

## TRACCIA 1

Si vuole sviluppare una applicazione per la gestione delle offerte di una compagnia telefonica operante in Italia.

Si implementino le classi che modellano diversi tipi di utenze. Ai nostri scopi una utenza telefonica può essere una utenza fissa o una utenza mobile. Ogni utenza è caratterizzata da un codice (sia una stringa C allocata staticamente), un intestatario (sia una stringa C allocata dinamicamente e un numero telefonico (sia una stringa C allocata staticamente). Una utenza mobile è caratterizzata da profilo dell'utente che identifica l'insieme dei servizi attivati per quella utenza (L1, L2... sia un oggetto di tipo string). Una utenza fissa deve specificare la destinazione dell'immobile in cui si trova (appartamento, ufficio, ristorante, etc... sia una stringa C allocata dinamicamente) e il relativo indirizzo.

Per tali classi si implementino le funzionalità necessarie ad una corretta gestione dell'estensione dinamica, l'overloading dell'operatore <<, e una funzione membro che stabilisca se all'utenza è applicabile o meno una offerta. In particolare una offerta è applicabile ad una utenza fissa se è riferita ad un "appartamento" che si trova a "Napoli", mentre una offerta è applicabile ad una utenza mobile se il profilo è "L2" (si usino allo scopo costanti del relativo tipo, definite globali).

Per quanto riguarda l'indirizzo, si fornisca una opportuna classe le cui variabili membro rappresentino via, numero civico, città e CAP (tutte di tipo string) fornendo per tale classe *solo* costruttori, operatore << e funzioni get.

Sul numero di telefono di una utenza mobile deve essere possibile chiamare una funzione membro che controlli il formato. Il formato del numero mobile è valido se è una stringa alfanumerica costituita da più di 14 caratteri; il primo carattere deve essere '+', i successivi caratteri devono essere tutte cifre, di cui le prime due devono essere il prefisso per l'Italia (39), e inoltre -nell'ipotesi che le cifre riferibili all'operatore telefonico siano tre- dopo i primi 6 caratteri devono necessariamente aversi 7 cifre. Se anche una sola di tali condizioni non è verificata la funzione lancia una eccezione (sia un oggetto di una classe definita in modo da avere un messaggio diverso per ciascuno dei casi su specificati, così da fornire una informazione circa l'errore che ha provocato l'eccezione e il codice dell'utenza).

Si sviluppi un opportuno programma di test delle classi implementate che inoltre, data una sequenza di utenze, stabilisca a quali di esse è applicabile una offerta fornendo un esempio di late binding.

Si riporta un esempio di esecuzione:

*Utenti:*

*UF101 Vico Giambattista 0812225534 Via Mergellina 5 Napoli 80121 Appartamento*

*UF102 Morante Elsa 062121212 Via Margutta 10 Roma 00122 Appartamento*

*UM101 Foscato Ugo +393385577322 L2*

*UM102 Alighieri Dante +39338120032 L1*

*UM101: formato numero mobile valido*

*UM102: eccezione: numero di cifre dopo il prefisso operatore non valido*

*Utente: UF101 offerta applicabile*

*Utente: UM101 offerta applicabile*

*Utente: UF102 offerta NON applicabile*