

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 6

дисциплина: Компьютерный практикум

по математическому моделированию

Студент: Ли Тимофей Александрович

Группа: НФИбд-01-18

МОСКВА

2021 г.

## Постановка задачи

Основной целью работы является освоение специализированных пакетов для решения задач в непрерывном и дискретном времени.

## Выполнение работы

Сначала выполнил все примеры к лабораторной работе №6:

1. Ознакомился с моделью экспоненциального роста.

```
import Pkg
Pkg.add("DifferentialEquations")
```

```
Updating registry at `C:\Users\Xiaomi\.julia\registries\General`
Updating git-repo `https://github.com/JuliaRegistries/General.git`
Resolving package versions...
No Changes to `C:\Users\Xiaomi\.julia\environments\v1.6\Project.toml`
No Changes to `C:\Users\Xiaomi\.julia\environments\v1.6\Manifest.toml`
```

```
using DifferentialEquations
a = 0.98
f(u,p,t) = a*u
u0 = 1.0
tspan = (0.0,1.0)
prob = ODEProblem(f,u0,tspan)
sol = solve(prob)
```

retcode: Success

Interpolation: specialized 4th order "free" interpolation, specialized 2nd order "free" stiffness-aware interpolation

t: 5-element Vector{Float64}:

```
0.0
0.10042494449239292
0.35218603951893646
0.6934436028208104
1.0
```

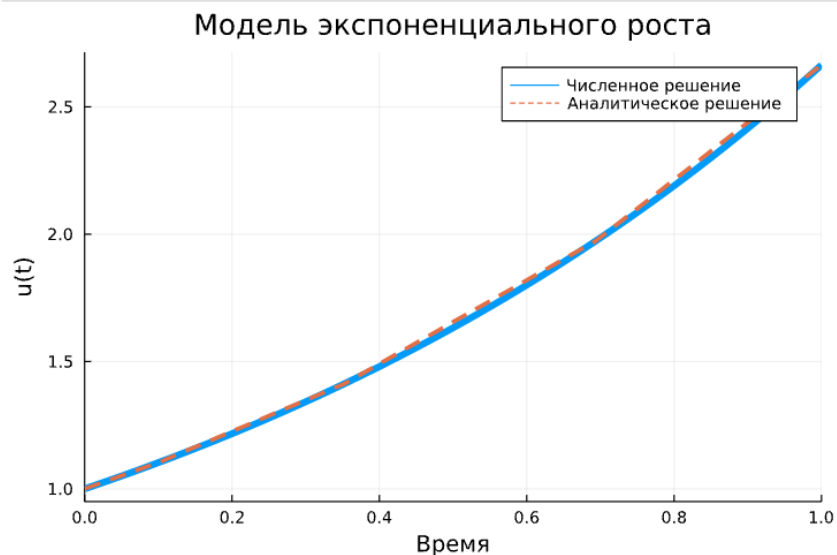
u: 5-element Vector{Float64}:

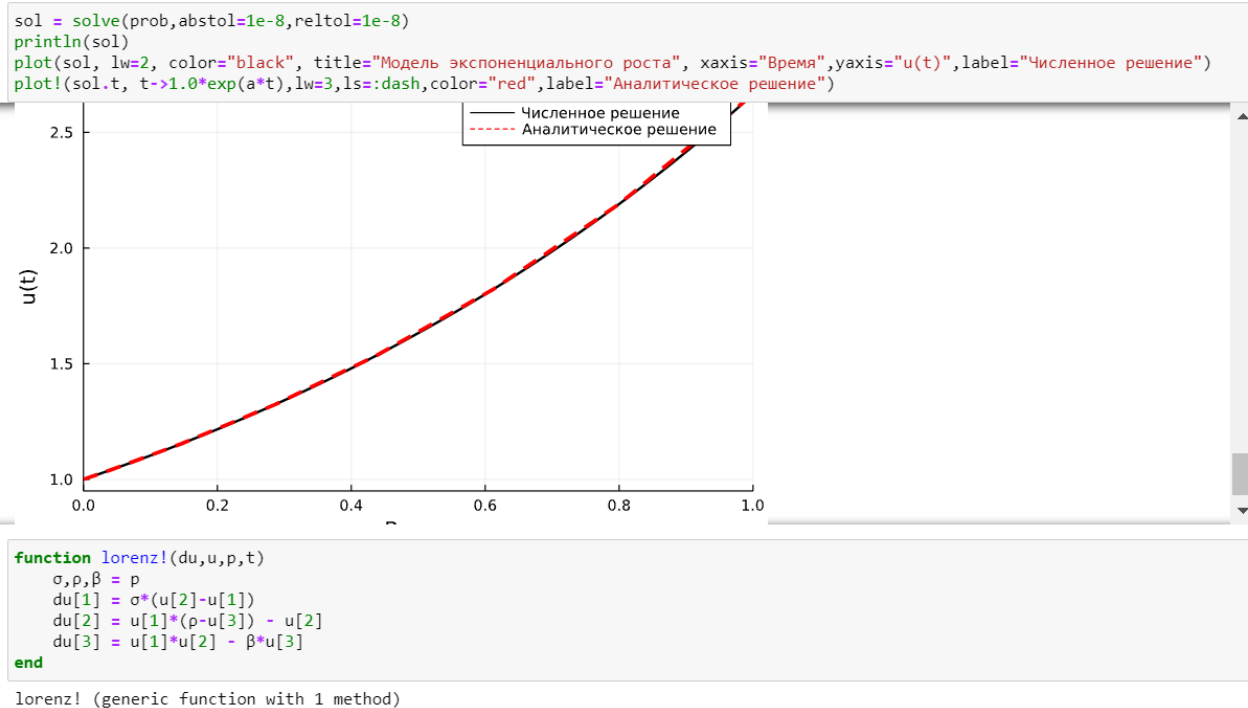
```
1.0
1.1034222047865465
1.4121908848175448
1.9730384275623003
2.664456142481452
```

```
Pkg.add("Plots")
using Plots
```

