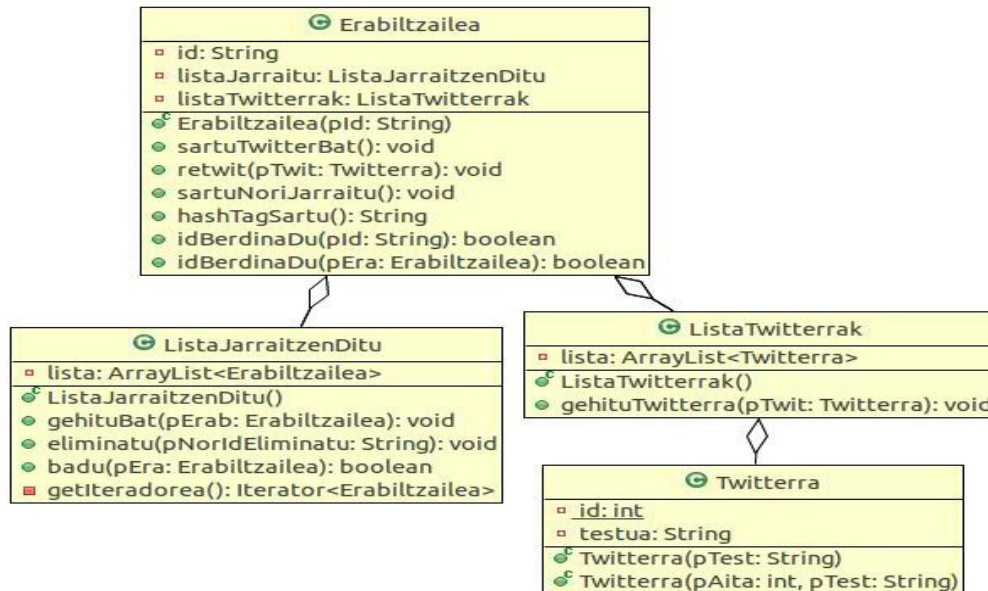


Twitter (1 puntu)

Twitterren, ezaguna den bezala, ezin dira 140 karaktere baino luzeagoak diren mezuak idatzi. Baina badaude hain ezagunak ez diren beste murriztapen batzuk. Adibidez, norbera ezin dela izan 1000 pertsona baino gehiagoen jarraitzailea. Murriztapen hauek salbuespen gisa tratatzea eskatzen zaigu, baita geroago azalduko diren beste egoera batzuk ere. Horretarako, jarraian agertzen den klase diagrama kontuan izanik, zein klaseetako metodoetan gertatzen diren salbuespen egoera horiek eta nola tratatu antzeman beharko duzue. Horrez gain, beharrezkoak diren salbuespen klaseak sortu beharko dituzue.



Salbuespen egoeren deskribapena:

- 1) Sartzen den twitterraren luzera 140 karaktere gainditzea. Kasu horretan, erabiltzaileak mezu bat jasoko du egoeraz abisatuz, eta aldi berean Twitter klaseko testua atributuan gordeko den karaktere katea 0-139 karakterera murriztuko da (erabili `karaKate=karaKate.substring(0,139);` twitterraren luzera murrizteko) **(0,15puntu)**.
- 2) Jakinda erabiltzaile batek hainbat erabiltzaileen jarraitzailea izan daitekeela, twitterren jarraitu nahi den pertsona bat `listaJarraitu`-n jada balego, programak mezu bat atera beharko luke erabiltzailea egoera honetaz informatuz, `listaJarraitu` modu bereborean utziz (jarraitu nahi den pertsona ez balego erabiltzailearen `listaJarraitu`-n orduan txertatuko da, beti ere 3. puntuko salbuespen egoera kontutan izanda) **(0,2puntu)**.
- 3) Jarraitu nahi den pertsona berri bat sartzean, jarraitu daitezken 1000 pertsonen muga gaindituko balu, erabiltzailea abisatzeko mezu bat inprimatuko da, eta erabiltzaileari bere `listaJarraitu`-tik ezabatu nahi duen pertsonaren izena (ziurtatu behar gara '@'-z hasiera izen hori, begira 4. puntua) sartzeko eskatuko zaio. Hori egitean, gerta daiteke pertsona hori `listaJarraitu`-n ez egotea, kasu horretan, erabiltzaileari egoera horretaz informatuko dion mezu bat aterako da, eta berriro sartzeko aukera emango zaio, gehienez 3 saiakera edo `listaJarraitu`-n dagoen @izen bat sartu arte. Ezabatu daiteken izen bat sartuko balu, azken hau ezabatuko da, jarraitu nahi den pertsonaren @izena txertatuz **(0,35puntu)**.
- 4) Jarraitu nahi den pertsonaren `hashTag/izena` sartzean (String), izen hau '@' sinboloz hasi behar da. Horrela ez balitz, erabiltzaileak mezu bat jasoko du egoeraz informatuz eta eskakizun prozesua errepikatuko da, erabiltzaileak '@' hasten den String bat sartu arte **(0,2puntu)**.

