```
SIMULA begin integer uninfected;

  activity sick person;
  begin integer day; Boolean symtoms; set environment;
    uninfected := uninfected-1; symptoms := false;
    hold(incubation); symptoms := true;
    for day:=1 step 1 until length do
    begin if draw(probtreat[day],ul)
      then activate new treatment(current);
      infect(Poisson(contacts,u2),environment);
      hold(1) end
    end sick person;

  procedure infect(n,S); value n; integer n; set S;
  begin integer i;
    for i := 1 step 1 until n do
    if draw(printf X uninfected / population,u3) then
    begin include(new sick person,S);
      activate last(S) end
    end infect;

  activity treatment(patient); element patient;
  begin element X;
    extract patient when sick person do
    if symptoms then
    begin terminate(patient);
      for X := first(environment) while exists(X) do
      activate new treatment(X) end
    else if draw(probmass,u4) then terminate(patient)
    end treatment;

  uninfected := population;
  activate new sick persion;  hold(simperiod)

end SIMULA
```

The Simula67 version:

```
Drawing("Simulating Pandemic Development",900,900) begin

SIMULATION begin
  real population=140;
  real incubation=3;
  integer length=50;
  real simperiod=40;
  real probmass=0.3;

  integer uninfected;
  real array probtreat(1:length);
  integer u1,u2,u3,u4,u5,u6;

  -- Statistics
  integer nInfected,nSick;

  Process class SickPerson;
  begin integer day; Boolean symptoms; real contacts; ref(Head) environment;

    procedure treatment; begin
      ref(Link) X;
      if not symptoms then begin
        for X :- environment.first while X in Process do X qua SickPerson.treatment;
      end else if draw(probmass,u4) then begin
        symptoms := false; nSick := nSick-1;
      end
    end treatment;

    environment :- new Head; uninfected := uninfected-1; symptoms := false;
    contacts := randint(1,6,u5); nInfected := nInfected+1;
    hold(incubation); symptoms := true; nSick := nSick+1;
    for day:=1 step 1 until length do if symptoms then begin
      hold(1);
      if draw(probtreat(day),u1) then treatment;
      infect(Poisson(contacts,u2),environment);
    end
  end SickPerson;

  procedure infect(n,S); value n; integer n; ref(Head) S;
  begin integer i;
    for i := 1 step 1 until n do
    if draw(uninfected / population,u3) then
    begin new SickPerson.into(S);
      activate S.last end
    end infect;

  Process class Reporter; begin
    long real prevTime,timeStep; ref(Curve) infected,sick;
    ref(Description) infectedLabel,sickLabel;
```

```
    setBackgroundColor(lightGray); setStroke(4.5);
    infected :- new Curve(yellow); sick :- new Curve(red);
    infectedLabel :- new Description(100,100,yellow,"Number of infected persons (accumulated)");
    sickLabel :- new Description(100,50,red,"Number of sick persons (not accumulated)");
    prevTime:=clocktime;
      while clocktime < prevTime+10 do;
    timeStep:=1; -- Seconds
    while true do begin
      infected.move(nInfected); sick.move(nSick);
      hold(1);
      while clocktime < prevTime+timeStep do;
      prevTime:=prevTime+timeStep;
    end
  end;

  class Description(x,y,color,txt); long real x,y; integer color; Text txt; begin
      setFillColor(color); fillRectangle(x, y, 50, 10);
      setDrawColor(black); setFontSize(24); drawText(txt,x+70,y+11);
  end;

  class Curve(color); value color; integer color; begin
    long real x,y,nextX,nextY;
    Procedure move(nxt); value nxt; long real nxt; begin
      nextX := x+20; nextY := 900-10-(nxt*5);
      setDrawColor(color); drawLine(x,y,nextX,nextY);
      x:=nextX; y:=nextY;
    end;
    x:=0; y:=900-10;
  end;

  begin -- Initiate variables
    integer i;
    uninfected := population;
    for i:=1 step 1 until length do probtreat(i) := uniform(0,0.5,u6);

    u1 := 4001;  -- Start values chosen to
    u2 := 4013;  -- avoid dependent series
    u3 := 4023;
    u4 := 4033;
    u5 := 4043;
    u6 := 4053;

  end;

  -- Start Simulation
  activate new Reporter;
  activate new SickPerson;  hold(simperiod);
  while true do;

end SIMULATION
end DRAWING
```