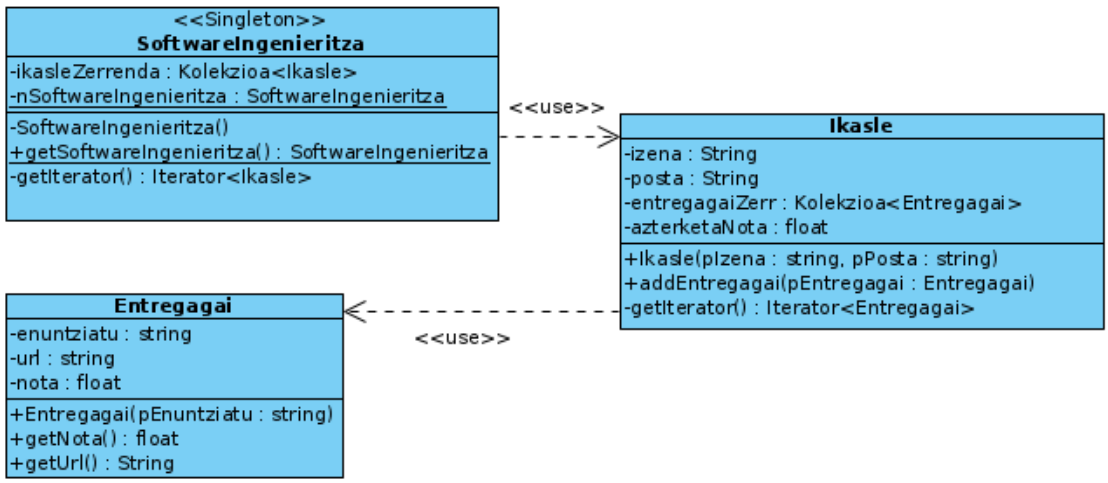
Java8 Laborategia

Software Ingeniaritza ikasgaiko noten kudeaketa automatizatzeko aplikazio bat eraiki nahi da. Ikasgai honetako kalifikazioa lortzeko entregagai desberdinen notak eta azterketa finaleko nota hartzen dira kontutan, honako formularen arabera:

*(0,4 * entregagarrienBB) + (0,6 * azterketaNota)*

Formula horretan *entregagaienBB*-k entrgagarrien noten batazbestekoa adierazten du, eta *azterketaNota*-k azterketa finelko nota. Hurrengo klase diagramak ariketa hau egiteko oinarrizko metodoak baino ez ditu.



Hurrengo puntuetan deskribatutako metodoak inplementatu eta gehitu beharko dituzu **`SoftwareIngenieritza`** eta **`Ikasle`** klaseetan. Betiere, Java8-k eskeinitako agregazio eta *stream* operazioen bidez egin beharko duzu, eta, hori lortzeko, kontsulta itzazu hurrengo interfazeen dokumentazioa: **`Stream`**, **`IntStream`**, **`DoubleStream`**, **`Collectors`**, **`Comparator`**, **`Predicate`** eta **`DoubleSummaryStatistics`**.

`Ikasle` klasea

1. `entregagarrienNotaKalkulatu` metodoa inplementatu ezazu. Metodo horrek ikasle baten noten batazbestekoa bueltatzen du.

```
public double entregagarrienNotaKalkulatu(){
   //TODO
}
```

Laguntza: erabili ezazu `mapToDouble` bitarteko operazioa. Batazbestekoa egiteko `average` operazioa erabili, azken horrek batazbestekoa `OptionalDouble` balioan bueltatzen du (fluxua hutsa bada , `OptionalDouble` hutsik egongo da). `Optional` hutsa ebitatzeko, `orElse` metodoak defektuzko balorea sartzen uzten du.

