es. "pizza", "pesce", ecc.). Devono essere inoltre previste tre varianti del metodo ordina per ordinare una consegna, che restituiscono se la consegna è possibile o no, e il tempo e prezzo della consegna. La differenza tra le tre varianti è nei parametri in input:

- La prima versione prevede solo la posizione dell'utente che effettua l'ordine
- La seconda (oltre alla posizione dell'utente) anche la quantità di cibo da ordinare
- La terza (oltre alla posizione e la quantità) anche il cibo richiesto dall'utente

NOTA: le tre versioni sono incremental, quindi una successiva si può ottenere dalla precedente.

Si vogliono modellare due tipologie di FoodDelivery. Il **Ristorante** è caratterizzato in aggiunta da un tempo medio di preparazione del cibo tipico (che varia da ristorante a ristorante e può essere memorizzato, ad esempio, con un int che indica i minuti) e da un parco di MezziDiConsegna (si veda dopo). La **Rosticceria** è caratterizzata da un prezzo di consegna (che nel caso del Ristorante non c'è, come in Rosticceria non c'è un tempo di preparazione perché sono cibi preparati in anticipo) e da parco di MezziDiConsegna.

La classe **MezziDiConsegna** è caratterizzata da un modello (nome del mezzo, stringa) e un valore che indica la sua attuale disponibilità o meno. Esistono due tipologie di MezziDiConsegna: **Auto** e **Motorino**. I **Motorini** hanno in aggiunta la caratteristica di numero massimo di porzioni trasportabile. Inoltre, sia Auto che Motorino hanno una velocità media (uguale per tutti gli oggetti della classe) che è di 60 Km/h per tutte le auto di 60 Km/h e di 30 Km/h per tutti i motorini. Infine, nel parco di mezzi dei Ristoranti ci sono solo Auto e nel parco di mezzi delle Rosticcerie ci sono solo i Motorini.

(SEGUE PROSSIMA PAGINA)

Il programma deve inizializzare una serie (non ne servono molti) di Ristoranti e Rosticcerie, creando anche un parco mezzi per ciascuno di essi. Successivamente, il programma deve chiedere iterativamente all'utente (fino ad un valore preciso che provoca l'uscita e la chiusura del programma) di scegliere tra queste opzioni:

- Prenotare il **primo cibo disponibile**, considerando una sola porzione e senza preferenza sul cibo, ma scegliendo quello che viene consegnato prima (indipendentemente da un Ristorante o da una Rosticceria);
- Prenotare un **certo numero di porzioni**, e scegliere il cibo che costa meno (indipendentemente da un Ristorante o da una Rosticceria e dal tempo di consegna);
- Prenotare un **certo numero di porzioni di uno specifico cibo** e scegliere il FoodDelivery che lo consegna prima (senza "preoccuparsi" del suo prezzo).

Il tempo di consegna si calcoli considerando una distanza Euclidea tra la posizione del Cliente e quella del FoodDelivery. Per il FoodDelivery scelto il mezzo scelto per la consegna deve essere disponibile. Se scelto deve essere prenotato (variabile disponibilità a *false*) all'inizio del metodo prenota e rilasciato (disponibilità a *true*) alla fine del metodo prenota.

Tutte le versioni del metodo devono visualizzare a schermo il nome del FoodDelivery scelto, il prezzo e il tempo di consegna.

Per il raggiungimento della lode, la gestione delle prenotazioni degli utenti deve avvenire mediante l'uso dei thread.

Sviluppare le classi che permettono una soluzione basata sulla programmazione Object Oriented al problema.