

# Esercitazione 2

# La “libreria” Math

```
Double pi=Math.acos(-1.0);
```

```
sin, cos, tan, sinh, asin...
```

```
floor, ceil, round...
```

```
log, log10, exp, pow, sqrt, cbrt...
```

```
max, min...
```

```
Math.random() => n ∈ [0.0,1.0)
```

View the API!

<https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html>

# Come deve essere un buon numero casuale?

- Ben distribuito (la probabilità di estrarre un qualunque numero deve essere uguale a quella di estrarne qualunque altro)
- Scorrelato (dato un numero appena estratto, la probabilità che il prossimo numero sia un numero dato non dipende da quello appena estratto)
- Avere probabilità adeguabile alla distribuzione voluta (es. Gaussiana)
- Riproducibile (!)

# Numeri pseudocasuali

Sono detti numeri pseudo-casuali (pseudo-random numbers) i numeri

- generati da un algoritmo deterministico
- che produce una sequenza
- avente le stesse proprietà statistiche di una sequenza di numeri generata da un processo casuale.

# Class java.util.Random

Random(long seed)

- Creates a new random number generator using a single long seed.

Random generator =

new Random(System.currentTimeMillis());

public int nextInt(int n)

Returns a pseudorandom, uniformly distributed int value between **0 (inclusive)** and the **specified value (exclusive)**, drawn from this random number generator's sequence.

nextFloat(), nextBoolean(), nextGaussian()...

# Per le Liste c'è di meglio!

`public static void shuffle(List<?> list)`

Randomly permutes the specified list using a default source of randomness. All permutations occur with approximately equal likelihood.

Es.:

```
List<Carta> mazzo=new ArrayList();
```

... popola il mazzo...

```
Collections.shuffle(mazzo);
```

# Parte 1

Creare una mazzo di carte da ramino  $\{A, 2...10, J, Q, K\}_i, i \in \{C, Q, F, P\}$ , due carte per tipo.

Mescolare il mazzo in modo casuale e mostrare le prime N carte (default N=10).

Controllare se tra le carte c'è una doppia e mostrare una finestra, che dirà "hai vinto" se la si è trovata, "hai perso" altrimenti.

## Parte 2

Verificare cosa cambia nell'andamento del programma cambiando la classe che implementa il mazzo:

- Usando ArrayList
- Usando LinkedList
- Usando HashSet

# Varianti

- Rendere N scegliibile dall'utente, e validarne la risposta.
- Dare, nella finestra finale, una rappresentazione grafica semplificata della carta vincente.
- Dopo la eventuale vittoria, scegliere a caso una carta dal mazzo. Se questa ha lo stesso valore della vincente (indipendentemente da seme), la vittoria vale doppio. Dare comunicazione all'utente dell'esito di tale operazione.
- Implementare "a mano" (usando random) il mescolamento del mazzo