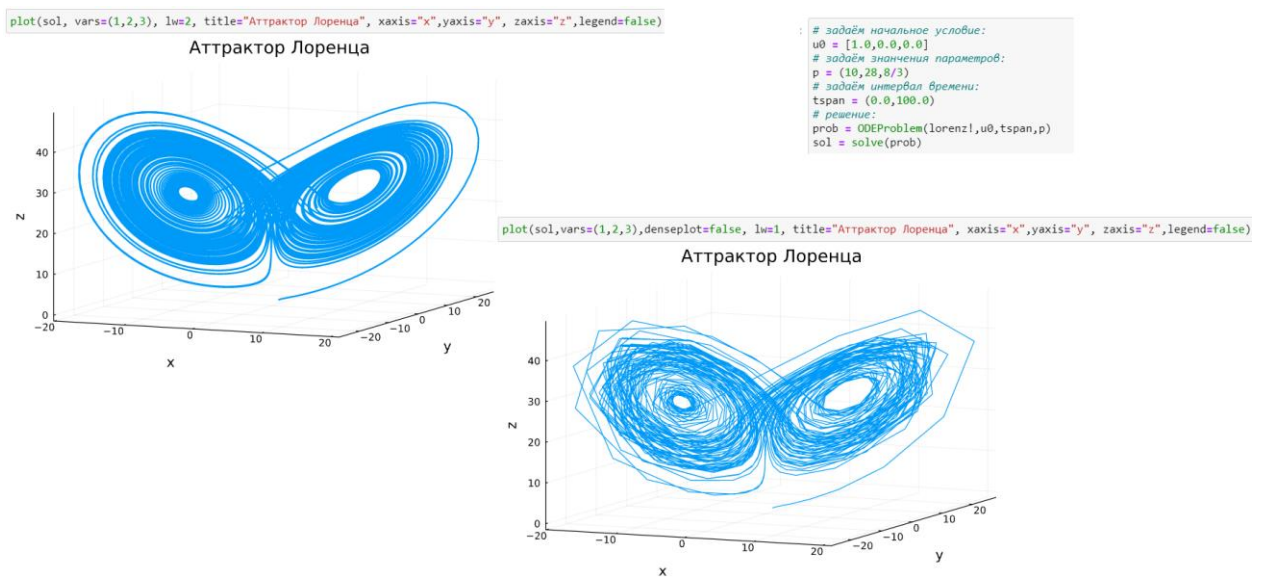
```
Resolving package versions...
No Changes to `C:\Users\Xiaomi\.julia\environments\v1.6\Project.toml`
No Changes to `C:\Users\Xiaomi\.julia\environments\v1.6\Manifest.toml`
```

```
plot(sol, linewidth=5, title="Модель экспоненциального роста", xaxis="Время", yaxis="u(t)", label="Численное решение")
plot!(sol.t, t->1.0*exp(a*t), lw=3, ls=:dash, label="Аналитическое решение")
```



