[1]:

*#health status:*

SUCCESS = 0

HEALTHY = 1

SICK = 2

INFECTIOUS = 3

IMMUNITY = 4

DEAD = 5

[2]:

*#human role:*

STUDENT = 0

TEACHER = 1

[3]:

**from calendar import** Calendar

**import random**

[4]:

**class Human**:

**def** init ( self, role, come\_to\_university, meeting\_person\_expected\_value, meeting\_person\_dispersion, incubation\_period\_expected\_value, incubation\_period\_dispersion, mortality, illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value, immunitet\_period\_dispersion):

self.health = HEALTHY self.role = role

self.come\_to\_university = come\_to\_university self.meeting\_person\_expected\_value = meeting\_person\_expected\_value self.meeting\_person\_dispersion = meeting\_person\_dispersion self.time\_period = -1

self.mortality = mortality

self.incubation\_period\_expected\_value = incubation\_period\_expected\_value self.incubation\_period\_dispersion = incubation\_period\_dispersion self.illness\_time\_expected\_value = illness\_time\_expected\_value self.illness\_time\_dispersion = illness\_time\_dispersion self.immunitet\_period\_expected\_value = immunitet\_period\_expected\_value self.immunitet\_period\_dispersion = immunitet\_period\_dispersion

**def** infectious(self, infectiouness):

**if** self.health == HEALTHY **and** random.uniform(0, 1) < infectiouness: self.health = INFECTIOUS

self.time\_period = int(random.normalvariate( self.incubation\_period\_expected\_value, self.incubation\_period\_dispersion))

##### return True return False

**def** process(self):

**if** self.health == INFECTIOUS: self.time\_period -= 1

**if** self.time\_period <= 0: self.health = SICK

self.time\_period = int(random.normalvariate( self.illness\_time\_expected\_value, self.illness\_time\_dispersion

))

**return** self.health

**if** self.health == SICK: self.time\_period -= 1

**if** self.time\_period <= 0:

**if** random.uniform(0, 1) < self.mortality: self.health = DEAD

##### else:

self.health = IMMUNITY

self.time\_period = int(random.normalvariate( self.immunitet\_period\_expected\_value, self.immunitet\_period\_dispersion

))

**return** self.health

**if** self.health == IMMUNITY: self.time\_period -= 1

**if** self.time\_period <= 0: self.health = HEALTHY self.time\_period = -1 **return** self.health

**return** SUCCESS

**def** is\_come(self):

**if** self.health != SICK **and** \

random.uniform(0, 1) < self.come\_to\_university:

##### return True return False

**def** get\_meetings(self):

meetings = int(random.normalvariate(

self.meeting\_person\_expected\_value, self.meeting\_person\_dispersion))

**if** meetings > 0:

**return** meetings

**return** 0

[5]: **class Population**:

**def** init ( self, students\_num, student\_meeting\_person\_expected\_value, student\_meeting\_person\_dispersion, student\_come\_to\_university, teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion, teacher\_come\_to\_university, infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion, mortality, illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value, immunitet\_period\_dispersion

):

self.human\_list = []

*#add students*

**for** i **in** range(students\_num): self.human\_list.append( Human(STUDENT,

student\_come\_to\_university, student\_meeting\_person\_expected\_value, student\_meeting\_person\_dispersion, incubation\_period\_expected\_value, incubation\_period\_dispersion, mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value, immunitet\_period\_dispersion)

)

*#add teachers*

**for** i **in** range(teachers\_num): self.human\_list.append( Human(TEACHER,

teacher\_come\_to\_university, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion, incubation\_period\_expected\_value, incubation\_period\_dispersion, mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value, immunitet\_period\_dispersion)

)

self.dead = 0

self.healthy = len(self.human\_list) self.infectious = 0

self.sick = 0

self.immunity = 0 self.mortality = mortality

self.infectiousness = infectiousness

**def** process( self ):

*#create comming list* come\_to\_university = list() **for** human **in** self.human\_list:

result = human.process()

**if** result == DEAD: self.dead += 1

self.sick -= 1 self.human\_list.remove(human) **continue**

**elif** result == SICK: self.infectious -= 1

self.sick += 1

##### continue

**elif** result == IMMUNITY: self.sick -= 1

self.immunity += 1

##### continue

**elif** result == HEALTHY: self.immunity -= 1

self.healthy += 1

##### continue

**if** human.is\_come(): come\_to\_university.append(human)

**if** len(come\_to\_university) == 0:

##### return

*#meetings*

**for** human **in** come\_to\_university:

**if**(human.health == INFECTIOUS):

**for** i **in** range(human.get\_meetings()):

**if** random.choice(come\_to\_university).infectious(self.

*<→*infectiousness):

self.healthy -= 1

self.infectious += 1

**def** process\_weekend( self ):

**for** human **in** self.human\_list: result = human.process() **if** result == DEAD:

self.dead += 1

self.sick -= 1 self.human\_list.remove(human)

**elif** result == SICK: self.infectious -= 1

self.sick += 1

**elif** result == IMMUNITY: self.sick -= 1

self.immunity += 1

**elif** result == HEALTHY: self.immunity -= 1

self.healthy += 1

**def** first\_infection( self ):

human = random.choice(self.human\_list) human.infectious(1.)

