

## **2. GAIA. ITERAZIOZKO AGINDUAK EDO AGINDU ERREPIKAKORRAK**



1

# SARRERA



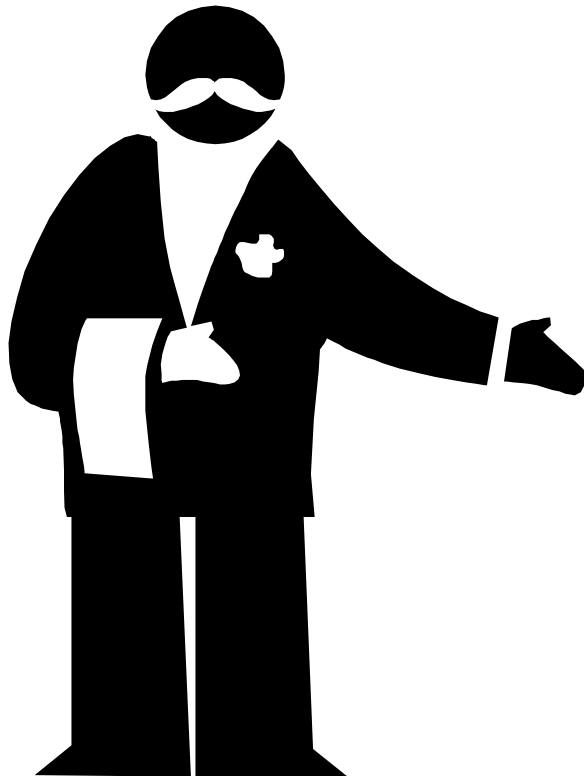
- 1) **Helburua**
- 2) Motibazioa
- 3) Iterazio eragiketak
- 4) Adibideak

# Gaiaren helburuak

- 1) Iterazioaren beharra ulertzea
- 2) iteraziozko kontrol egitura desberdinak  
Errepikatu atera (baldintza) egia denean  
aldagai hasierako\_balioa bukaerako\_balioa errepikatu  
(baldintza) egia den bitartean errepikatu



# SARRERA



- 1) Helburua
- 2) **Motibazioa**
- 3) Iterazio eragiketak
- 4) Adibideak

# MOTIBAZIOA

Demagun kalkulatu nahi dugula kurtso batean ikasleek lortutako batazbesteko nota

Espezifikazioa:

Aurre: notak  $\geq 0$

Post: Momentu arteko batazbestekoa pantailatik aterako da

Proba kasuak:  $0 \rightarrow 0$ ,  $10,0 \rightarrow 5$  ....

# MOTIBAZIOA

- Zenbat aldagai?
- Demagun esaten digutela gehienez jota 100 ikasle izango ditugula.

nota1, nota2, nota3,.. nota100, kont, nota\_akum:  
osoa;

# MOTIBAZIOA

```
kont←0; nota_akum←0;  
idatzi("Sartu nota bat");  
irakurri(nota1);  
kont ←kont+1;  
nota_akum←nota_akum+nota1;  
idatzi("Sartu nota bat");  
irakurri(nota2);  
kont ←kont+1;  
nota_akum←nota_akum+nota2;
```

?????



# MOTIBAZIOA

```
kont ← 0; nota_akum ← 0;
```

Errepikatu atera kont=100 egia denean

```
    idatzi("Sartu nota bat");
```

```
    irakurri(notaX);
```

```
    kont ← kont+1;
```

```
    nota_akum ← nota_akum+notaX;
```

```
amaitu_errepikatu;
```

Nola ulertu behar da hau?

```
kont ← kont+1; --- kont aldagaiaren balio berria da, lehenago  
                zeukana +1
```

```
nota_akum ← nota_akum+notaX; --- berdin hemen
```



# MOTIBAZIOA

Tranpa egin dugu!!!

Ariketan esaten ziguten GEHIENEZ 100 ez 100.

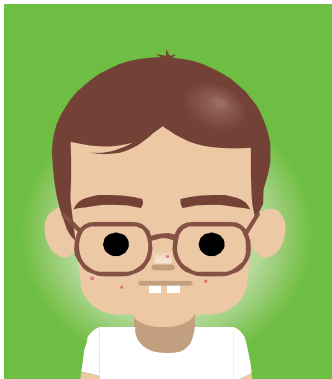
Nola egingo genuke hau? Demagun  
erabiltzailearekin adosten dugula kodea bat



Erabiltzaileak -1 bat sartuko du bukatu nahi duenean.

# Galdera

Nola gelditko litzateke algoritmoa?



Iterazioa bukatzeko  
baldintza egokia topatu  
behako dugu. Kasu  
kritikoak ere ager  
daitezke....



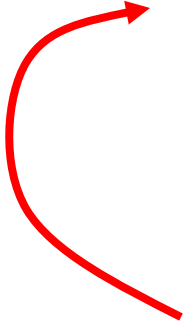
# MOTIBAZIOA

Hasieraketak

Errepikatu-atera Baldintza egia denean

Errepikatuko diren aginteak

amaitu\_errepikatu;



# ITERAZIOZKO AGINDUAK EDO KONTROL-EGITURAK

- Iteraziozko aginduetan programatzaileak zehaztutako baldintza bat betetzen den bitartean agindu-multzo bera egikaritzen da behin eta berriz.