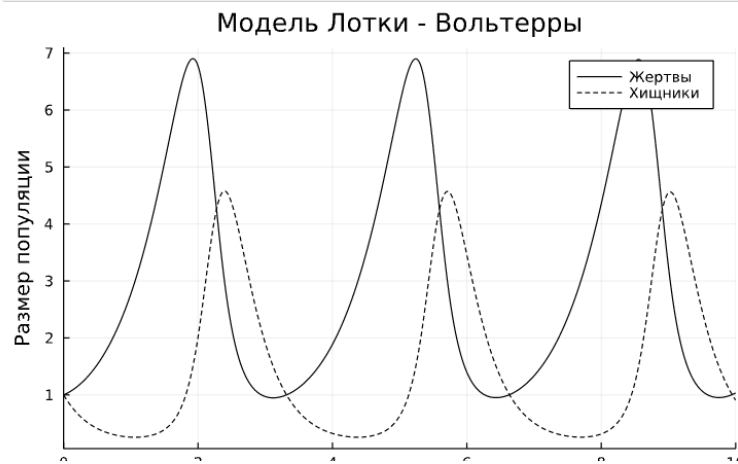


2. Затем ознакомился с системой Лоренца (начало на прошлом скриншоте). Также построил график с отключенной интерполяцией.

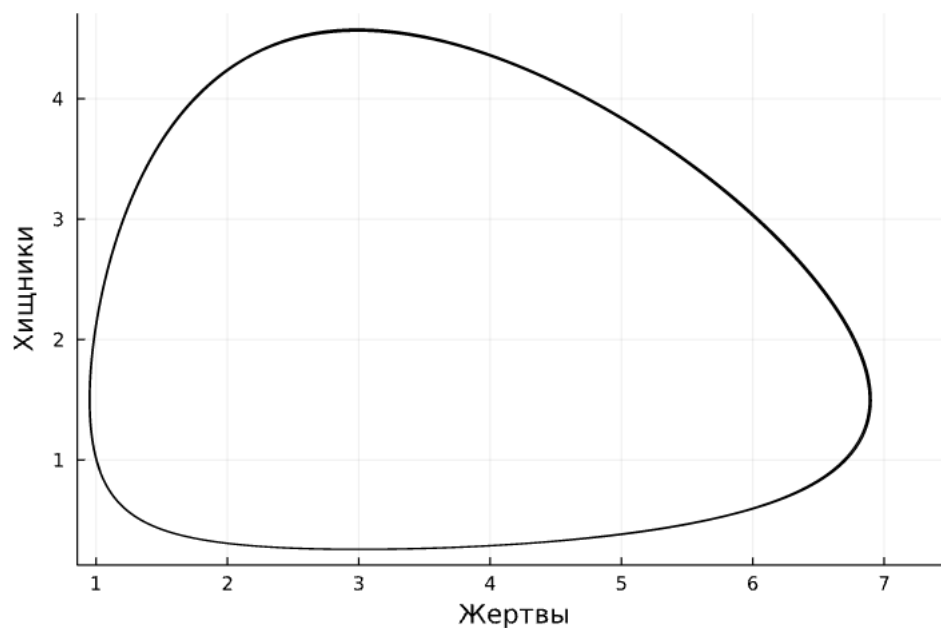


3. Далее ознакомился с моделью Лотки-Волтерры, а также построил фазовый портрет для данной модели.

```
lv! = @ode_def LotkaVolterra begin
dx = a*x - b*x*y
dy = -c*y + d*x*y
end a b c d
# задаём начальное условие:
u0 = [1.0,1.0]
# задаём значения параметров:
p = (1.5,1.0,3.0,1.0)
# задаём интервал времени:
tspan = (0.0,10.0)
prob = ODEProblem(lv!,u0,tspan,p)
sol = solve(prob)
plot(sol, label = ["Жертвы" "Хищники"], color="black", ls=[:solid :dash], title="Модель Лотки - Вольтерры", xaxis="Время",yaxis="Размер популяции")
```



```
plot(sol,vars=(1,2), color="black", xaxis="Жертвы",yaxis="Хищники", legend=false)
```



Далее выполнил поставленные задачи.

1. Реализовал модель Мальтуса. По сути, она строится как модель экспоненциального роста из примеров, поэтому код взял оттуда. Также аналогично прошлой лабораторной работе добавил код для сохранения графика в виде гиф-анимации. Значения для графика выбирал такие, чтобы график выглядел адекватно.

```

function maltus(u0, b, c, tmax)
    n=tmax
    a = b-c
    f1(u,p,t) = a*u
    tspan1 = (0.0,n)
    prob1 = ODEProblem(f1,u0,tspan1)
    sol1 = solve(prob1)
    anim = @animate for i in 0:0.01:n
        a = b-c
        f(u,p,t) = a*u
        tspan = (0.0,i)
        prob = ODEProblem(f,u0,tspan)
        sol = solve(prob)
        plot(xlim=(0,n),ylim=(0,maximum(sol1)), sol, linewidth=5,title="Модель Мальтуса", хaxis="Время",уaxis="численность",label=i)
    end
    gif(anim,"maltus.gif")
end