self.healthy -= 1

self.infectious += 1

**def** add\_infectious( self, human): human.infectious(self.infectiousness) self.healthy -= 1

self.infectious += 1

[6]:

**import datetime**

[7]:

**def** autum\_semestr\_loop( population, drawer):

**try**:

calendar = Calendar()

my\_calendar = calendar.yeardayscalendar( year = 2020 )

**for** season **in** range(2, 5):

**for** month **in** range(3):

**if** (season)\*3+1+ month < 9:

**continue**

**if** season == 4 **and** month > 0:

**break**

**for** week **in** range(len(my\_calendar[season][month])):

**for** day **in** range(7):

**if not** my\_calendar[season][month][week][day] == 0:

**if** day == 6:

*#print('sunday')*

population.process\_weekend()

**else**:

*#print('{} - {}'.format(*

*#my\_calendar[season][month][week][day],*

*#(season)\*3+1+ month*

*#))*

*#print(str(season)+' '+str(month)+'*␣

*<→'+str(week)+' '+str(day))*

population.process()

*#print('{} {} {} {} {}'.format(*

*# population.healthy,*

*# population.infectious,*

*# population.sick,*

*# population.immunity,*

*# population.dead*

*# ))*

date = datetime.date(year=2020,

month=(season)\*3 +1 + month,

␣

*<→*day=my\_calendar[season][month][week][day])

drawer.add(population, date)

**except IndexError**: **return**

[8]:

**import matplotlib.pyplot as plt**

%**matplotlib** inline

[9]:

**class Drawer2D**:

**def** init (self): self.healthy = [] self.infectious = [] self.sick = [] self.dead = [] self.date = [] self.immunity = []

**def** add(self, population, date): self.healthy.append(population.healthy+population.immunity) self.infectious.append(population.infectious) self.sick.append(population.sick)

self.dead.append(population.dead)

self.immunity.append(population.immunity) self.date.append(date)

**def** draw(self):

fig, ax = plt.subplots(figsize=(15, 9))

ax.plot(self.date,

ax.plot(self.date, ax.plot(self.date, ax.plot(self.date, ax.plot(self.date, ax.legend()

self.healthy, 'g', label='

self.infectious, 'b', label='

')

')

self.sick, 'r', label='

')

self.dead, 'black', label='

self.immunity, 'c', label='

')

')

ax.set\_xlabel('

ax.set\_ylabel(' ax.set\_title("

')

')

")

[10]: **def** experiment1():

students\_num = 100

student\_meeting\_person\_expected\_value = 10

student\_meeting\_person\_dispersion = 2

student\_come\_to\_university = 0.9

teachers\_num = 0

teacher\_meeting\_person\_expected\_value = 0

teacher\_meeting\_person\_dispersion = 0

teacher\_come\_to\_university = 0

infectiousness = 0.2

incubation\_period\_expected\_value = 14

incubation\_period\_dispersion = 2

mortality = 0.1

illness\_time\_expected\_value = 20

illness\_time\_dispersion = 15

immunitet\_period\_expected\_value = 40

immunitet\_period\_dispersion = 10

population = Population(students\_num, student\_meeting\_person\_expected\_value, student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() drawer2D = Drawer2D()

autum\_semestr\_loop( population, drawer2D)

**return** drawer2D

[11]: **def** experiment2():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100

population = Population(students\_num, student\_meeting\_person\_expected\_value, student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() drawer2D = Drawer2D()

autum\_semestr\_loop( population, drawer2D)

**return** drawer2D

[12]: **def** experiment3():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799\*0.7 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

[*#https://www.kommersant.ru/doc/4432704*](http://www.kommersant.ru/doc/4432704)

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100

population = Population(students\_num, student\_meeting\_person\_expected\_value, student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() drawer2D = Drawer2D()

autum\_semestr\_loop( population, drawer2D)

**return** drawer2D

#### 

[13]: **def** experiment4():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799\*0.7 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

[*#https://www.kommersant.ru/doc/4432704*](http://www.kommersant.ru/doc/4432704)

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 20

immunitet\_period\_dispersion = 10

population = Population(students\_num, student\_meeting\_person\_expected\_value, student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() drawer2D = Drawer2D()

autum\_semestr\_loop( population, drawer2D)

**return** drawer2D

[14]:

**import pylab**

**from mpl\_toolkits.mplot3d import** Axes3D

**import numpy as np**

[15]: **class Drawer3D**:

**def** init (self, k\_name): self.Y = []

self.X = [] self.date = [] self.Zhealthy = []

self.Zinfectious = [] self.Zsick = [] self.Zdead = [] self.Zimmunity = [] self.healthy = [] self.infectious = [] self.sick = [] self.dead = [] self.immunity = [] self.k\_name = k\_name

**def** setK(self, k):

self.X.append([i **for** i **in** range(len(self.date))]) self.Y.append([k **for** i **in** range(len(self.date))].copy()) self.Zhealthy.append(self.healthy.copy()) self.Zinfectious.append(self.infectious.copy()) self.Zsick.append(self.sick.copy()) self.Zdead.append(self.dead.copy()) self.Zimmunity.append(self.immunity.copy())

self.date = [] self.healthy = [] self.infectious = [] self.sick = [] self.dead = [] self.immunity = []

