

# Korisnički definisani literali

- ❖ Kada neki apstraktni tip podataka implementira neku fizičku veličinu, numeričke vrednosti te veličine u programu ne mogu da sadrže informaciju o jedinici u kojoj je izražena ta vrednost. Takav program može biti osetljiv na greške ukoliko se na različitim mestima koriste vrednosti, date kao literali, izražene u različitim jedinicama mere. Na primer:

```
class Pressure {  
public:  
    Pressure (long double); // Pressure in Pa  
    ...  
};
```

```
Pressure atm = 10.e5;
```

```
Pressure p0 = 1.0;
```

Šta ako je ova vrednost greškom izražen u barima?

- ❖ Za ovakve potrebe, jezik C++ omogućava upotrebu *korisnički definisanih literala* (*user-defined literals*), za koje se mogu definisati *literalni operatori* (*literal operator*), kao operatorske funkcije koje pretvaraju korisničke literale u neki tip:

```
class Pressure {  
private:  
    friend Pressure operator"" _pa (long double);  
    friend Pressure operator"" _bar (long double);  
    Pressure (long double);  
    ...  
};  
  
Pressure operator"" _pa (long double val) {  
    return Pressure(val);  
}  
  
Pressure operator"" _bar (long double val) {  
    return Pressure(val*10.e5);  
}
```

```
Pressure atm = 10.e5_pa;
```

```
Pressure p0 = 1.0_bar;
```

Zadatak: rešite ovaj problem bez korišćenja korisnički definisanih literala!  
Uporedite preglednost notacije u oba slučaja.

Poziva se `operator"" _pa (10.e5)`

Poziva se `operator"" _bar (1.0)`

# Korisnički definisani literali

- ❖ Korisnički definisani literali imaju isti format kao i ugrađeni literali, s tim što imaju sufiks koji mora da počne donjom crtom `_` i malim slovom iza nje; sami literali mogu biti u formatu celobrojnih, racionalnih, znakovnih ili string-literala, ali bez sufiksa koji se koriste za takve ugrađene literale
- ❖ Literalski operatori moraju da imaju naziv *operator*""", i za koga sledi identifikator koji počinje donjom crtom i malim slovom iza nje. Kao parametre mogu da imaju samo neke od dozvoljenih tipova, u koje spadaju samo *unsigned long long* za celobrojne literale, *long double* za racionalne literale, *char* i njegove varijacije za znakovne literale i *const char\** i varijacije za string literale (za njih se koristi i drugi argument koji daje veličinu niza znakova u string-literalu). Na primer:

```
constexpr long double operator"" _deg (long double deg) {  
    return deg*3.1415926/180;  
}
```

- ❖ Korisnički string-literali mogu da se koriste za identifikaciju instanci apstrakcije koje se ne mogu predstaviti numeričkim literalima; na primer:

```
class Date {  
public:  
    Date (unsigned y, unsigned m, unsigned d);  
    ...  
};  
  
Date operator"" _date (const char*, size_t);  
  
Date today = "04.11.2018."_date;
```