```

```
maltus(10, 10, 5, 1)
```

```

Info: Saved animation to
  fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по модели
  @ Plots C:\Users\Xiaomi\.julia\packages\Plots\PomtQ\src\animation.jl:114

```



- Далее по схожей схеме реализовал логистическую модель роста популяции. Также добавил условие корректности вводимых значений  $r$  и  $k$ .

```

function logist(u0, r, k, tmax)
    if r<0 || k<0
        println("incorrect operands")
    else
        n=tmax
        f1(u,p,t) = r*u*(1-u/k)
        tspan1 = (0.0,n)
        prob1 = ODEProblem(f1,u0,tspan1)
        sol1 = solve(prob1)
        anim = @animate for i in 0:0.01:n
            f(u,p,t) = r*u*(1-u/k)
            tspan = (0.0,i)
            prob = ODEProblem(f,u0,tspan)
            sol = solve(prob)
            plot(xlim=(0,n),ylim=(0,maximum(sol1)), sol, linewidth=5,title="Логистическая модель", хaxis="Время",уaxis="численность")
        end
        gif(anim,"logist.gif")
    end
end
end

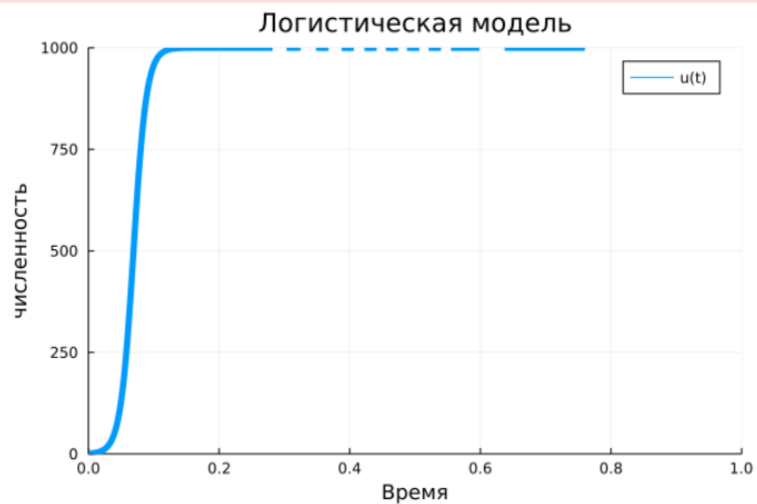
```

```
logist(1, 100, 1000, 1)
```

```

Info: Saved animation to
  fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по модели
  @ Plots C:\Users\Xiaomi\.julia\packages\Plots\PomtQ\src\animation.jl:114

```



- Далее реализовал модель SIR. Здесь мы имеем дело с системой ДУ, поэтому добавил функцию, собирающую эту систему. Значения для графика выбирал, чтобы он смотрелся красиво (модель проходили в прошлом году).

```

function sir(n, i0, r0, b, v, tmax)
    s0=n-i0-r0
    function syst(du,u,p,t)
        du[1]=-b*u[1]*u[2]
        du[2]=b*u[1]*u[2]-v*u[2]
        du[3]=v*u[2]
    end
    u0=[s0,i0,r0]
    tspan1=(0.0,tmax)
    prob1=ODEProblem(syst,u0,tspan1)
    sol1=solve(prob1)
    anim = @animate for i in 1:tmax
        tspan=(0.0,i)
        prob=ODEProblem(syst,u0,tspan)
        sol=solve(prob)
        plot(xlim=(0,tmax),ylim=(0,maximum(sol1)), sol, linewidth=5,title="Модель SIR", xaxis="Время",yaxis="численность",label=
    end
    gif(anim,"sir.gif")
end

```

```

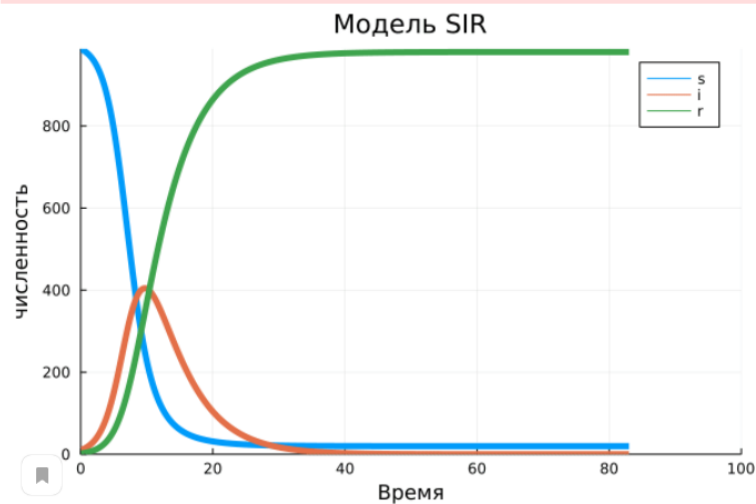
sir(1000, 10, 2, 0.0008, 0.2, 100)

