

```
In [1]: import numpy as np
import skfuzzy as fuzz
import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

%matplotlib inline
```

```
In [2]: #Зададим 5 входных и 1 выходную лингвистические переменные
price = ctrl.Antecedent(np.arange(0, 101, 1), "Цена за акцию")
company_index = ctrl.Antecedent(np.arange(0, 11, 1), "Индекс надежности компании")
country_index = ctrl.Antecedent(np.arange(0, 11, 1), "Индекс надежности страны")
date_of_dividends = ctrl.Antecedent(np.arange(0, 11, 1), "Близость даты выплаты дивидендов")

purchase_decision = ctrl.Consequent(np.arange(0, 101, 1), "Решение о покупке")
```

```
In [3]: #Задаем входную переменную price через трапецевидную функцию
price["низкая"] = fuzz.trapmf(price.universe, [0, 0, 15, 35])
price["нормальная"] = fuzz.trapmf(price.universe, [25, 30, 56, 65])
price["высокая"] = fuzz.trapmf(price.universe, [55, 87, 100, 100])

#Задаем входные переменные company_index и country_index через функцию Гаусса
company_index["не надежная"] = fuzz.gaussmf(company_index.universe, 0, 1.2)
company_index["средняя"] = fuzz.trapmf(company_index.universe, [2, 3.5, 5.5, 8])
