

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



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

Vježbe 04 – Neuređeni spremnici

**Matej Mihelčić**

# Neuređeni spremnici

- Četiri vrste neuređenih spremnika: **unordered\_set**, **unordered\_map**, **unordered\_multiset**, **unordered\_multimap**.
- Umjesto operatora usporedbe i sortiranja elemenata, koriste **hash funkciju** kojom objekt smještaju u jedan od pretinaca. Provjera postojanja objekta unutar spremnika se odvija korištenjem **==** operatora na tipu ključa (ili objekta ukoliko nema ključa).
- Funkciju hashiranja i operator **==** trebamo **definirati sami**
- Korisno koristiti kada **radimo s objektima među kojima nema jasno definiranog uređaja** ili kada je dosta **skupo kontinuirano održavati poredak između objekata** (jako veliki broj objekata, objekti se kontinuirano dodaju i izbacuju iz spremnika).

## Svojstva hash funkcije:

- Dani objekt **uvijek** preslikavaju u istu vrijednost.
- Poželjno da objekte spremaju disperzirano (raspršeno) po pretincima. Odnosno, da su pretinci podjednako popunjeni (pretraga unutar pretinca **je linearna**).
- Poželjno da se **brzo računaju** (u praksi uglavnom slučaj).

# Zadatak

## Zadatak:

- Napravite klasu student koja ima ime, prezime, jmbag i niz ocjena. U 6 datoteka imena „kolegij\_i.txt” su zapisane ocjena za 20 različitih studenata. Svaki redak u datoteci je oblika JMBAG ocjena. Datoteka „studenti.txt” sadrži 20 redaka s imenom, prezimenom i JMBAG-om svakog studenta (elementi odvojeni razmakom). Program treba spremiti studente u uređenu mapu (leksikografski po JMBAG-u). Nakon toga, učitavaju se ocjene studenata redom iz datoteka. Nakon svake nove pročitane ocjene studenta, računa se prosjek ocjena. Svi studenti s prosjekom  $\geq 4.5$  („odlikaši”) se spremaju u poseban skup (u kojem poredak studenata nije bitan). Svaki puta kada prosjek studenta padne ispod 4.5, on se miče iz skupa odlikaša, također svaki puta kada prosjek postane  $\geq 4.5$  ponovo se vraća u skup odlikaša. Ispišite stanje skupa odlikaša nakon učitavanja ocjena iz svake datoteke (ukupno 6 ispisa).

# Zadatak

## Zadatak:

Učitavajte riječi s terminala i brojite koliko je puta upisana koja riječ (koristite `unordered_map`). Učitavanje se prekida unosom EOF. Ispišite broj pretinaca, najveći pretinac, broj elemenata u svakom pretincu, prosječan broj elemenata po pretincu, najveći mogući broj pretinaca u spremniku. Zatim, učitajte dodatnih 10 riječi te ponovo ispišite iste informacije o spremniku.

# Zadatak

## Zadatak:

- U datoteci „pjesma.txt” se nalazi pjesma jednog poznatog pjesnika. Izbrojite broj pojavljivanja svakog slova u navedenoj datoteci. Izbacite iz spremnika sva slova koja se javljaju manje puta od prosjeka te ispišite listu preostalih slova uz brojeve njihovog pojavljivanja u pjesmi. Isprobajte nekoliko različitih načina hashiranja slova i proučite razlike u strukturi spremnika.

# Zadatak

## Zadatak:

U knjižnici je svako izdanje neke knjige na jedinstven način određeno svojim ISBN brojem (string oblika zzz-z-zz-zzzzzz-z, z - znamenka). Svaka osoba je na jedinstven način određena brojem svoje članske iskaznice (maksimalno 5-znamenkasti prirodan broj). Dostupni naslovi su zapisani u datoteci „knjige.txt” a članovi u datoteci „clanovi.txt”. Pretpostavljamo da imamo neograničenu zalihu svakog naslova. Posudba knjige se simulira učitavanjem ISBN broja i id-a člana iz datoteke „posudbe.txt”. Napišite program koji broji koliko puta je posuđen svaki naslov. Svi naslovi iz datoteke „knjige” moraju biti navedeni u konačnoj listi (čak i ako im je broj posudbi 0).