```

```

Info: Saved animation to
fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по модели
@ Plots C:\Users\Xiaomi\.julia\packages\Plots\PomtQ\src\animation.jl:114

```



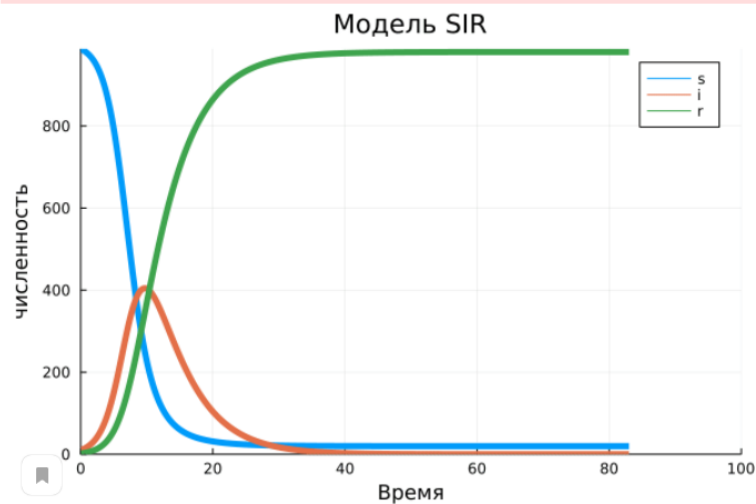
- Затем, перешел к модели SEIR. Для этого немного откорректировал код из прошлого пункта. По сравнению с SIR видим, что число людей в инкубационном периоде чуть снижает число больных людей, а в остальном картинка остается примерно той же.

```

function sir(n, i0, r0, b, v, tmax)
    s0=n-i0-r0
    function syst(du,u,p,t)
        du[1]=-b*u[1]*u[2]
        du[2]=b*u[1]*u[2]-v*u[2]
        du[3]=v*u[2]
    end
    u0=[s0,i0,r0]
    tspan=(0.0,tmax)
    prob1=ODEProblem(syst,u0,tspan1)
    sol1=solve(prob1)
    anim = @animate for i in 1:tmax
        tspan=(0.0,i)
        prob=ODEProblem(syst,u0,tspan)
        sol=solve(prob)
        plot(xlim=(0,tmax),ylim=(0,maximum(sol1)), sol, linewidth=5,title="Модель SIR", хaxis="Время",уaxis="численность",label=
    end
    gif(anim,"sir.gif")
end

```

Info: Saved animation to  
 fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по модели  
 @ Plots C:\Users\Xiaomi\.julia\packages\Plots\PomtQ\src\animation.jl:114



5. Попробовал реализовать дискретную модель Лотки-Вольтерры согласно условию, однако получился очень ломанный график. Что-то схожее с истиной тут есть, но до нее я в итоге не дошел.

```

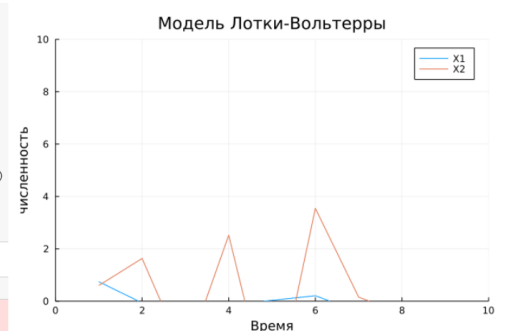
function lv(x0, y0, tmax)
    n=tmax
    a=2
    c=1
    d=5
    x=zeros(1,tmax)
    y=zeros(1,tmax)
    x[1]=x0
    y[1]=y0
    anim = @animate for i in 1:n-1
        x[i+1]=a*x[i]*(1-x[i])-x[i]*y[i]
        y[i+1]=c*y[i]+d*x[i]-y[i]
        plot(xlim=(0,n),ylim=(0,10), x[1:i],title="Модель Лотки-Вольтерры", хaxis="Время",уaxis="численность",label="X1")
        plot!(y[1:i], label="X2")
    end
    gif(anim,"lv.gif")
end

```

lv (generic function with 1 method)

lv(0.744, 0.6, 10)

Info: Saved animation to  
 fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по моделированию\lab6\lv.gif  
 @ Plots C:\Users\Xiaomi\.julia\packages\Plots\PomtQ\src\animation.jl:114



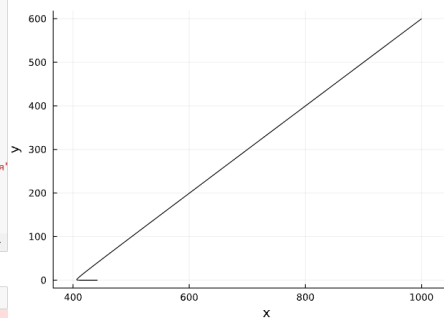
6. Построил модель отбора на основе конкурентных отношений и фазовый портрет.