2. `notaFinalaKalkulatu` metodoa inplementatu ezazu. Metodo horrek ikaslearen nota finala kalkulatzen du, sarreran aipatutako formula erabilita.

```
public double notaFinalaKalkulatu(){
   //TODO
}
```

3. `entregagarriGuztiakGaindituDitu` metodoa inplemetatu ezazu. Metodo horrek `true` bueltatzen du ikasleak entregagarri guztiak gainditu baditu (nota ≥ 5)

```
public boolean entregagarriGuztiakGaindituDitu(){
   //TODO
}
```

Laguntza: `allMatch` operazio finala erabili. Azken horrek fluxu baten elementu orok baldintza jakin bat betetzen duen bueltatzen du.

4. `entregagarrietakoBatekNotaGaindituDu` metodoa inplementatu ezazu. Metodo horrek `true` bueltatzen du, ikasleak pNota baino altuagoa badauka entregagarrietako batetan.

```
public boolean entregagarrietakoBatekNotaGaindituDu(double pNota){
   //TODO
}
```

Laguntza: `anyMatch` operazio finala erabili. Azken horrek fluxu baten elementuetako batek baldintza jakin bat betetzen duen bueltatzen du.

SoftwareIngenieritza klasea

5. `notaTotalakErakutsi` metodoa inplementatu ezazu. Metodo horrek ikasgaiko ikasle ororen nota totalak pantailaratuko ditu.

```
public void notaTotalakErakutsi(){
      //TODO
}
```

Gainera, `Ikasle` klaseko `notaFinalaKalkulatu` erabili ezazu.

Laguntza: `mapToDouble` eta `forEach` erabili. Azken horrek ekintza bat (void metodoa) aplikatzen dio fluxuko elementu bakoitzari.

6. `gainditutakoakLortu` metodoa inplementatu ezazu. Metodo horrek ikasgaia gainditutako ikasleen zerrenda bueltatuko du eta, ondoren, inprimatu egingo du.

```
public List<Ikasle> gainditutakoakLortuJava8(){
      //TODO
}
```

Laguntza: `filter` erabili filtrorako eta `collect` ikasleen fluxua zerrenda batean bihurtzeko. Hori egiteko, `Collectors` klaseko `toList` erabili.

7. `gainditutakoakIzenezOrdenatutaLortu` metodoa inplementatu ezazu. Metodo horrek ikasgaia gainditutako ikasleen zerrenda bueltatuko du, baina, izenetik ordenatuta.

```
public List<Ikasle> gainditutakoakIzenezOrdenatutaLortu(){
   //TODO
}
```

Laguntza: ikasleen fluxua ordenatzeko `sorted` erabili. Azken horren barruan `Comparator` interfazeko `comparing` erabilita, konparaketa irizpide bati jarraiki egiten da.

8. `gainditutakoakIzenezAbizenezOrdenatutaLortu` metodoa inplementatu ezazu. Metodo horrek ikasgaia gainditutako ikasleen zerrenda bueltatuko du; lehenik izenez ordenatuko ditu, eta, ondoren, izen berdinekoak abizenez ordenatuko ditu.

```
public List<Ikasle> gainditutakoakIzenezAbizenezOrdenatutaLortu(){
      //TODO
}
```

Laguntza: aurrekoaren antzera, baina `comparing` ondoren `Comparator` interfazeko `thenComparing` gehituta. Azken horrek ordenazio iripide gehigarria txertatzen du, eta lehenengo ordenazio irizpideko elementuei aplikatzen zaie.

9. `gaindituenPortzentaiaLortu` metodoa inplementatu ezazu. Metodo horrek ikasgaia gainditutako ikasleen protzentaia bueltatuko du.

```
      public double gaindituenPortzentaiaLortu(){
            //TODO
      }
```

Laguntza: `count` operazio finala erabili.

10. `herrialdeakLortu` metodoa inplementatu ezazu. Metodo horrek ikasgaiko ikasleen herrialdeen zerrenda bueltatuko du (ezin dira errepikatu).

```
   public List<String> herrialdeakLortu(){
            //TODO
  }
```

Laguntza: `map` eta `distinct` erabili. Azkenengo operazio finalak fluxu batetako elemetu bereiziak bueltatzen ditu, eta errepikatutakoak ezabatzen.

