

GESTIONE DELLE ECCEZIONE

PRECONDIZIONI

- Un metodo di solito necessita un parametro che rispetti dei **vincoli**
 - es. numero deve essere positivo

ARGOMENTI INATTESI

- Cosa succede se il parametro NON rispetta le precondizioni?
- Modi (ingenui) per gestirli
 - 1. eseguire solo se le precondizioni sono soddisfatte (attraverso `if`)
 - 2. terminare il programma
 - attraverso `System.exit(1)`
 - 3. usare una **asserzione**
 - `assert <boolean expression>;`
 - eseguire attraverso `java -ea NomeClasse`
- Modo consigliato per farlo
 - lanciare un'**eccezione**

LANCIARE ECCEZIONE

- Throw exception
 - in risposta a un parametro che non rispetta una precondizione
 - Libreria standard che mette a disposizione l'eccezione: `IllegalArgumentException`
- Sintassi per lanciare eccezione

```
throw new ClasseEccezione();  
throw new IllegalArgumentException(); // esempio
```

- Conseguenze
 - esecuzione metodo **interrotta**
 - eccezione propagata fino al metodo chiamante
 - se eccezione non viene gestita e arriva al metodo main() -> si termina programma

GESTIRE ECCEZIONE

- Invocazione del metodo deve essere racchiuso in un blocco `try`

- Nel blocco `catch` si trova il codice da eseguire nel caso si verifichi l'eccezione
- Si possono mettere + blocchi `catch`

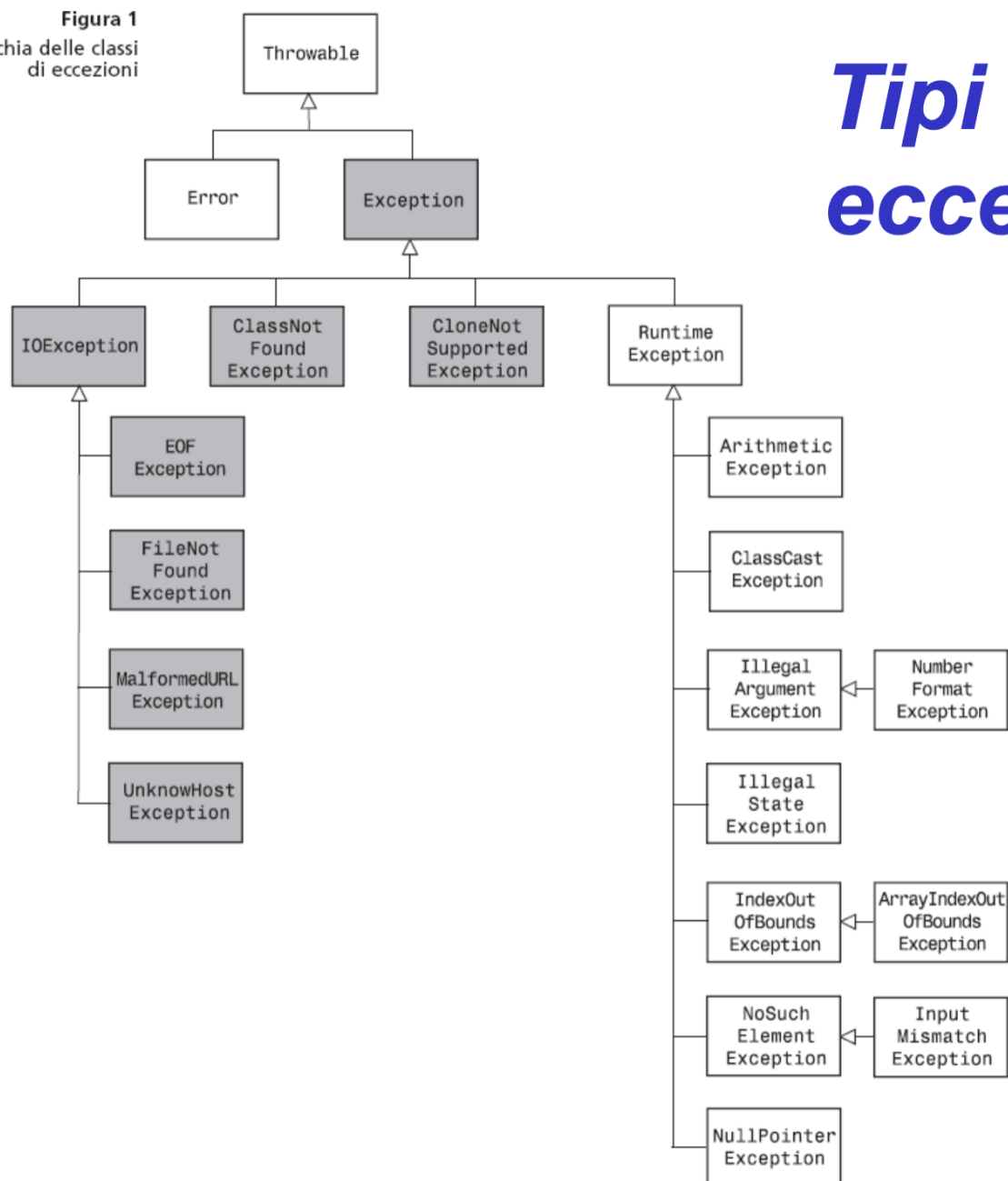
```
try {
    int n = Integer.parseInt(line);
} catch (NumberFormatException e) {
    System.out.println("Numero non valido");
}
```

COME CAPIRE QUALI ECCEZIONI POTREBBERO ESSERE SOLLEVATE?

- Documentazione

TIPI DI ECCEZIONI

Figura 1
chiave delle classi
di eccezioni



***Tipi di
eccezioni***



ERROR

- Eccezione che dipende dall'**ambiente** in cui il programma funziona
- Esempi
 - `OutOfMemoryError`

EXCEPTION

- Condizione anomala del **programma** stesso
- Eccezioni **non controllate** (bianco)
 - gestione facoltativa
- Eccezioni **controllate** (grigio)
 - gestione obbligatoria
- Esempi
 - `IOException` (eccezione controllata)

MARCATORE THROWS

- Per propagare un'eccezione bisogna aggiungere il marcatore `throws` nella firma del metodo
- Per non gestire eccezioni controllate, aggiungere marcatore `throws` su main

```
public static void main(String[] args) throws IOException {  
    // enunciati che potrebbero sollevare IOException  
}
```

CLAUSOLA FINALLY

- Il corpo della clausola `finally` viene eseguito sempre

```
try {  
    // enunciati che potrebbero sollevare eccezioni  
} finally {  
    // esegui sempre questi enunciati  
    // es. chiudi Scanner  
}
```