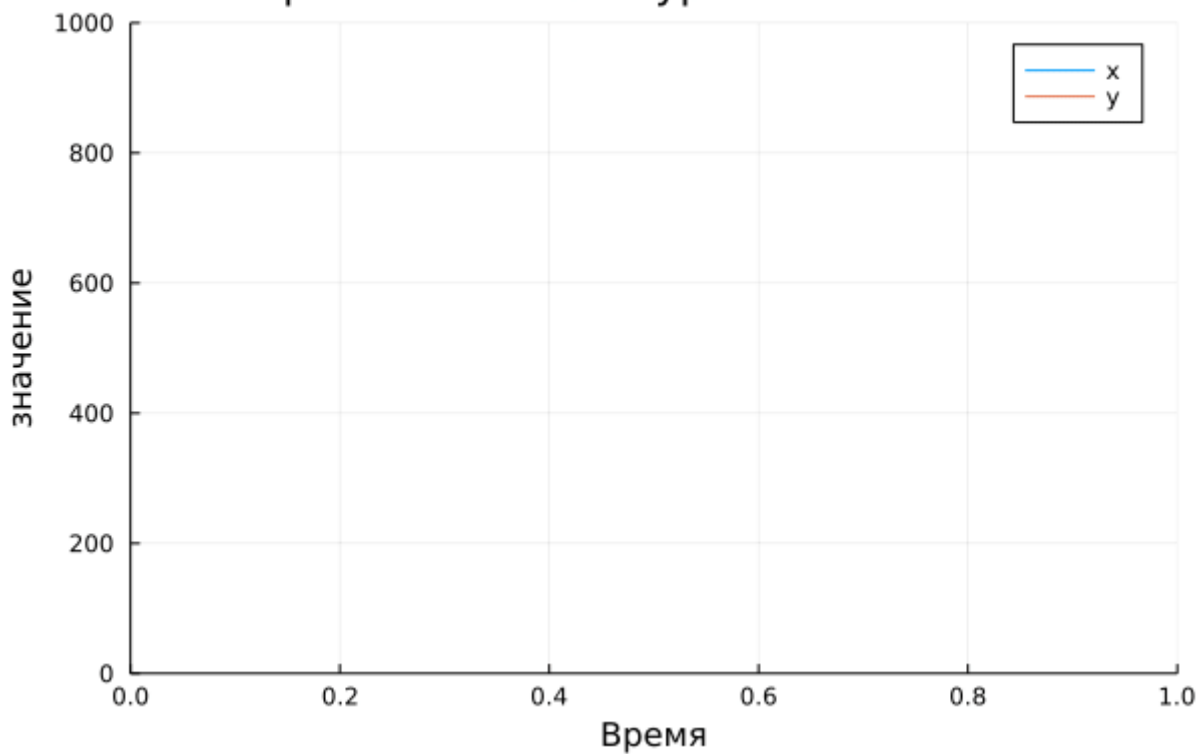
```
function m6(x0, y0, a, b, tmax)
    function sys(du,u,p,t)
        du[1]=a*u[1]-b*u[1]*u[2]
        du[2]=a*u[2]-b*u[1]*u[2]
    end
    u0=[x0,y0]
    tspan=(0.0,tmax)
    prob=ODEProblem(sys,u0,tspan)
    sol=solve(prob)
    anim = @animate for i in 0:0.01:tmax
        tspan=(0.0,i)
        prob=ODEProblem(sys,u0,tspan)
        sol=solve(prob)
        plot(xlim=(0,tmax),ylim=(0,maximum(sol)), sol, linewidth=5,title="отбор на основе конкурентных отношений", хaxis="Время")
    end
    gif(anim,"m6.gif")
    plot(sol,vars=(1,2), color="black", хaxis="x",yaxis="y", legend=false)
end

m6(1000, 600, 0.1, 0.1, 1)
```

Info: Saved animation to  
 fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по моделированию\lab6\m6.gif  
 @ Plots C:\Users\Xiaomi\julia\packages\Plots\PomtQ\src\animation.jl:114