11. `entregagarriGuztiakGainditutakoakLortu` metodoa inplementatu ezazu. Metodo horrek entregagai guztiak gainditutako ikasleen zerrenda bueltatuko du.

```
public List<Ikasle>
 entregagarriGuztiakGainditutakoakLortu(double pNota){
      //TODO
}
```

`Ikasle` klaseko `entregagarriGuztiakGaindituDitu` erabili ezazu.

12. `entregagarrianNotaGainditzenDutenIkasleakLortu` metodoa inplementatu ezazu. Metodo horrek nota bat jasotzen du, eta entregagarrietako batean nota hori gainditu duten ikasleen zerrenda bueltatzen du.

```
  public List<Ikasle>
      entregagarrianNotaGainditzenDutenIkasleakLortu(double pNota){
         //TODO
  }
```

`Ikasle` klaseko `entregagarrietakoBatekNotaGaindituDu` erabili ezazu.

13. `ikasleenEstatiskikakInprimatu` metodoa inplementatu ezazu. Metodo horrek ikasle guztien estatistika inprimatzen ditu; hots, nota *maximoa*, *minimoa* eta *batazbestekoa*.

```
      public void ikasleenEstatiskikakInprimatu(){
            //TODO
```

```
    }
```

Laguntza: `DoubleStream` barruko `summaryStatistics` operazio finalaren bitartez ariketa honetan eskatutako guztiak kalkulatu daitezke.

14. `gaindituakSuspendituakLortu` metodoa inplementatu ezazu. Metodo horrek gainditutako eta suspenditutako ikasleen zerrendak batzen dituen mapa bueltatzen du.

```
public Map<Boolean,List<Ikasle>> gaindituakSuspendituakLortu(){
    //TODO
}
```

Laguntza: `partitioningBy` metodoa erabili. Azken horrek fluxua bitan banatzen du; alde batetik, baldintza betetzen dutenak, eta, bestetik, betetzen ez dutenak.

15. `ikasleakHerrialdekaLortu` metodoa inplementatu ezazu. Metodo horrek herrialde bakoitzeko ikasleen mapa bueltatzen du.

```
public Map<String, List<Ikasle>> ikasleakHerrialdekaLortu(){
    //TODO
}
```

Laguntza: `groupingBy` metodoa erabili. Azken horrek, fluxua zehaztutako irizpideei jarraiki hainbat multzotan banatzen du.

16. `batazbestekoNotakHerrialdekaLortu` metodoa inplementatu ezazu. Metodo horrek herrialde bakoitzeko batazbesteko noten mapa bueltatzen du.

```
public Map<String, Double> batazbestekoNotakHerrialdekaLortu(){
    //TODO
}
```

Laguntza: aurrekoaren antzera, baina `Collectors` klaseko `averagingDouble` metodoa gehituta. Azken horrek fluxu bat batazbesteko batetara erreduzitzen du, behar diren transformazioak eginda.

17. `notaMaximodunIkasleaHerrialdekaLortu` metodoa inplementatu ezazu. Metodo horrek herrialde bakoitzean nota altuen eukitako ikaslearen mapa bueltatzen du.

```
public Map<String, Ikasle> notaMaximodunIkasleaHerrialdekaLortu(){
    //TODO
}
```

Laguntza: aurrekoaren antzera, baina `groupingBy` barruan `Collectors` klaseko `collectingAndThen` gehituta. Azken hori `groupingBy`-ren bigarren parametro legez sar daiteke, eta bigarren transformazio bat aplikatzen du (gure kasuan `maxBy`). `maxBy` erabilita datu fluxuan irizpide bati jarraitzen dion maximoa lor daiteke. Kontuan izan azken eragiketa horrek fluxu hutsa bueltatu dezakeela, eta horrexegatik `Optional` bueltatzen duela.

18. `notaMaximoaHerrialdekaLortu` metodoa inplementatu ezazu. Metodo horrek herrialde bakoitzeko nota altuenaren mapa bueltatzen du.

```
public Map<String, Double> notaMaximoaHerrialdekaLortu(){
    //TODO
}
```