**def** add( self, population, date): self.date.append(date) self.healthy.append(population.healthy) self.infectious.append(population.infectious) self.sick.append(population.sick) self.dead.append(population.dead) self.immunity.append(population.immunity)

**def** draw(self):

x = np.array(self.X) y = np.array(self.Y)

zHealthy = np.array(self.Zhealthy) zInfectious = np.array(self.Zinfectious) zImmunity = np.array(self.Zimmunity) zDead = np.array(self.Zdead)

zSick = np.array(self.Zsick)

fig = plt.figure(figsize=(15, 15))

ax = fig.add\_subplot(3, 2, 1, projection='3d') ax.set\_title(' ')

ax.set\_xlabel(' ') ax.set\_ylabel(self.k\_name) ax.set\_zlabel(' ')

surf = ax.plot\_surface(x, y, zHealthy)

ax = fig.add\_subplot(3, 2, 2, projection='3d') ax.set\_title(' ')

ax.set\_xlabel(' ') ax.set\_ylabel(self.k\_name) ax.set\_zlabel(' ')

surf = ax.plot\_surface(x, y, zInfectious)

ax = fig.add\_subplot(3, 2, 3, projection='3d') ax.set\_title(' ')

ax.set\_xlabel(' ') ax.set\_ylabel(self.k\_name) ax.set\_zlabel(' ')

surf = ax.plot\_surface(x, y, zSick)

ax = fig.add\_subplot(3, 2, 4, projection='3d') ax.set\_title(' ')

ax.set\_xlabel(' ') ax.set\_ylabel(self.k\_name) ax.set\_zlabel(' ')

surf = ax.plot\_surface(x, y, zImmunity)

ax = fig.add\_subplot(3, 2, 5, projection='3d') ax.set\_title(' ')

ax.set\_xlabel(' ') ax.set\_ylabel(self.k\_name) ax.set\_zlabel(' ')

surf = ax.plot\_surface(x, y, zDead)

[16]:

%%time

students\_num = 100

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 0

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100 drawer3D = Drawer3D(' ')

**for** infectiousness **in** np.arange(0.01, 0.4, 0.02):

population = Population(students\_num, student\_meeting\_person\_expected\_value, student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

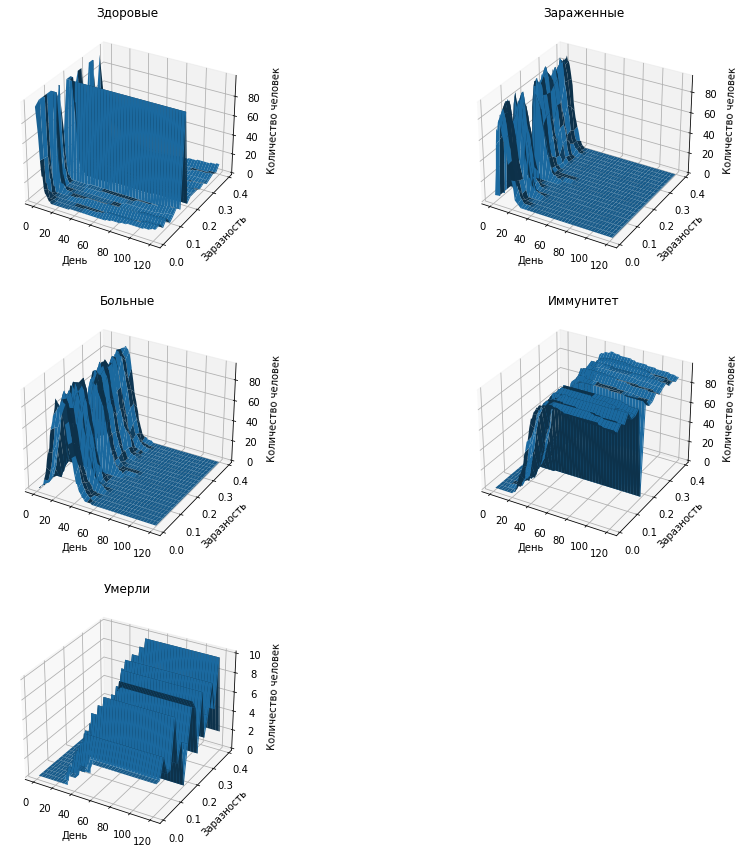
illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D) drawer3D.setK(infectiousness)

drawer3D.draw()

CPU times: user 866 ms, sys: 177 µs, total: 866 ms Wall time: 887 ms



[17]: **def** experiment5():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100 drawer3D = Drawer3D(" ")

**for** infectiousness **in** np.arange(0.0, 0.95, 0.05): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D) drawer3D.setK(infectiousness)

**return** drawer3D

[18]: **def** experiment6():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100

drawer3D = Drawer3D(" ")

**for** student\_come\_to\_university **in** np.arange(0., 0.95, 0.05): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D) drawer3D.setK(student\_come\_to\_university)

**return** drawer3D

[19]: **def** experiment7():

students\_num = 19000

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100 drawer3D = Drawer3D(" ")

**for** student\_meeting\_person\_expected\_value **in** np.arange(10, 200, 10): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D)

drawer3D.setK(student\_meeting\_person\_expected\_value)