## отбор на основе конкурентных отношений

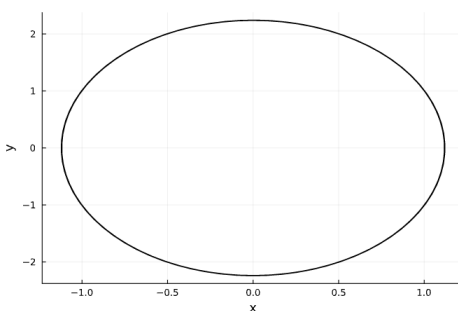


## 7. Построил модель консервативного осциллятора и фазовый портрет.

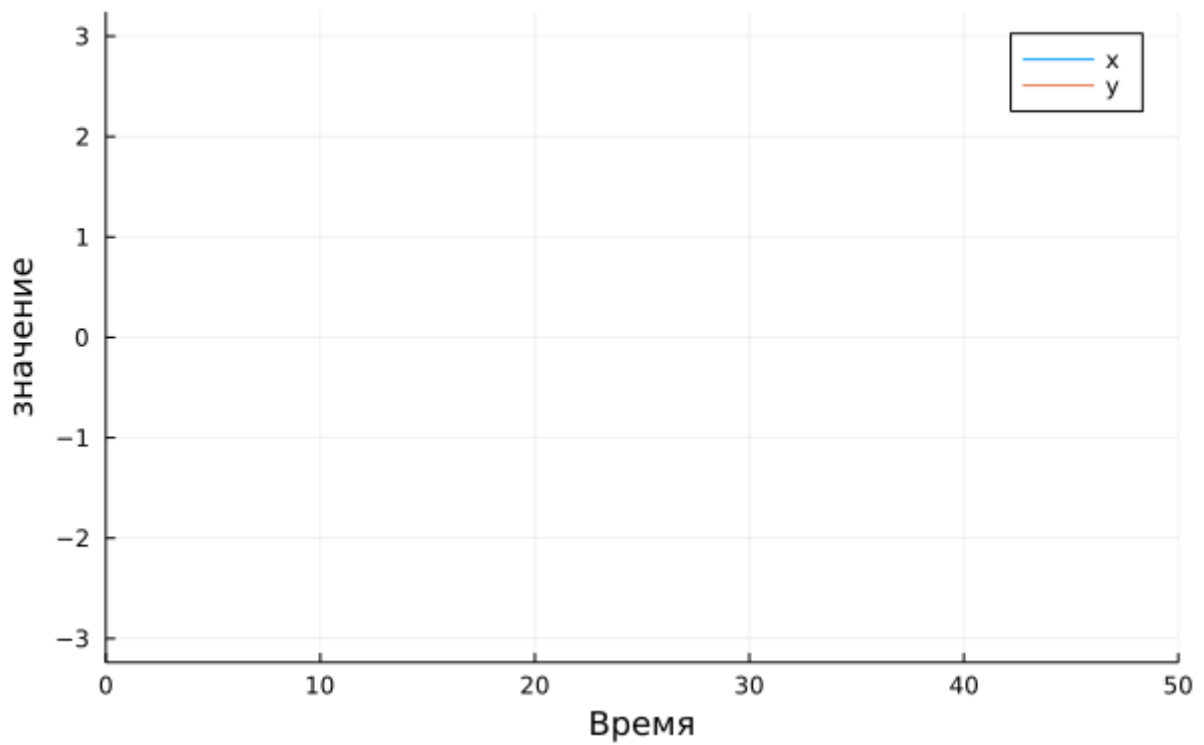
```
function m7(x0, y0, w, tmax)
    function sy(du,u,p,t)
        du[1]=w*u[2]
        du[2]=-w*u[1]
    end
    u0=[x0,y0]
    tspan=(0.0,tmax)
    prob=ODEProblem(sy,u0,tspan)
    sol=solve(prob)
    anim = @animate for i in 0:tmax
        tspan=(0.0,i)
        prob=ODEProblem(sy,u0,tspan)
        sol=solve(prob)
        plot(xlim=(0,tmax),ylim=(minimum(sol)-1,maximum(sol)+1), sol, linewidth=5,title="консервативный осциллятор", хaxis="Вп")
    end
    gif(anim,"m7.gif")
    plot(sol,vars=(1,2), color="black", хaxis="x",yaxis="y", legend=false)
end

m7(1,1,2,50)
```

Info: Saved animation to  
 fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по моделированию\lab6\m7.gif  
 @ Plots C:\Users\Xiaomi\julia\packages\Plots\PomtQ\src\animation.jl:114