company_index["надежная"] = fuzz.trapmf(company_index.universe, [7.5, 8, 10, 10])

country_index["не надежная"] = fuzz.gaussmf(country_index.universe, 0, 2)
country_index["средняя"] = fuzz.trapmf(country_index.universe, [3, 4, 6, 7])
country_index["надежная"] = fuzz.trapmf(country_index.universe, [7, 8, 10, 10])

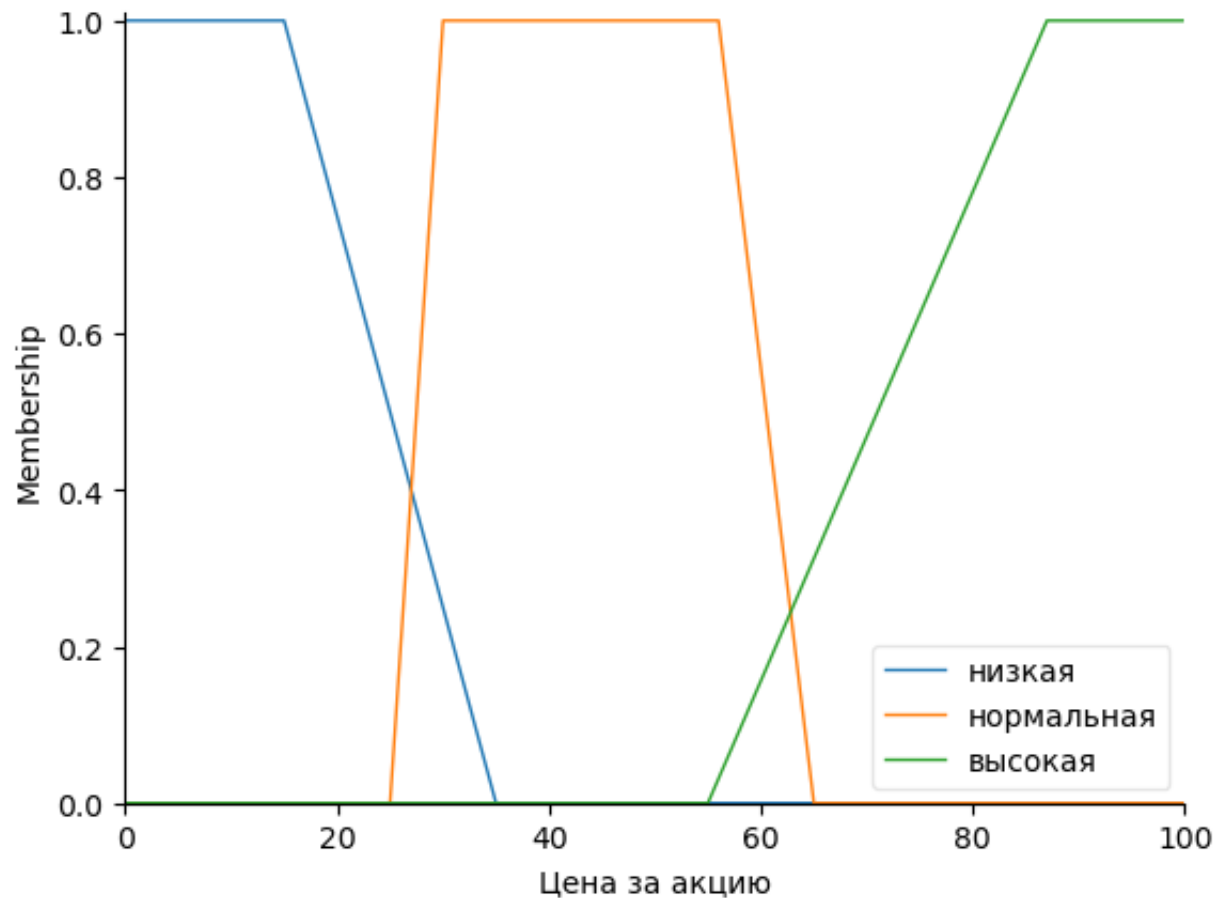
date_of_dividends["скоро"] = fuzz.trapmf(date_of_dividends.universe, [0, 0, 0.8, 1.2])
date_of_dividends["не скоро"] = fuzz.trapmf(date_of_dividends.universe, [3.5, 8, 10, 12])
```

```
In [4]: #Задаем выходную переменную
purchase_decision["не покупать"] = fuzz.trapmf(purchase_decision.universe, [0, 0, 15, 45])
purchase_decision["покупать"] = fuzz.trapmf(purchase_decision.universe, [50, 65, 100, 100])
```

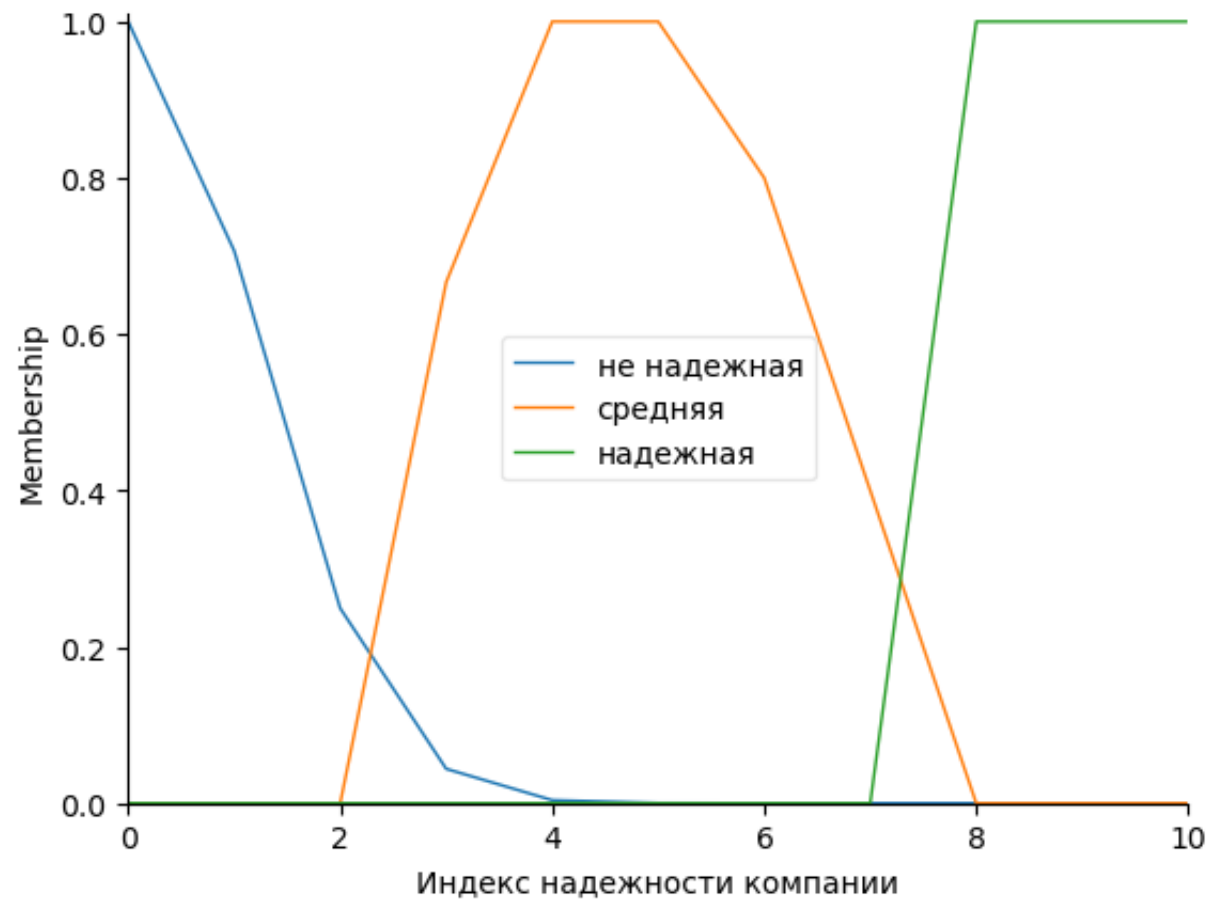
```
In [5]: price.view()
```

/Users/user/anaconda3/lib/python3.11/site-packages/skfuzzy/control/fuzzyvariable.py:122: UserWarning: Matplotlib is currently using module://matplotlib_inline.backend_inline, which is a non-GUI backend, so cannot show the figure.

