Klasa kao realizacija softverske mašine

```
class Character {
public:
  …
  virtual bool step ();
  …
protected:
  virtual void calcMove (int& dx, int& dy);
  int myX, myY;
};

class Pacman : public Character {
public:
  …
  virtual bool step ();
  …
};

class Ghost : public Character {
public:
  …
  virtual bool step ();
  …
};

class Engine {
public:
  …
  void run ();
  …
private:
  list<Character*> myChars;
  …
};
```

Zaštićeni (*protected*) članovi su dostupni u izvedenim klasama, ali ne i drugde van klase

```
void Egine::run () {
  while (!gameOver) {
    for (Character* c : this->myChars) {
      c->step();
      …
    }
  }
}

bool Ghost::step () {
  int dx, dy;
  this->calcMove(dx,dy);
  if (dx!=0 || dy!=0) {
    theField->clearGhostSprite(myX,myY);
    this->myX += dx;
    this->myY += dy;
    theField->drawGhostSprite(myX,myY);
  }
  return true;
}

bool Pacman::step () {
…
}
```

Klasa kao realizacija softverske mašine

- Problem: ako ponašanje nekog aktivnog objekta postane suviše složeno i dugotrajno, da bi se postigao privid paralelizma, to ponašanje se mora deliti na manje intervale; tada se kontekst (stanje) mora čuvati i prenositi između različitih poziva operacije *step*, pa programiranje postaje teško, a kod nepregledan
- Osim toga, šta ako aktivni objekat ne može ili ne treba da nastavi svoju aktivnost, odnosno treba da je suspenduje dok se ne ispuni neki uslov?
- Na primer: program za obradu teksta; korisnik pokreće operaciju snimanja dokumenta u fajl (*save*); da li čekati da se ova operacija završi i ne dozvoliti korisniku da bilo šta radi sve dok ona traje? Šta ako to potraje previše? Bespotrebno!
- Rešenje: ponašanje aktivnih objekata realizovati kao *niti* (*thread*)
- *Nit* (*thread*) je sekvenca izvršenih akcija (naredbi) koja teče *uporedo* sa drugim takvim sekvencama (nitima)
- Nit se kreira nad pozivom neke funkcije, a kontrola niti dalje ide kako diktira kod te funkcije i onoga što ona dalje poziva
- Svaka nit ima svoj *tok kontrole* (*control flow*), koji uključuje i automatske promenljive — svaka nit ima *svoj stek poziva*