

Opseg važenja

- ❖ *Oblast* ili *opseg važenja* (*scope*) je deo teksta programa, ponekad diskontinualan (razdvojen delovima koda koji ne pripadaju tom opsegu) u kom je neko ime važeće, odnosno u kom se ono može koristiti neposredno, bez posebnih kvalifikacija, odnosno bez upotrebe operatora `::` ili nekog drugog operatora ispred tog imena, npr. `Clock::setTime` ili `pc->setTime`
- ❖ Pod *imenom* (*name*) se podrazumeva identifikator, puno ime operatorske funkcije (npr. `operator+` ili `operator new`), ime korisnički definisanog konverzionog operatora (npr. `operator bool`) ili ime šablona sa stvarnim argumentima (npr. `stack<int>`)
- ❖ U opsegu važenja nekog imena, prevodilac može da izvrši tzv. *nekvalifikovanu potragu* (*unqualified lookup*) za deklaracijom tog imena, određujući tako entitet na koji se to ime odnosi. Na primer:

```
class Clock {  
    ...  
    void setTime (int hh, int mm, int ss);  
    ...  
    int h, m, s;  
};  
  
void Clock::setTime (int hh, int mm, int ss) {  
    h = (hh >= 0 && hh <= 23) ? hh : 0;  
    m = (mm >= 0 && mm <= 59) ? mm : 0;  
    s = (ss >= 0 && ss <= 59) ? ss : 0;  
}
```

Identifikatori `setTime`, `h`, `m`, i `s` su deklarirani kao članovi klase, pa imaju opseg važenja klase

Identifikator `Clock` se ovde može koristiti nekvalifikovano, jer je ovo mesto u oblasti njegovog važenja. Međutim, identifikator `setTime` ne može ovde da se koristi nekvalifikovano

Identifikatori `hh`, `mm`, i `ss` su deklarirani kao parametri funkcije, pa imaju opseg važenja te funkcije

Identifikatori `hh`, `mm`, i `ss`, ali i `h`, `m`, i `s` se ovde mogu koristiti nekvalifikovano, jer je ovo kod unutar njihovog opsega važenja

Opseg važenja

- ❖ Nekvalifikovana pretraga imena podrazumeva da se za dato ime pretražuje određena oblast važenja i, ako se u njoj nađe deklaracija tog imena, prevodilac datu upotrebu imena vezuje za deklarisanu entitet; u suprotnom, eventualno pretražuje neku drugu, okružujuću oblast važenja i u njoj traži deklaraciju tog imena itd. Ukoliko takvu deklaraciju ne nađe, upotreba imena nije ispravna
- ❖ Ovakva pretraga najpre po tekućoj oblasti važenja, pa onda po okružujućoj, može da se shvati i kao činjenica da ime deklarirano u ugnežđenoj oblasti važenja *sakriva* isto ime u okružujućoj oblasti važenja.
Na primer:

```
int x = 0;
```

```
void f () {  
    int x = 1;  
  
    x = 3;  
}
```

```
struct Dummy {  
    int x;  
};
```

- ❖ Dakle, u oblasti važenja, ime se može koristiti neposredno, nekvalifikovano. Van oblasti se može koristiti ili samo na određene načine, ili nikako
- ❖ Pojam oblasti važenja je isključivo vezana za tekst programa i vreme prevođenja, dakle statički koncept, bez semantike koja se odnosi na vreme izvršavanja