**return** drawer3D

[20]: **def** experiment8():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150 infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100

drawer3D = Drawer3D(" ")

**for** teacher\_come\_to\_university **in** np.arange(0.05, 1, 0.05): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D) drawer3D.setK(teacher\_come\_to\_university)

**return** drawer3D

[21]: **def** experiment9():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100

drawer3D = Drawer3D(" ")

**for** incubation\_period\_expected\_value **in** np.arange(1, 20, 1): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D)

drawer3D.setK(incubation\_period\_expected\_value)

**return** drawer3D

[22]: **def** experiment10():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100 drawer3D = Drawer3D(" ")

**for** mortality **in** np.arange(0, 1, 0.05): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D) drawer3D.setK(mortality)

**return** drawer3D

[23]: **def** experiment11():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_dispersion = 100

drawer3D = Drawer3D(" ")

**for** immunitet\_period\_expected\_value **in** np.arange(10, 100, 10): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D) drawer3D.setK(immunitet\_period\_expected\_value)

**return** drawer3D

[24]: **def** experiment12():

students\_num = 19000

student\_meeting\_person\_expected\_value = 150

student\_meeting\_person\_dispersion = 50

student\_come\_to\_university = 0.5

teachers\_num = 3297

teacher\_meeting\_person\_expected\_value = 200

teacher\_meeting\_person\_dispersion = 150

teacher\_come\_to\_university = 0.3

infectiousness = 0.799 *#https://rg.ru/2020/05/04/*

*<→issledovanie-veroiatnost-zarazitsia-covid-vyshe-vsego-doma-i-v-transporte.*

*<→html*

incubation\_period\_expected\_value = 11 *#https://iz.ru/989894/2020-03-22/*

*<→nazvan-srednii-inkubatcionnyi-period-koronavirusa*

incubation\_period\_dispersion = 5

mortality = 0.06

illness\_time\_expected\_value = 20 *#https://iz.ru/981482/2020-02-28/*

*<→voz-nazvala-sroki-vyzdorovleniia-ot-koronavirusa*

illness\_time\_dispersion = 5

immunitet\_period\_expected\_value = 240

immunitet\_period\_dispersion = 100 drawer3D = Drawer3D(" ")

**for** illness\_time\_expected\_value **in** np.arange(3, 30, 2): population = Population(students\_num,␣

*<→*student\_meeting\_person\_expected\_value,

student\_meeting\_person\_dispersion,␣

*<→*student\_come\_to\_university,

teachers\_num, teacher\_meeting\_person\_expected\_value, teacher\_meeting\_person\_dispersion,␣

*<→*teacher\_come\_to\_university,

infectiousness, incubation\_period\_expected\_value, incubation\_period\_dispersion,

mortality,

illness\_time\_expected\_value, illness\_time\_dispersion, immunitet\_period\_expected\_value,␣

*<→*immunitet\_period\_dispersion)

population.first\_infection() autum\_semestr\_loop( population, drawer3D) drawer3D.setK(illness\_time\_expected\_value)

**return** drawer3D

[25]:

**import multiprocessing**

num\_of\_cpu = multiprocessing.cpu\_count() print (num\_of\_cpu)

32

[26]:

experiment\_list = [experiment1,

experiment2, experiment3, experiment4, experiment5, experiment6, experiment7, experiment8, experiment9, experiment10, experiment11, experiment12]

[27]:

**def** experiment\_do(i):

**return** experiment\_list[i]()

[28]:

%%time

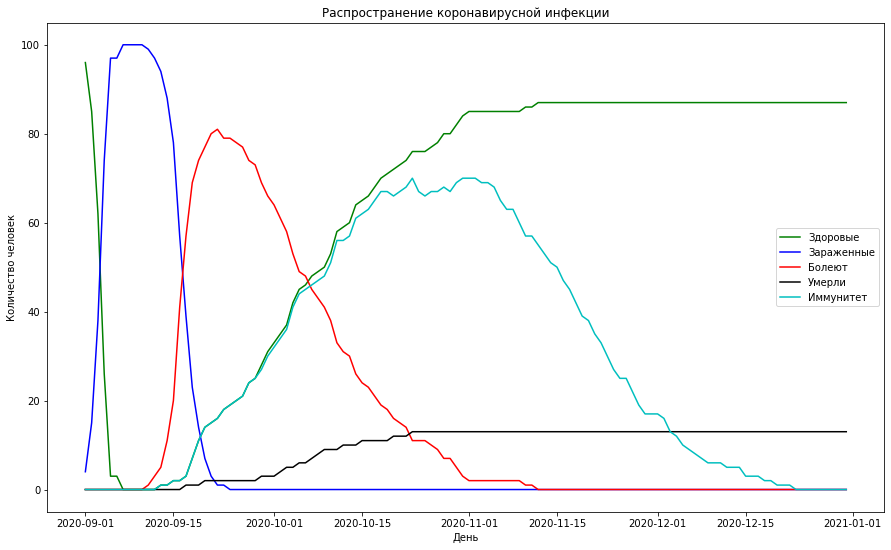
pool = multiprocessing.Pool(processes=num\_of\_cpu)

res = pool.map(experiment\_do, range(len(experiment\_list)))

