COMANDO -> rappresenta i comandi fondamentali di emotionalMaps, tipo (.gettipo) e argomento (.getargomento)

EMOTIONAL MAPS->

Utilità -> Vari metodi di uso comune nel programma (carica stringhe di file tramite buffer) , fau un controllo fino a quando bufferedReader è != da null, salvando in ArrayList, con messaggio ERRORE

ManagerComandi->

ManagerEventi->

StatoEmotivo-> stato gestito da sistema (NEUTRO, FELICE, TRISTE, ARRABBIATO, SORPRESO)

StatoRegistrazione-> Stato regiostrazione degli utenti gestiti da sistema (IN/OUT)

StatoUtente-> Stato di accesso utente gestito da sistema (LOGIN/LOGOUT)

Tipocomando-> Tipologia di comando gestito da sistema (importa/creamappa)

Evento->

ManagerPDI->

PuntodiInteresse->Estanziate le coordinate punti di interesse le emozioni totali (eventi totali) e le emozioni di utenti attivi. Viene generato un nuovo punto di interesse (nome e coordinata), instanziato a 0 i valori delle emozioni dei punti di interesse, creazione, dei relativi metodi get. Creazione del metodo .resetstatistiche Punti di interesse crezione metodo statisticheattive -> crea una stringa che contiene una percentuale di tutte le emozioni attive. .statistichetotali -> fa la stessa cosa del precedente ma per tutti gli utenti qualunque sia lo stato utente. .aggiungievento-> preleva l’evento e , incrementa eventitotali, controlla che iul tipo accesso sia uguale login se è login aggiorna eventitoaliuattivi quindi viene creato uno switch per gli syato emotivi che tramite il metodo ottiene lo stato e motivo dell’evento , in base allo stato emotivo viene incrementato il rispettivo contatore (totale) e viene eseguito un controllo, ricontrolla lo stato di accesso se è login viene incrementato il contatore per quello stato emotivo di utenti attivi, se lo stato emotivo non è tra quelli presenti viene stampato un messaggio WARNING specificando l’ID dell’evento.

Creazione calcola distanza controllando la distanza assoluta e viene richiamato il metodo della classe utilità che “ritorna” la distanza assoluta tra punto di interesse e coordinate passate .percentualetotale-> se gli eventi totali != ritorna il valore in int, arrotondato.

.toarray->

.bisestile-> verifica in boolean se l’anno che viene passato è bisestile se è cosi torna TRUE

.suddividestringhe-> viene passato come argomento l’arraylist, viene controllato se l’array è vuoto e crea una matrice salvando il risultato in posizione , prendendo la riga, finquando non trova la virgola .

.controllaarraystringhe-> viene passato l’array di tipo string, controllo se null o 0 , se è ok allora torna TRUE.

.aggiungiEstensione-> passa nomefile, se nonfinisce con estensione gliela aggiugne con “txt”

.distanza-> passano le coordinate di due punti, dove se la variabile x del primo punto è maggiore del secondo veiene creata una variabile temporanea con i valore di x. E vengono invertiti i valori infine trova la distanza assoluta.

ManagerPDI-> creazione classe manager “putni di interesse” , creazione lista di tipo puntodiinteresse, creazione del metodo managerPDI dove la lista viene instanziata. Come arraylist , tipo punto di interesse e veiene aggiunta tramite metodi il metodo .aggiungiPDI dei nuovi punti di interesse i cui attributi sono il nome del puntom la sua latitudine e longitudine per tutti i punti. Creazione metodo void chiamato elabora che ha come attributo l’evento e , all’interno vengono istanziati. Indice, indicePDI, distanza min e distanza calacolata, scansione arraylist e viene calcolata la distanza dal punto e ai vari punti di interesse e lo assegna al punto di interesse più vicino, e lo aggiunge ad arraylist se l’indice è uguale a -1 viene stampato un warning nell’elaborazione dell’evento. Creazzione di void.resetstatistiche, resetta le statistiche di tutti i punti di interesse . creazione void.stampastatistichetotali dove stampa.

-------------------

Eventi-> classe per la rappresentazione degli eventi, serve a gestire in modo più semplice i dati riguardanti l’evento. Creazione classe evento con all’interno le variabili, creazione elemento di tipo evento inizializzato con tutti i campi i valori dati in imput. Creazione metodo getaccesso facendo riferimento alla classe enum statistiche . creazione di getID che restituisce id utente, creaszione del metodo getX e getY che restituiscono le relative coordinate. Crezione getData che restituisce la data, creazione getstatoemotivo facendo riferimento alla classe enum statoemotivo.

