OBPM Proiektua

Baliabide lagungarriak

Aurkibidea

- **Helburuak**
- Bererabil daitezken klaseak
- S/I testu fitxategiekin

Helburuak

- Klaseetan aipatu ez diren eta proiektua garatzeko lagungarriak izan daitezken gai batzuk tratatzea
- Proiektu batzuetan (ber)erabili daitezken klaseak aurkeztu

Aurkibidea

- Helburuak
- **Bererabil daitezken klaseak**
- S/I testu fitxategiekin

Motibazioa

- Ber-erabilpeena
  - Teklatu klasea Jokalariek edozein momentuan informazioa teklatutil sartzeko erabiliko da
  - Erlojua/Kronometroa
  - Baraja: Mus, briska, eskoba etab
  - Dadoa: Oka, partxis, tribial

Teklatua

- Jokalarien Sarrerak zentralizatzen ditu
  - Metodo bakoitzak bere salbuespenak kudeatzen ditu (adibidez, NumberFormatException) eta egikariketa ez da bukatzen erabiltzaileak balio egoki bat sartu arte

Teclado
- static Teclado miTeclado; *- Scanner sc;*
+ leerString(String pMensajePrevio): void *+ leerEntero(String pMensajePrevio, int pDesde, int pHasta): int* *+ leerSiNo(String pMensajePrevio, String pSi, String pNo): boolean* *+ leerOpcion(String pMensajePrevio, ListaStrings pOpciones): String* *etcétera...*

Egikariketa gelditzea

- Milisegundu batzuetan
  - *Thread.sleep(numSegundu*1000);*
- Erabiltzaileak tekla bat ikustu arte
  - *Teklatua.getTeklatua().irakurriString("Sakatu enter jarraitzeko");*

Data eta ordua

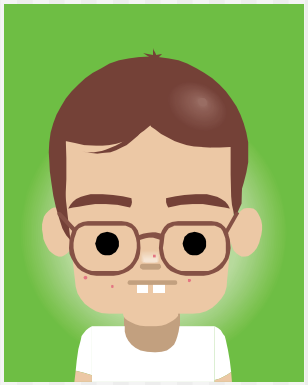
- Calendar (ikusi 3. laborategia)
  - Caldendar erabilita gorde daiteke galdera bat egitetik erantzuna jao arteko denbora

```
Calendar c = new GregorianCalendar();
int h12, h24, min, seg;

h12 = c.get(Calendar.HOUR);
h24 = c.get(Calendar.HOUR_OF_DAY);
min = c.get(Calendar.MINUTE);
seg = c.get(Calendar.SECOND);
```

Pasa den denbora

- Kalkulatu daiteke hurrengo erabiliz:
  - *System.currentTimeMillis(): long*
    - 1970ko Urriaren 1etik pasatu den denbora bueltatzen du



Horrela metodo bat egikatu aurretik denbora lortzen badugu eta gero berriro egikaritu ostean, pasaDenDenbora kalkulatu dezakegu

Kronometro

- Kronometro sinple bat hau izan daiteke

```
private long zeroMomentua;

public void zeroanJarri()
{
    zeroMomentua = System.currentTimeMillis();
}

public int pasaDirenSegunduakLortu()
{
    return (int) (System.currentTimeMillis() - zeroMomentua) / 1000;
}
```

Dadoa

- El dado 1 eta aldeZenbaki arteko balio randomak lortzeko erabil daiteke

```
private int nAldeak = 6;

public int botaDadoa()
{
   Random r = new Random();
   int tirada = r.nextInt(nAldeak) + 1;
   return tirada;
}
```

DadoTest

- Dado baten Juniat egiteko *truke dado erabilkiko da*
  - Bueltatuko da eskatzen zaiona
  - Bakarrik Junitetan erabilkiko da

```
public int botaDadoa()
{
   return Teklatua.getTeklatua().leerEntero
          ("Hautatu balio bat: ", 1, nAldeak);
}
```

Baraja

- Karta Klasea
  - Izena eta paloa atributu gisa
  - Eta batzuetan balio atributua ere izango dute

izena	palo	balio
Caballo	Espadas	10 (Mus) 9 (Escoba) 3 (Tute) 0.5 (Siete y medio)

Barajaren metodo erabillenak

- Eraikitzailea *Baraja()*

Baraja bat sortuko du

- *lapurtu(): Karta*

Karta bat bueltatuko du barajatik kenduta

- *barajatu(): void*
  - Collections.shuffle(this.kar
- *gehitu(Karta pKarta): voi*

Baraja
- static Baraja miBaraja; ***- ArrayList<Carta> cartas;***
- Baraja() ***+ robar(): Carta*** ***+ barajar(): void*** ***+ anadir(Carta pCarta): void*** ***...***

BarajaTest

- Aurreragotrukatu metodoaren isppiritu berdinaerkin *kartaKokatu* erabiliko da. Metodo hau probak egiteko baino ez da erabiliko

```
public void kartaKokatu(String pIzena, String pPalo, int pPos)
{
    Karta kartaBat = bilatuKarta(pIzena, pPalo);
    this.kartak.remove(kartaBat);
    this.kartak.add(pPos, kartaBat);
}
```

Aurkibidea

- Helburuak
- Bererabil daitezken klaseak
- **S/I testu fitxategiekin**

Irakurri teklatutik

Zenbakiak irakurtzeko

```
Scanner sc = new Scanner(System.in);
int i = sc.nextInt();
```

Irakurri teklatutik

Lerroak irakurtzeko sc.nextLine().

```
System.out.println("Enter your username: ");
Scanner scanner = new Scanner(System.in);
String username = scanner.nextLine();
System.out.println("Your username is " + username);
```

Informazio gehiago: https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html

Bidea (path) fitxategi ibat erabiltzeko

- Absolutua (C:\Users\aitzi\Desktop\datuak.txt)
  - Zuzenean kodean erabiliko da
  - Arazoak ematen ditu beste ordenagailu batetan egikaritzeko
- Erlatiboa (.\datuak.txt)
  - Sortu behar duzuen .jar edonondik erabiltzea ahalbidetzen du

Bide erlatiboa ezarri

- Horretarako egikariketa bidearekin konkatenatu behar da
  - File.separator portabilitatea ziurtatzen du
    - "\" (Windows) edo "/" (Unix/Linux, Mac)-eko baliokidea

```
String egungoDir = System.getProperty("user.dir");
String pathIn = egungoDir + File.separator + "sarrera.txt";
String pathOut = egungoDir + File.separator + "irteera.txt";
```

- Irakurri fitxategi bat
  - InputStream + Scanner
  - String.split()
- http://www.javamexico.org/foros/comunidad/creacion_de_archivos_dentro_de_un_jar

Idatzi fitxategi batetan

- Gogoratu jokoa .jar fitxategi batetan sartu behar duzue
  - Idatzi behar baduzue fitxategi batetan, fitxategi hori .jar fitxategitik kanpo egon behar da
    - .jar fitxategiko bide erlatiboa baina .jar kanpo

.jar fitxategia

- Sortzeko
  - Export→Java→Runnable JAR File
    - Main() matodo bat dagoelaz ziurtatu behar zarie
- Egikaritzeko MsDos (cmd) edo linux-eko terminal bat zabaldu behar duzue
  - java -jar jokoa.jar


galderarik?