CPU times: user 1.09 s, sys: 102 ms, total: 1.19 s Wall time: 3min 11s

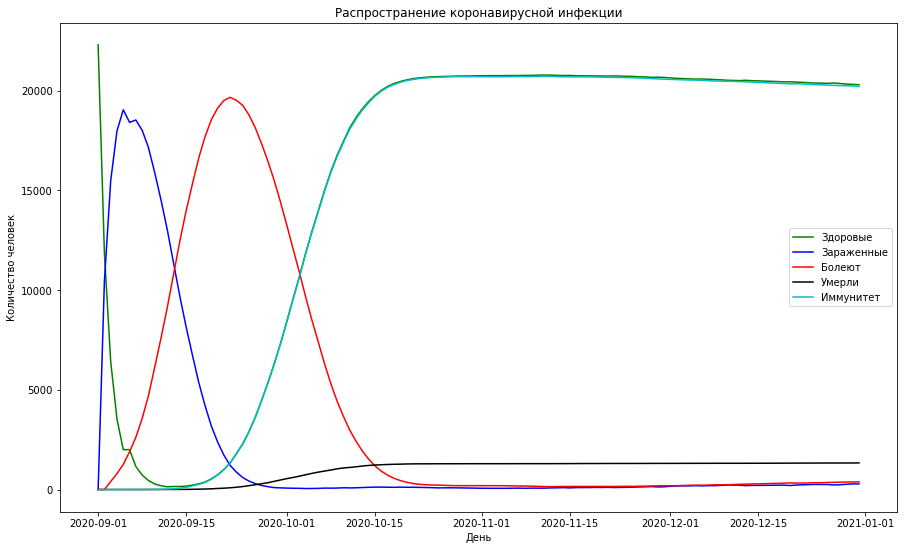
[29]:

res[0].draw()



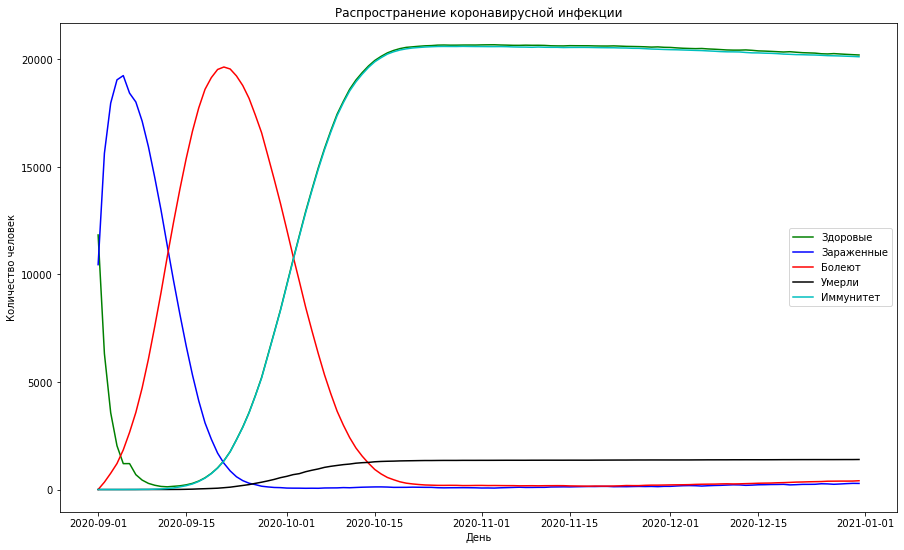
[30]:

res[1].draw()



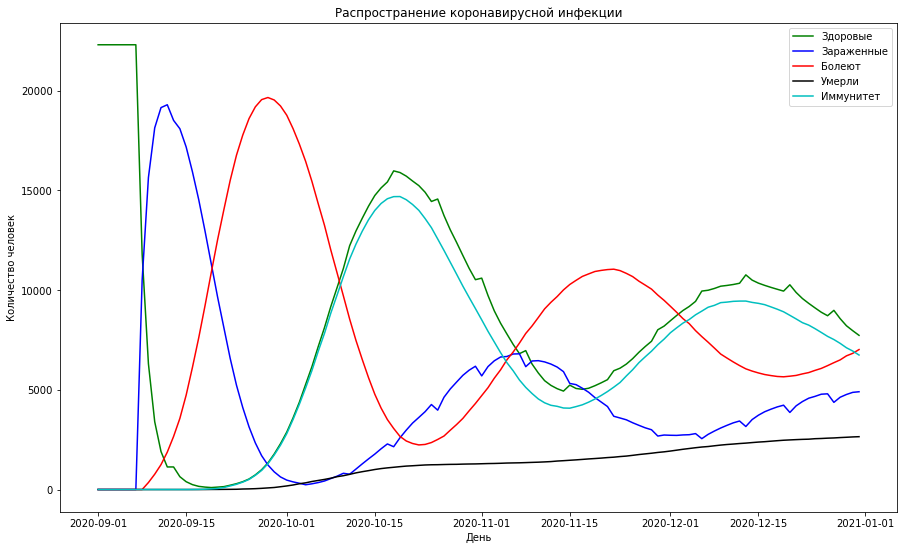
[31]:

res[2].draw()