---------------------------

Managereventi-> viene creata una classe managereventi e viene creato un HASHMAP implementata nelle api di java , la mappa è una collezione di oggetti il cui scopo principale è rendere veloce ed efficiente operaszioni di inserimento e ricerca elementi. La mappa memorizza coppie (chiave, valore). Come chiave ha delle stringhe mentre come valore ha un arraylist, chiamata vistaeventi. Creazione nuovo elemento di managerveventi dove al suo interno viene stanziato listaeventi. Crezione metodo void aggiungieventi di attributo nomefile, con l’arraylist settato a null e anche la stringa. Creazione di array di stringhe dove tramite il richiamo di un metodo della classe utilità viene caricata la stringa nel file. Viene verificato se l’array stringhe non e null e contestualmente viene controllato tramite il metodo .controllaarraystringhe se l’array e il suo contenuto è formalmente corretto. Successivamente tutti gli elementi dell’array vengono convertiti in tipo evento, se e evento è != da null viene controllata la data in listacorrente se è = a null oppure se e != dalla data dell’evento in questo caso datalistacorrtente prende il valore della data dell’evento. Imposta listacorrente con dataevento , controlla se = null, se è cosi viene instaziata listacorrente come arraylist di tipo evento e viene associata alla chiave datacorrente il valore di listacorrente, se è uguale a null viene aggiunto l’evento e all’aarylist di listacorrente, nel caso invece e= null, viene comunque aggiunto. Nel caso in cui l’array stringhe è = null oppure il metodo controllaarraystringhe restitusce false viene stampatoa schermo un warning indicando che non è stato possibile aggiungere evento alla lista, nel caso venga riscontrata un eccezione di tipo IOException viene stampato il medessimo warning. Creazione nuvo elemento di tipo reseteventi che ristanzia la listaeventi come hashmap, viene creato il metodo converti che è un array di tipo evento che ha come input un array di stringhe chiamato datigrezzi.

Viene creato un arraylist di tipo evento chiamato lista convertito e vengono salvati in una matrice si stringhe, chiamata datisuddivisi , i dati grezzi separati tramite il metodo suddivistringhe e viene istanziato un indice uguale a 1. Se i dati suddivisi sono null torna null. Vengono salvati i datisuddivisi in un array di tipo dato e viene creato un booleano “corretto “impostato a true. Richiama registrazione, login ed emozione dai rispettivi enumerativi e li imposta a null. Ed effettua un controllo dove se la prima posizione dell’array è “IN” viene impostata come stato registazione IN , altimenti OUT ……. Se non viene verificato nessuno dei 2 casi imposta corretto a false. Secondo controllo è per login che deve essere o LOGIN O LOGOUT altrimenti FALSE. Viene effettuato un controllo per la correttezza sintattica per la 3 posizione altimenti MHANZ. Viene creato uno switch con i valori in 7 posizione per ogni emozione consetita nell’enum viene impostata l’emozione appropiata, se il valore non dovesse esserci corretto viene impostato a false

Se i dati[i] sono corretti viene aggiunto all’arraylist l’evento altimenti se corretto = false stampa un warning indicando l’indice . in ultimo ritorna la lista come array.

Viene creato il metodo ricerca con argomenti data1 e data 2 viene istanziata un arraylist chiamato eventi, viene effettuato un controllo se le date sono in formato corretto. Controlla se data1<data2, selo è la ricerca viene saltata e ritorna null. Recupero eventi compresi tra le date e vengono salvati sull’arraylist di tipo listam se la lista != null viene aggiunto ad eventi, tutti gli eventi contenuti nell’hashmap che hanno quella data, restituisce gli eventi trovati. Viene creato il metodo contieneventoid con argomento l’ID è una lista, e verifica che nella lista sia presente l’ID dell’evento specificato con ID

Viene creato il metodo controlladata, viene passato come argomento la data. Se la lunghezza dovesse essere maggioener di 8 se il formato della data non dovesse essere di soli numeri se l’anno di inizion fosse inferiore a 1970 o il formato della data nomn dpvessere essere valido restituisce false.

Viene creato un metodo boolean primadidata che controlla chje fdaya1 preceda data2. Viene creato un metdoo boolean chiamato stessadata che controlla che dfta 1 e data2 facciano rifeirmento alla stessa data.

Viene creato un metodo boolean dopodidata che controlla che controlla che data1 succeda data2

Creazione metodo creadatasuccessiva con argomento uno strign data viene generata una dfata successbvia a quella data in input se quella daya in input non è corretta restituisce null.

Manager comandi->