# ITERAZIOZKO AGINDUAK EDO KONTROL-EGITURAK

- Errepikatu-atera (Baldintza) egia denean
- Aldagaia X hasierako\_balio\_tik  
bukaerako\_balio\_arte errepikatu
- Errepikatu (Baldintza) egia den bitartean

# ERREPIKATU-ATERA AGINDUA: ITXURA

Errepikatu-atera (baldintza) egia denean  
aginduak

amaitu\_errepikatu;

# ERREPIKATU-ATERA AGINDUA: ESANAHIA

- **Baldintza betetzen ez den bitartean**, hau da, baldintza gezurra den bitartean, Errepikatu eta amaitu\_errepikatu artean idatzita dauden aginduak egikarituko dira.

# ERREPIKATU-ATERA AGINDUA: ESANAHIA

- Errepikatu-atera agindu batera iristen garenean, **baldintza** betetzen al den ala ez **aztertuko** da. Betetzen ez bada, errepikatu eta bukatu-errepikatu markadoreen arteko agidu-multzoa egikaritu eta **berriro baldintzara itzuli beharko da.**
- Baldintza berriro aztertu eta oraingoan ere ez balitz beteko, aginduak berriz egikaritu eta baldintzara itzuli, baldintza egia izan arte



# ERREPIKATU-ATERA AGINDUA: ESANAHIA

- Hau da, prozesu hori behin eta berriz errepikatuko da **baldintza egiazkoa ez den bitartean.**
- Baldintza hasieratik **beteko balitz, aginduak exekutatu gabe geldituko lirateke,** hau da Errepikatu-atera horretan ez ginateke sartuko.

# ERREPIKATU-ATERA AGINDUA: ESANAHIA

- Errepikatu-atera agindua ulertzeko beste era bat honako hau da:  
**baldintza egia izan arte** agindumultzoa exekutatu behar da.

## ERREPIKAPENEN BALDINTZAK

- Iteraziozko aginduetako baldintzak, baldintzazko aginduetarako azaldutako baldintzak bezala eraikitzen dira.
- Beraz, baldintzak eragile **erlazionalak** (  $=$ ,  $<$ ,  $>$ ,  $>=$ ,  $<=$  ) eta eragile **logikoak** ( $!$ , and, or) erabiliz eratutako espresio logikoak izango dira.

## ADIBIDEA

- Eskatu erabiltzaileari zenbakiak sartzen joateko 0 bat sartu arte. Gure programak zenbaki bakoitzeko, zenbakia bikoitia den ala ez adieraztebn duen mezua pantailatik aterako du.

## 1) Espezifikazioa

Aurre:  $zenb \geq 0$

Post:  $zenb/2$  hondarra zero denean mezu bat atera, 0-rekin bukatu

## 2) Proba kasuak

Sarrera

Irteera

1,2,3,4,0

1,0

2,4,0

## 3) Algoritmoa

```
zenb: osoa;  
idatzi("Sartu zenbaki bat");  
irakurri(zenb);  
errepikatu-atera(zenb=0) egia denean
```

```
(zenb/2 hondarra = 0) egia denean orduan  
                                idatzi(zenb "zenbakia bikoitia da");
```

```
amaitu;  
idatzi("Sartu beste zenbaki bat");  
irakurri(zenb);
```

```
amaitu_errepikatu;
```

# 1. ARIKETA: 1. BERTSIOA.

- Ariketa honetan lehenengo 100 zenbaki oso eta bikoitiak pantailan aurkezten dituen programa idatzi behar da.

0,2,4,6,8 ....200

## 1) Espezifikazioa

Aurre:

Post:

2) Proba kasuak

3) Algoritmoa



## 2. ARIKETA.

- Ariketa honetan idatzi beharreko programak erabiltzaileari osoa eta positiboa ( $\geq 1$ ) den  $n$  zenbaki bat eskatu eta lehenengo  $n$  **zenbaki naturalen** batura kalkulatu eta pantailan aurkeztu behar du.

## 2. ARIKETA.

- Erabiltzaileari  $n$  zenbaki bat sartzeko eskatuko dion programa bat idatzi. Erabiltzaileak tekleetutako  $n$  zenbakia 1 baino txikiagoa balitz, mezu bat aurkeztu beharko lukezenbakia egokia ez dela esanez, eta beste zenbaki bat eskatuko du (eskaera prozesua ez da bukatuko zenbakia egokia izan arte, hau da  $\geq 1$ ).

# 1) Espezifikazioa

Aurre:

Post:

# 2) Proba kasuak

# 3) Algoritmoa

# ALDAGAIA X HASIERAKO\_BALIOTIK BUKAERAKO\_BALIO\_ARTĒ ERREPIKATU\_AGINDŪA: ITXURA

aldagaia X hasierako\_baliotik ..bukaerako\_balio\_arte Errepikatu

aginduak

amaitu\_errepikatu;

# ERREPIKATU (BALDINTZA) EGIA DEN BITARTEAN

Errepikatu (Baldintza) egia den bitartean

aginduak

amaitu\_errepikatu;