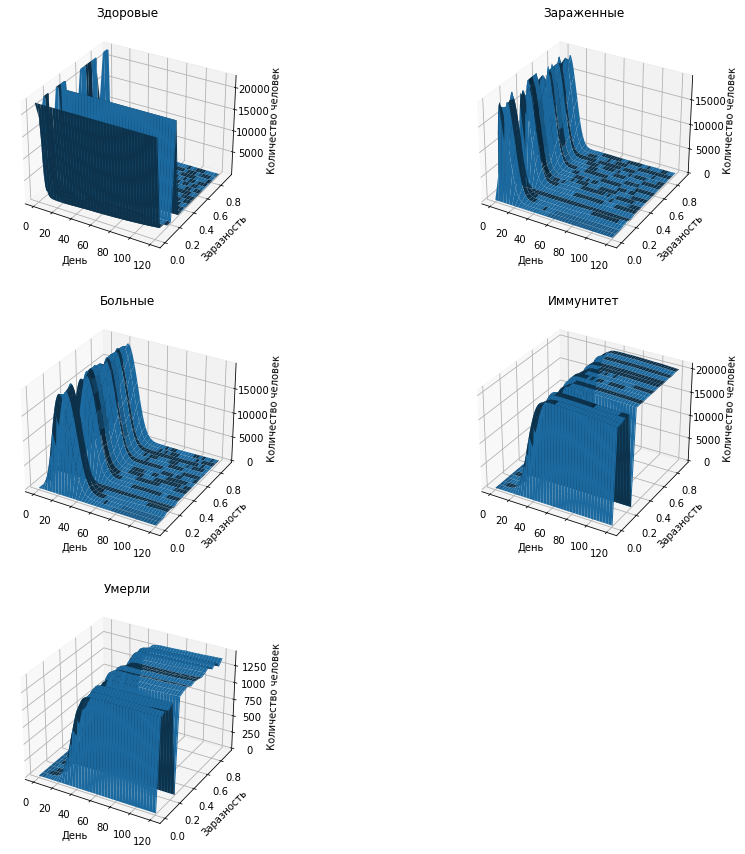
[32]:

res[3].draw()



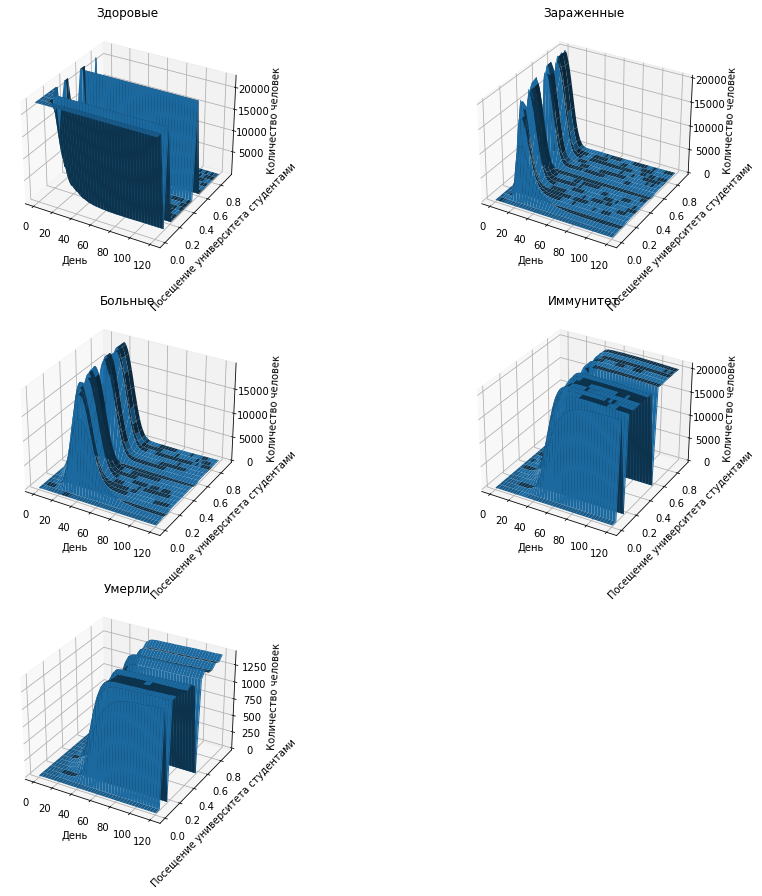
[33]:

