

Poziv funkcije

- ❖ U suprotnom, ukoliko se poziv funkcije članice vrši preko objekta, poziv nije polimorfan:

```
void h (Base b) {  
    b.f();  
}
```

Objekat *b* je direktna instanca klase *Base* i sigurno ništa osim toga, pa poziv nije polimorfan. Ovakve pozive prevodilac može da reši statičkim vezivanjem (za vreme prevođenja)

```
void main () {  
    Derived d;  
    h(d);
```

Formalni parametar *b* je objekat klase *Base* i nezavisan je od stvarnog argumenta kojim se inicijalizuje prilikom poziva funkcije

```
    Base b;  
    h(b);  
}
```

Poziva se *Base::f()* u oba slučaja

- ❖ Dinamičko vezivanje je implementaciona tehnika kojom se realizuju polimorfni pozivi. Treba primetiti to da bi u ovakvim pozivima za objekat (ne preko pokazivača ili reference), čak i ako bi taj poziv bio izvršen dinamičkim vezivanjem, rezultat bio isti (jer VTP objekta tipa *Base* ukazuje na VT baš te klase). Za pozive preko pokazivača ili reference, prevodilac, u opštem slučaju, ne može znati šta se krije iza objekta, jer to može biti i promenljivo (npr. za parametre funkcija), pa će te pozive uvek implementirati dinamičkim vezivanjem

Glava 11: Životni vek varijabli

- ❖ Trajanje skladišta i životni vek
- ❖ Automatski životni vek
- ❖ Statički životni vek
- ❖ Dinamički životni vek
- ❖ Životni vek vezan za nit
- ❖ Privremeni objekti
- ❖ Ugrađeni objekti