## консервативный осциллятор

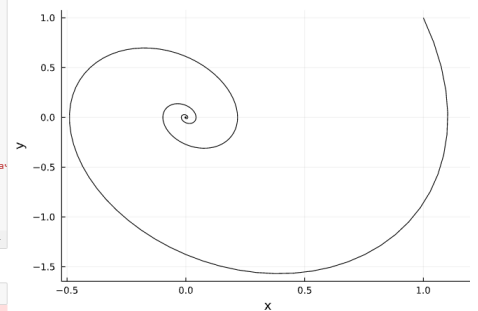


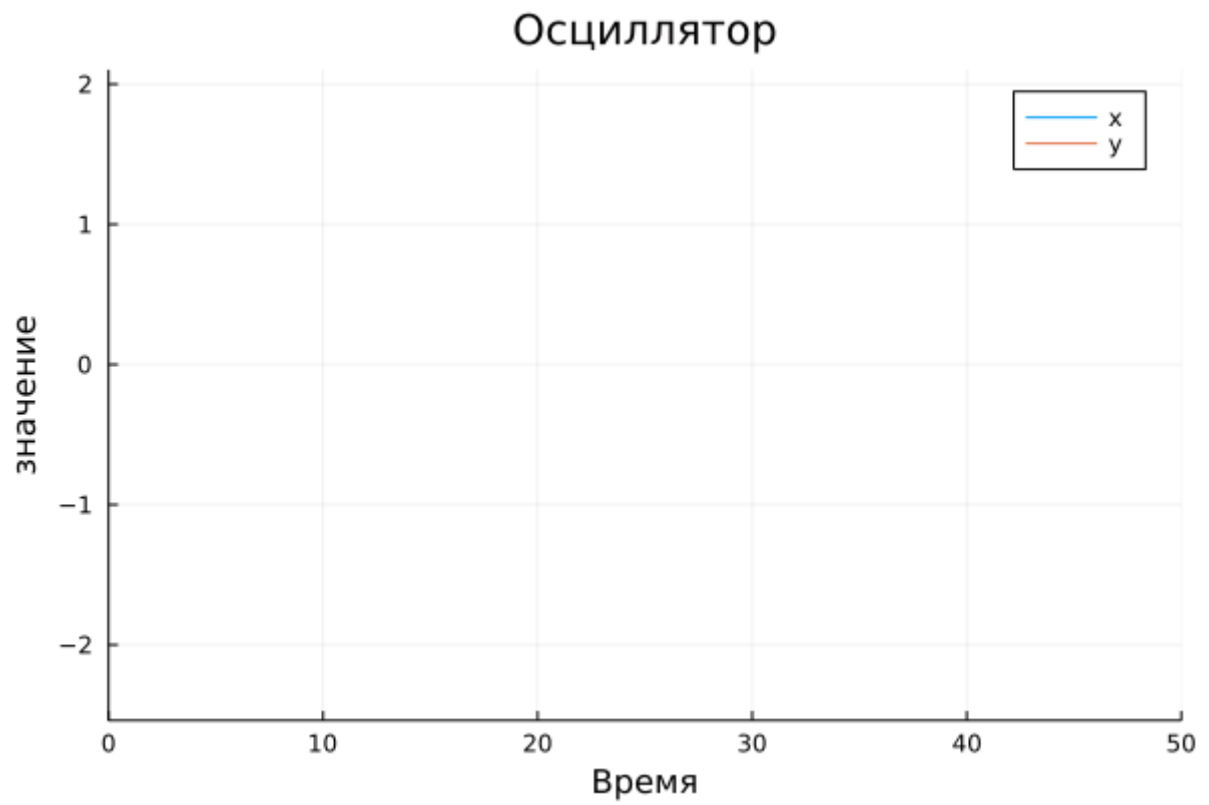
8. Добавив затухания, реализовал модель осциллятора.

```
function m8(x0, y0, g, w, tmax)
    function syy(du, u, p, t)
        du[1] = u[2]
        du[2] = -w^2*u[1] - g*u[2]
    end
    u0 = [x0, y0]
    tspan1 = (0.0, tmax)
    prob1 = ODEProblem(syy, u0, tspan1)
    sol1 = solve(prob1)
    anim = @animate for i in 0:tmax
        tspan2 = (0.0, i)
        prob2 = ODEProblem(syy, u0, tspan2)
        sol2 = solve(prob2)
        plot(xlim=(0, tmax), ylim=(minimum(sol1)-1, maximum(sol1)+1), sol, linewidth=5, title="Осциллятор", xaxis="Время", yaxis="значение")
    end
    gif(anim, "m8.gif")
    plot(sol1, vars=(1,2), color="black", xaxis="x", yaxis="y", legend=false)
end

m8(1,1,1,2,50)
```

Info: Saved animation to  
fn = C:\Users\Xiaomi\OneDrive\Документы\work\2021-2022\Практикум по моделированию\lab6\m8.gif  
@ Plots C:\Users\Xiaomi\julia\packages\Plots\PlotQ\src\animation.jl:114





## Выводы

Освоил специализированные пакеты для решения задач в непрерывном и дискретном времени.