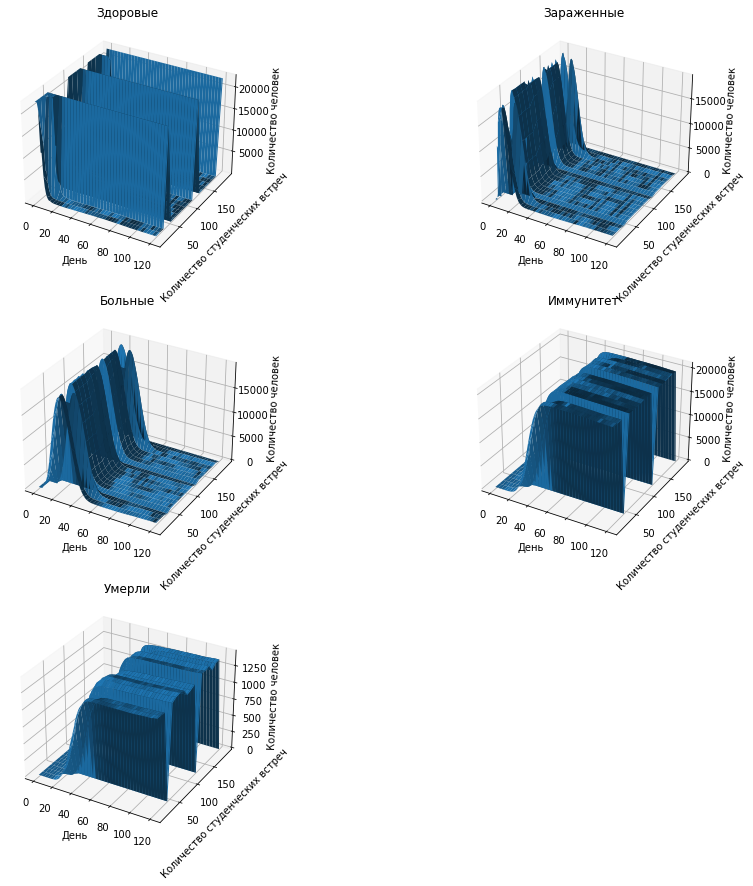
res[4].draw()

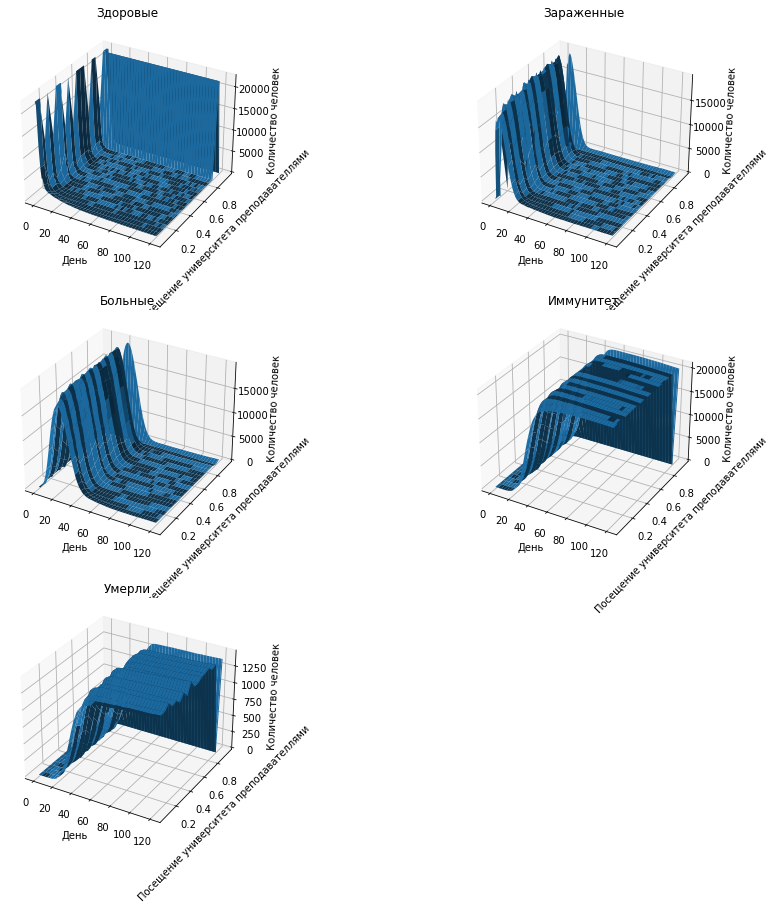


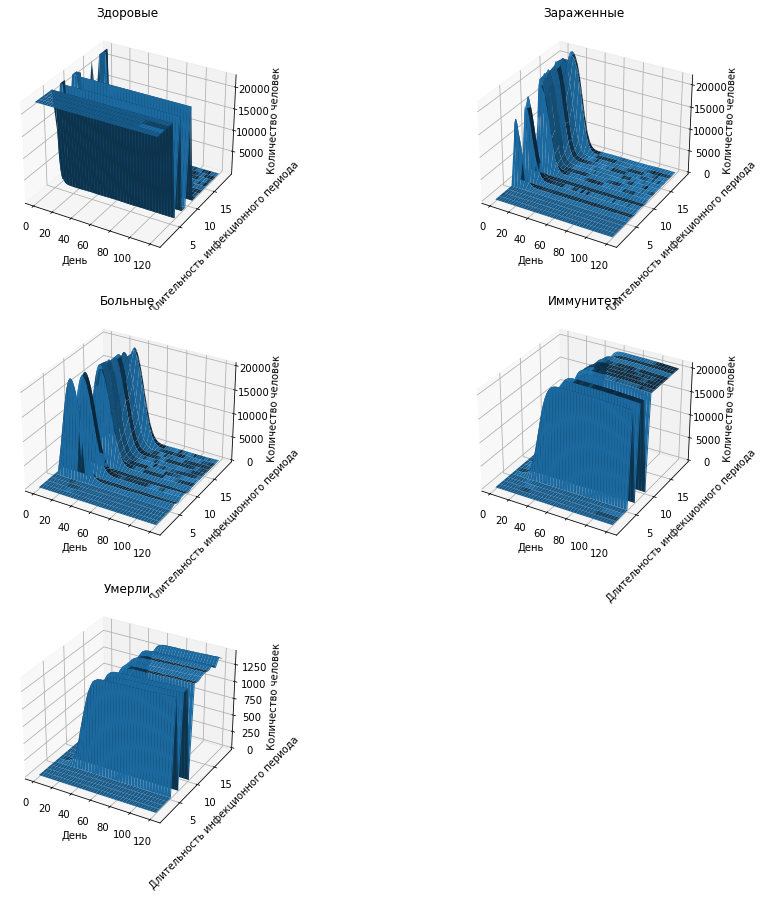
[34]:

