

Eccezioni

Liceo G.B. Brocchi - Bassano del Grappa (VI)
Liceo Scientifico - opzione scienze applicate
Giovanni Mazzocchin

Eccezioni

- In molti linguaggi di programmazione, le **eccezioni** (*exceptions*) costituiscono un metodo per gestire gli errori
- L'idea fondamentale è separare la rilevazione dell'errore dalla sua gestione
- Idea di fondo: se una funzione rileva un errore che non è in grado di gestire, dovrebbe lanciare (**throw**) un'eccezione, indicando cosa è andato storto
- Il chiamante della funzione (diretto o indiretto tramite la catena delle chiamate) dovrebbe occuparsi di *afferrare* (**catch**) l'eccezione lanciata
- Il chiamante specifica interesse in un possibile lancio di eccezioni in un blocco **try**

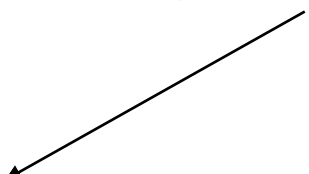
Eccezioni

```
class bad_radius {};
```

```
double perim_circle(double radius)
{
    if (radius < 0) {
        throw bad_radius();
    }
    return 2 * 3.1415 * radius;
}
```

Eccezioni

```
int main(int argc, char* argv[]) {  
    perim_circle(-3);  
    cout << "program goes on..." << endl;  
}
```



il programma termina prima, quindi non stampa la stringa

messaggio a runtime:

terminate called after throwing an instance of 'bad_radius'
Aborted

L'eccezione è stata lanciata, ma nessuno l'ha gestita: programma abortito

Eccezioni

```
int main(int argc, char* argv[]) {  
    try {  
        perim_circle(-3);  
    }  
    catch(bad_radius) {  
        cout << "bad radius argument" << endl;  
    }  
    cout << "program goes on..." << endl;  
}
```

L'eccezione è stata gestita nel blocco catch.
Il programma continua la sua esecuzione