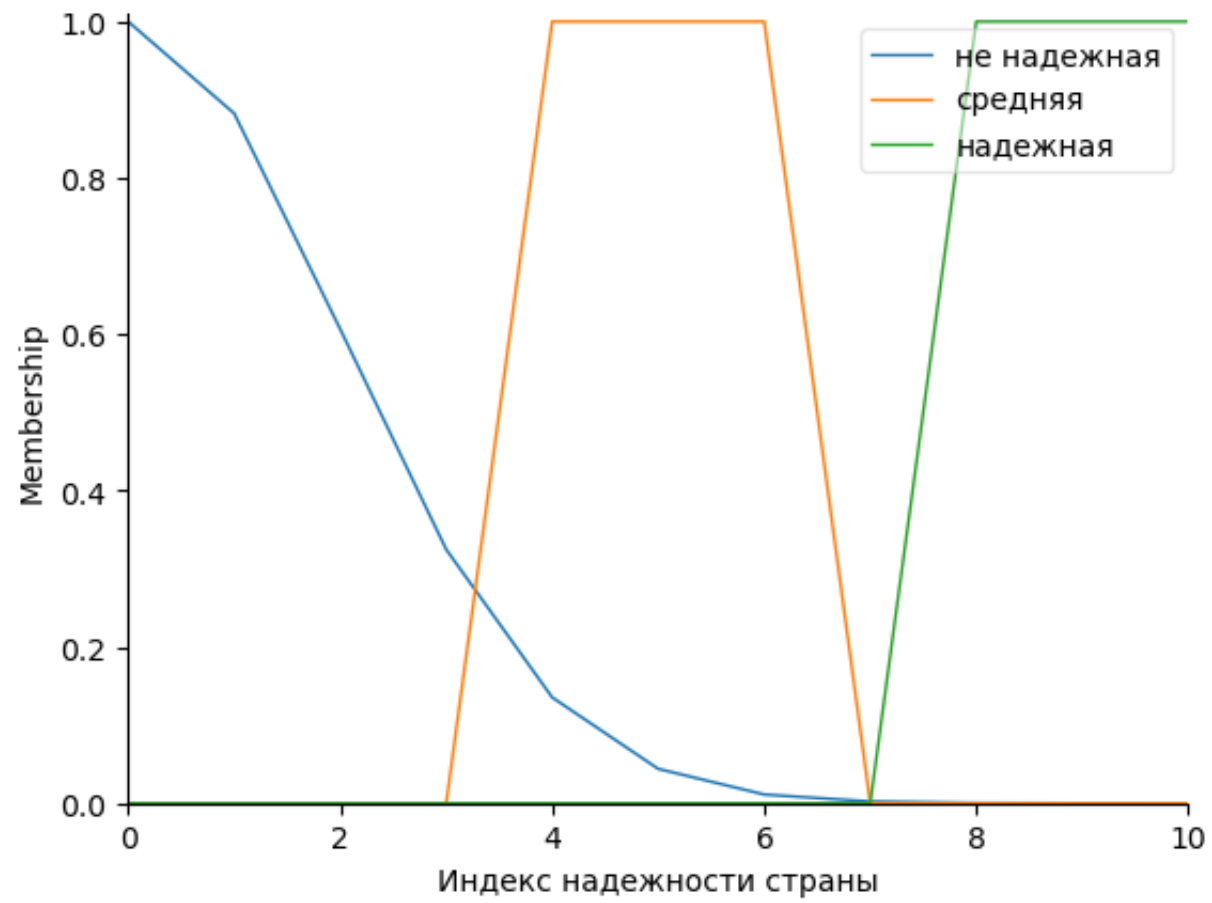
```
fig.show()
```



