

**Sveučilište u Zagrebu**  
**PMF – Matematički odsjek**



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

Predavanja 11 - Operatori

**Vinko Petričević**

# GMP – alokacija memorije

- Sve strukture/kalse mogu raditi sa proizvoljno velikim brojevima (koliko god imamo memorije na računalu)
- Po defaultu će koristiti normalne funkcije alloc/realloc/free
- Ali te funkcije možemo jednostavno promijeniti sa  
void mp\_set\_memory\_functions (  
    void \*(\*alloc\_func\_ptr) (size t),  
    void \*(\*realloc\_func\_ptr) (void \*, size t, size t),  
    void (\*free\_func\_ptr) (void \*, size t))  
ali to moramo napraviti prije nego što je aktivan ijedan GMP objekt
- Isto tako imamo i mp\_get\_memory\_functions

# Expression templates

- Prilikom rada sa različitim tipovima podataka često dolazi do nepotrebnog kopiranja podataka
- Npr. Radimo li s vektorima, i napravimo li uobičajene operacije za zbrajanje/oduzimanje/množenje vektora skalarom, najvjerojatnije će raditi višestruko sporije od C-izvedbe
- Ali templateovi nam omogućuju da definiramo operatore slično kao na nekim ranijim predavanjima, pa tek na operatoru pridruživanja radimo stvarno računanje, a templateovi nam pamte stablastu strukturu operacija koje vršimo
- To se događa za vrijeme kompajliranja programa
- No treba pariti na neke detalje – pogledati primjere
- Slično rade i klase u GMP-u

# RTTI

- RTTI je kratica za run-time type identification, odnosno identificiranje dinamičkog tipa objekta, kojeg dohvaćamo kroz referencu ili pokazivač, za vrijeme izvršavanja programa.
- typeid operator danoj referenci ili pokazivaču vraća dinamički tip objekta. Vraća objekt tipa `std::type_info` koji sadrži informacije o tipu (header `<typeinfo>`)
- `dynamic_cast` operator koji vrši sigurnu konverziju pokazivača/reference na bazni tip u pokazivač/reference na izvedeni. Da bismo ga koristili, tip mora imati barem jednu virtualnu funkciju
- Primjer – `dynamic_cast.cpp`, `typeid`
- Klasa `type_info` ima operatore `==`, `!=`, funkciju `before`, te funkciju `name()` koja vraća ime tipa. Primjer `type_name`
- Primjena – uspoređivanje objekata
- Primjeri – različite implementacije funkcije za različite izvedene tipove