

**Sveučilište u Zagrebu**  
**PMF – Matematički odjel**



# **Objektno programiranje (C++)**

Vježbe 12 – Iznimke

**Matej Mihelčić**

# Iznimke

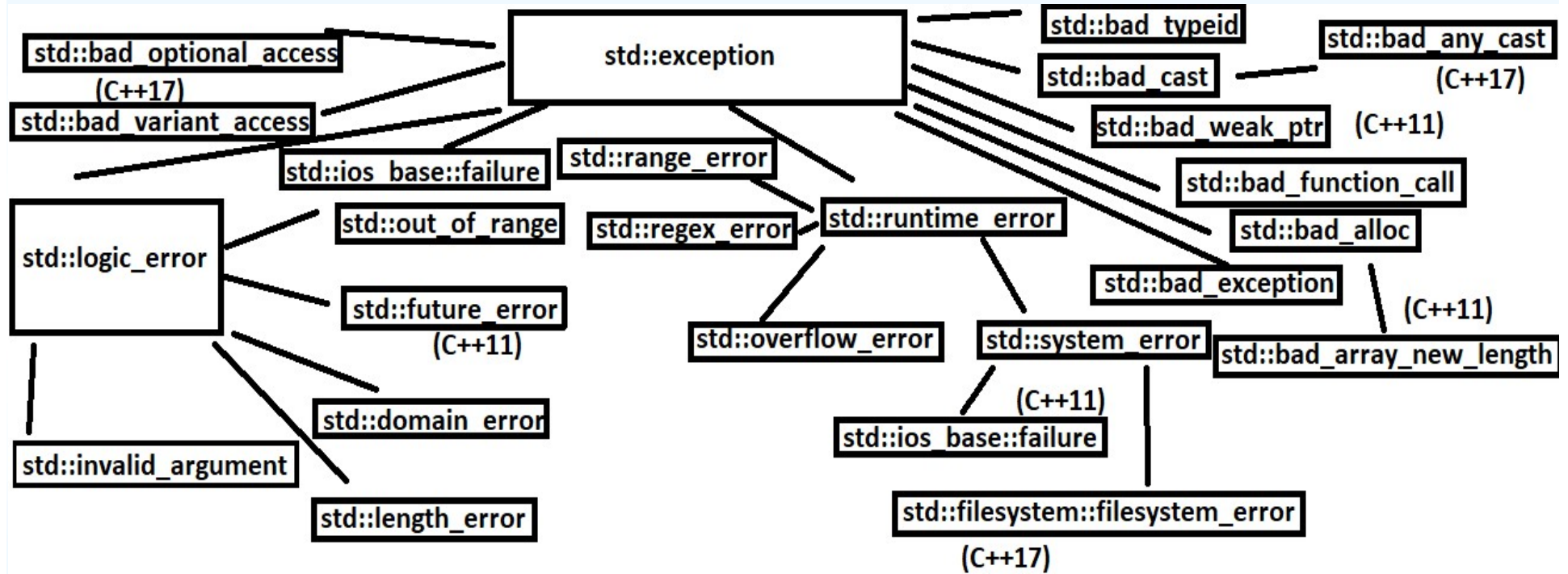
- Nastaju kao odgovor programa na iznimna stanja pri izvršavanju.
  - Omogućavaju prijenos kontrole s jednog dijela programa u drugi.
  - Glavni dijelovi svake iznimke su **try**, **catch** i **throw**.
  - **throw** – program prijavljuje iznimku na mjestu pojavljivanja ključne riječi **throw** (poziva se kada u programu dođe do iznimnog stanja, npr. dijeljenje s nulom, indeks koji nije sadržan u polju, nepostojeća datoteka, nemogućnost alokacije memorije itd).
  - **try** – označava dio koda u kojem bi moglo doći do prijavljivanja iznimke (odnosno dio koda se pokušava izvršiti). Nakon bloka koda koji je označen ključnom riječi **try** dolazi jedan ili više blokova koda za obradu iznimki (označeni ključnom riječi **catch**).
  - **catch** – označava dio koda koji sadrži naredbe za obradu iznimke određene vrste. Tip iznimke koji se obrađuje i odgovarajuća instanca iznimke se definiraju unutar uvjeta **catch** naredbe. Posebni oblik uvjeta **catch(...)** označava da dio koda za obradu iznimke obrađuje proizvoljnu iznimku.
- Kod unutar bloka **try/catch** se zove **zaštićeni kod**.