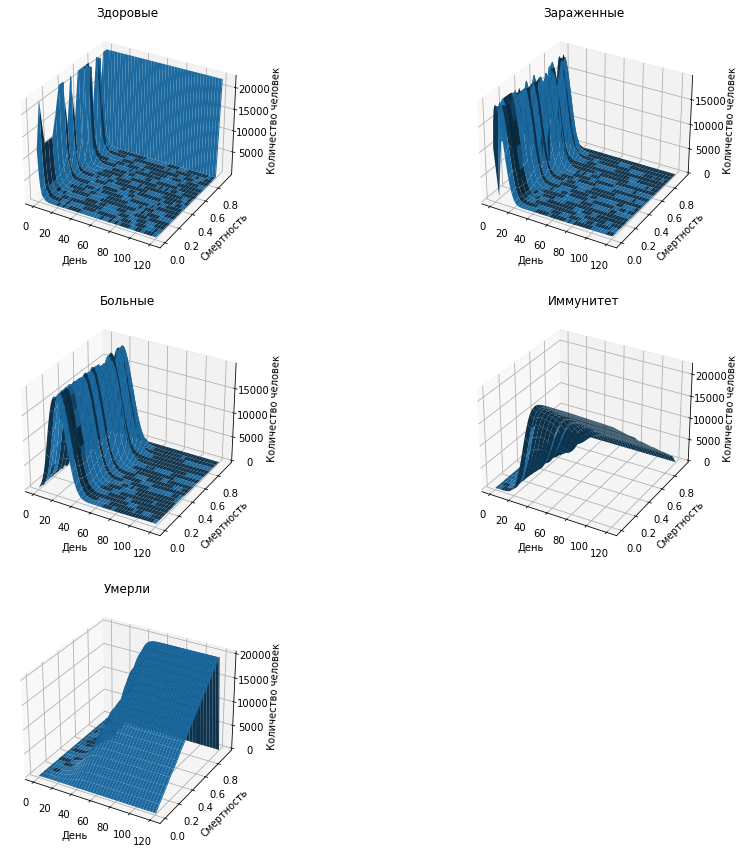
res[5].draw()





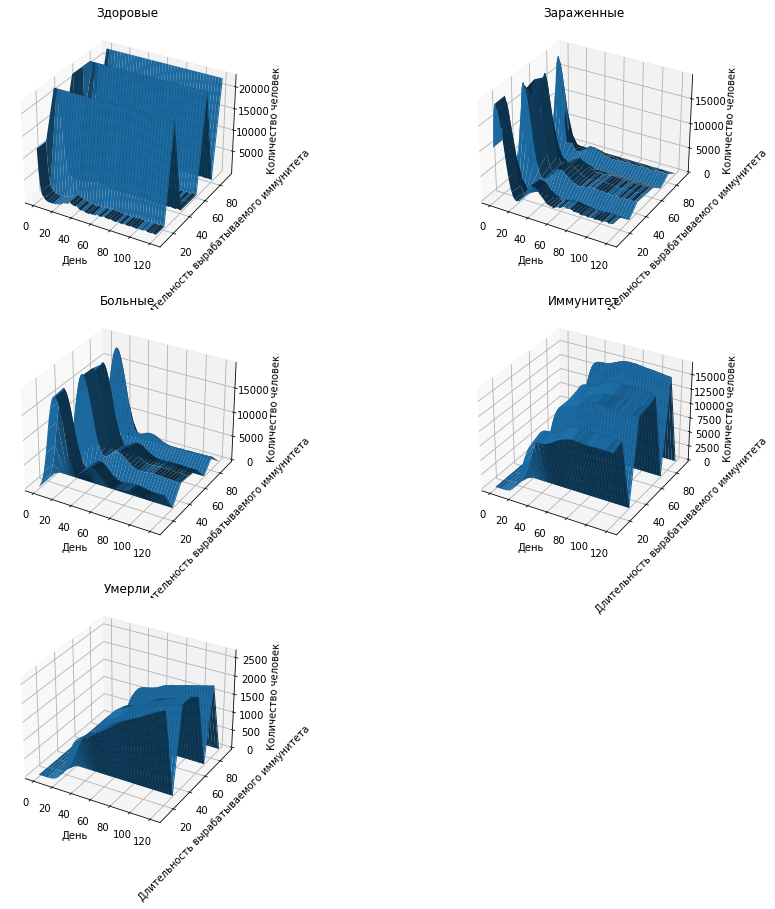






[39]:

res[10].draw()



[40]:

res[11].draw()

