Hijerarhijska dekompozicija

- Samo u izuzetnim situacijama i sa posebnim razlogom, neka operacija ne treba da bude polimorfna, tj. potrebno je sprečiti njeno redefinisanje; takve operacije nazivaju se u mnogim jezicima *završnim* (*final*) i mogu se tako označiti
- Primer su operacije koje fiksiraju neki postupak (algoritam), tako da se on ne može promeniti (jer bi to moglo da uzrokuje propuste), ali dozvoljavaju da neke korake tog postupka izvedene klase definišu ili redefinišu ("kukice")
- I na jeziku C++ virtuelna funkcija može da se označi kao *final;* tada se ona ne može redefinisati u izvedenim klasama (prevodilac će prijaviti grešku u suprotnom):

```
class Controller {
public:
  virtual void main () final;
  …
};
```

Reč *final* je identifikator sa posebnim značenjem samo na ovim mestima, a nije rezervisana ključna reč (može se koristiti kao identifikator na drugim mestima)

- I klasa može biti *završna,* čime se sprečava njena dalja specijalizacija:

```
class ExtendedTranslator final : public Translator {
  …
};
```

Hijerarhijska dekompozicija

- Generalizacijom se uvode osnovne klase koje uopštavaju, apstrahuju, grupišu zajednička svojstva i ponašanje posebnih klasa
- Sama činjenica da neke klase imaju zajednička svojstva ili operacije može, ali ne mora biti dobar razlog za uvođenje njihove generalizacije

```
class Rectangle {
public:
  void draw (Viewport*);
  …
};

class Polyline {
public:
  void draw (Viewport*);
  …
};
```

- Osnovni motiv za generalizaciju jeste to što neka druga apstrakcija ili drugi deo softvera (klijent) ima potrebu da instance datih različitih pojedinačnih klasa *posmatra* i *koristi* na isti način, kroz isti uopšteni *interfejs*, ne praveći razliku između njih
- Na primer, u programu za crtanje dijagrama, na *crtežu* (*Drawing*) se nalaze figure različitih vrsta: pravougaonici (*Rectangle*), poligonalne linije (*Polyline*), krugovi (*Circle*) itd. Kako ih iscrtati?

```
class Circle {
public:
  void draw (Viewport*);
  …
};

void Drawing::draw (Viewport* vp)
{
  // How to draw figures
  // of different kind?
}

…
```