# Iznimke

```
- try {  
    // zaštićeni kod  
} catch( Tiplznimke1 e1 ) {  
    // blok obrade iznimke Tiplznimke1  
} catch( Tiplznimke e2 ) {  
    // blok obrade iznimke Tiplznimke2  
    .  
    .  
    .  
} catch( Tiplznimke eN ) {  
    // blok obrade iznimke TiplznimkeN  
}
```

Klasa **std::exception** je generalna klasa za definiranje iznimki u C++-u. Postoje i druge izvedene klase iznimaka (nasljeđuju `std::exception`) koje se razlikuju po svojoj namjeni (vrsti iznimaka koje obrađuju).

# Iznimke



# Iznimke

## Zadatak:

- Definirajte klasu **interval** koji reprezentira zatvoreni interval realnih brojeva. Definirajte operatore **presjek (-)** i **kontinuirana unija (+)**. **Kontinuirana unija** je definirana kao i obična unija ali samo za dva intervala koji imaju ne trivijalni presjek. Ukoliko se navedena unija pozove za dva disjunktna intervala treba prijaviti iznimku tipa **invalid\_argument**.
- Dinamički alocirajte polje od **n** pokazivača na elemente tipa **interval** (obavezno obradite iznimku **bad\_alloc**).
- Definirajte funkciju **interval\*\* trazi(double d)** koja vraća polje pokazivača na one intervale koji sadrže **d**. Ukoliko takav interval ne postoji, funkcija prijavljuje iznimku **out\_of\_range**.

# Iznimke

## Zadatak:

- Definirajte klasu fizikalna jedinica koja nasljeđuje klasu interval. Klasa uz minimalnu i maksimalnu vrijednost fizikalne jedinice, sadrži i string (ime fizikalne jedinice), te vektor mjerenja.
- Napišite funkciju void dodajMjerenje(double m) koja dodaje novo mjerenje zadane fizikalne jedinice. Ukoliko je mjerenje izvan dozvoljenog raspona, funkcija prijavljuje grešku tipa **range\_error**.
- Napravite **dynamic\_cast** iz klase fizikalna jedinica u klasu interval i natrag u klasu fizikalna jedinica.
- Probajte napraviti isto s recimo klasom osoba.

# Iznimke

## Zadatak:

- Napišite klasu **pozitivan interval** koja reprezentira pozitivne zatvorene realne intervale. Ukoliko je bilo koja granica intervala negativna, konstruktor treba prijaviti iznimku tipa **invalid\_argument**. Postavite standardnu funkciju za obradu svih iznimaka void obradi(); Navedena funkcija izbacuje iznimku ukoliko je interval dobro definiran (donja granica je manja/jednaka od gornje). Napravite da se program ne ruši ukoliko je interval korektno definiran već redefinira interval i nastavlja izvođenje.

# Iznimke

## Zadatak:

- Napišite klasu životinja koja sadrži string ime\_zivotinje i string glasanja. Ukoliko se pokuša kreirati instanca koja ima nedopušteni par ime\_zivotinje, glasanje, treba prijaviti iznimku klase KrivoGlasanje (korisnički definirana iznimka). Instance te iznimke proizvedu tekst: „Životinja *k* se ne glasa kao *v*”. Dozvoljena glasanja spremite u globalnu mapu.