Jarraian moldatu behar diren metodoen kodea ikus daiteke:

Erabiltzailea klasearen metodoak

```
public void sartuTwitterBat(){
// TODO: Aldatu kodea 140 karaktereen murriztapena salbuespen gisa antzemanetz eta posible bada metodo
// honetan bertan tratatuz, bestela goratuz, dagokionak tratatu dezan.
// Aurre: -
// Post: Erabiltzaileak teklatutik sartutako testua bere listaTwitterretan Twitter berri bat gisa gordeta
// izango du. Testuak 140 karaktere baino gehiago izango balitu, mezu bat aterako da pantailatik eta testua
// ebaki egingo da, Twitter berrian bakarrik lehenengo 140 karaktere gordez.
    String twitter="";
    Scanner sc=new Scanner(System.in);
    System.out.println("Sartu Twitter berri bat");
    twitter=sc.nextLine();
    Twitterra twit=new Twitterra(twitter);
    this.listaTwitterrak.gehituTwitterra(twit);
}

public void sartuNoriJarraitu(){
// TODO: Aldatu kodea gertatu daitezken salbuespen egoerak antzemateko eta posible bada metodo honetan
// bertan tratatuz, bestela goratuz, dagokionak tratatu dezan.
// Aurre: -
// Post: Erabiltzaileak teklatutik sartutako @izena (String) bere listaJarraitu-n sartuko du. KONTUZ!!
// listaJarraitu ListaJarraitzenDitu klasekoa da bakoitza Erabiltzailea bat den (ez String bat!!). Elementu
// kopurua ezin da izan 1000 baino handiagoa inongo kasuan.
    String twitter="";
    boolean hashTag=false;
    boolean badago;
    String nori="";
    String norEliminatu="";
    Erabiltzailea er=null;

    System.out.println("Nori jarraitu nahi diozu?: ");
    nori=this.hashTagSartu();
    er=new Erabiltzailea(nori);
    this.listaJarraitu.gehituBat(er);
}

public String hashTagSartu(){
// TODO: Aldatu kodea gertatu daitezken salbuespen egoerak antzemateko eta posible bada metodo honetan
// bertan tratatuz, bestela goratuz, dagokionak tratatu dezan.
// Aurre: -
// Post: Erabiltzaileak teklatutik sartutako String-a '@' hasiko da.
    String nori;
    Scanner sc=new Scanner(System.in);
    nori=sc.nextLine();
    if(!nori.startsWith("@")){

    }
    return(nori);
}
```

ListaJarraitzenDitu klaseko metodoak

```
public void gehituBat(Erabiltzailea pErab) {
// TODO: Aldatu kodea gertatu daitezken salbuespen egoerak antzemateko eta posible bada metodo honetan
// bertan tratatuz, bestela goratuz, dagokionak tratatu dezan.
// Aurre:-
// Post: pErab zerrendan txertatua geldituko da, ez balego jada.

    this.lista.add(pErab);
}

public void eliminatu(String norEliminatu) {
// TODO: Aldatu kodea gertatu daitezken salbuespen egoerak antzemateko eta posible bada metodo honetan
// bertan tratatuz, bestela goratuz, dagokionak tratatu dezan.
// Aurre:-
// Post: pErab zerrendatik ezabatuko da, zerrendako elementua baldin bada.
    Iterator<Erabiltzailea> itr;
    itr=this.getIteradorea();
    boolean topatua=false;
    while(itr.hasNext()&&!topatua){
        Erabiltzailea er=itr.next();
        if(er.idBerdinaDu(norEliminatu)){
            this.lista.remove(er);
            topatua=true;
        }
    }
    if(topatua==false){

    }
}
```