

Universita' degli Studi di Catania

Corso di Laurea in Informatica

Esame di L.A.P.1

24/09/2013

Si progetti un package C++ per la simulazione di un sistema di comunicazione composto dalle seguenti classi:

- **ReceiverPeer**: rappresenta un “peer” di comunicazione in grado soltanto di ricevere dati.
 - Attributi:
 - **id**, intero, indirizzo del peer
 - Metodi:
 - costruttore, con specifica dell'id
 - overloading dell'operatore << per la stampa su “cout” (stampare l'id).
 - **void new_data_arrived(Peer * from, char * s)**, metodo di callback invocato nel momento in cui un dato, proveniente dal peer **from**, viene consegnato a questo peer. L'azione di questo metodo e' la stampa delle informazioni su dato ricevuto (stringa **s**), peer sender e peer receiver.
- **FullPeer**: sottoclasse di ReceiverPeer, rappresenta un “peer” di comunicazione in grado sia di ricevere che di trasmettere
 - Metodi:
 - **void send_data(Peer * dest, char * s)**, metodo invocato per inviare un certo dato al peer **dest**; tale metodo dovra' poi opportunamente invocare il metodo di callback **new_data_arrived()** del peer di destinazione.
- **Channel**: classe che rappresenta l'insieme di peer coinvolti nella comunicazione
 - Metodi:
 - overloading dell'operatore +=(**Peer**) per l'aggiunta di un nuovo peer al canale.
 - overloading dell'operatore << per la stampa su “cout” (stampare l'elenco dei peer).
 - metodo **void send(int id_sender, int id_receiver, char * data)** per l'invio di un dato dal peer “**id_sender**” al peer “**id_receiver**”. Tale metodo dovra' lanciare un'eccezione (opportunamente definita) nel caso in cui il sender selezionato non sia di tipo **FullPeer**.
 - Supporre che gli “id” dei peer siano univoci.

Scrivere un “main” che permetta di testare le classi create.

Requisiti:

- **Tempo: 1h30m**
- **Non e' consentito l'uso di libri o appunti**

Cognome _____

Nome _____

Matricola _____

Votazione _____