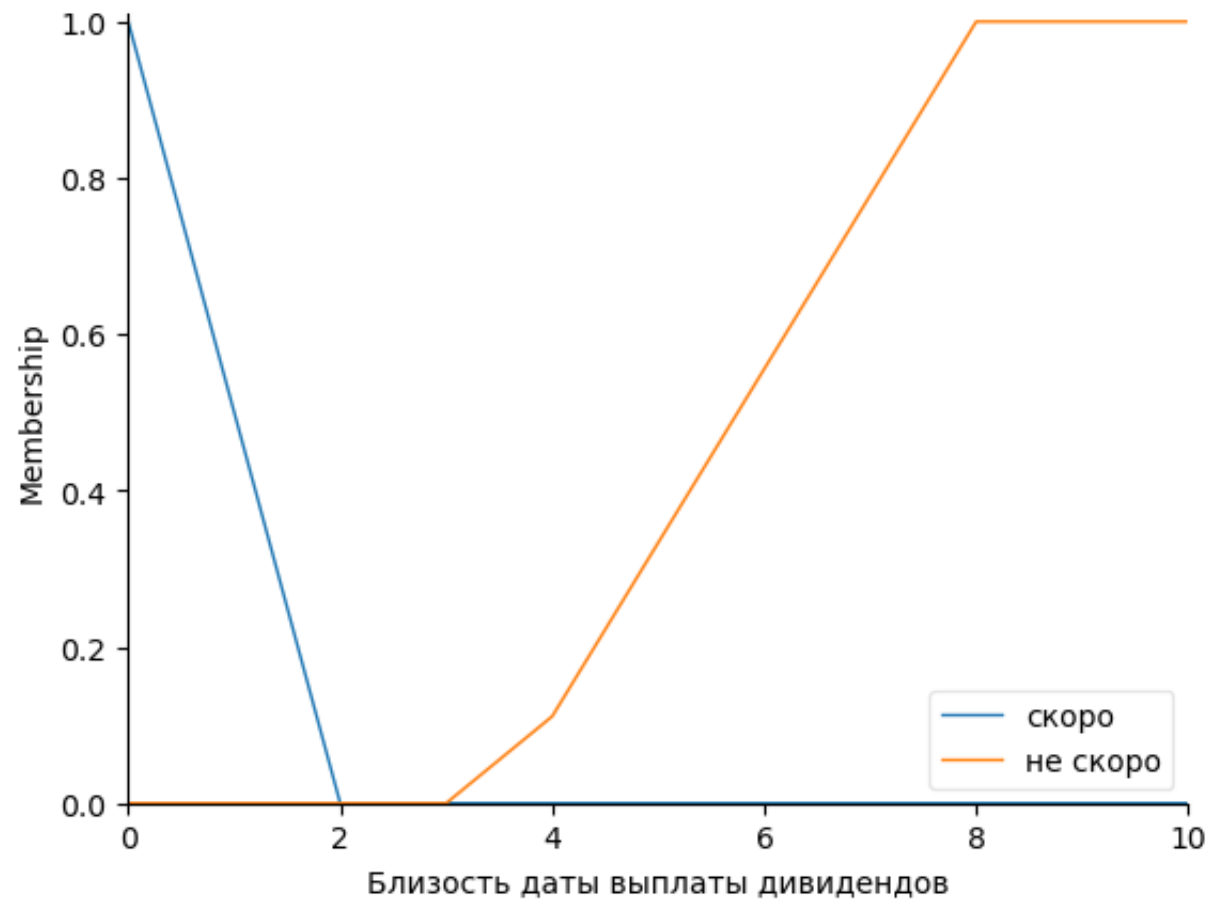
```
In [6]: company_index.view()
```



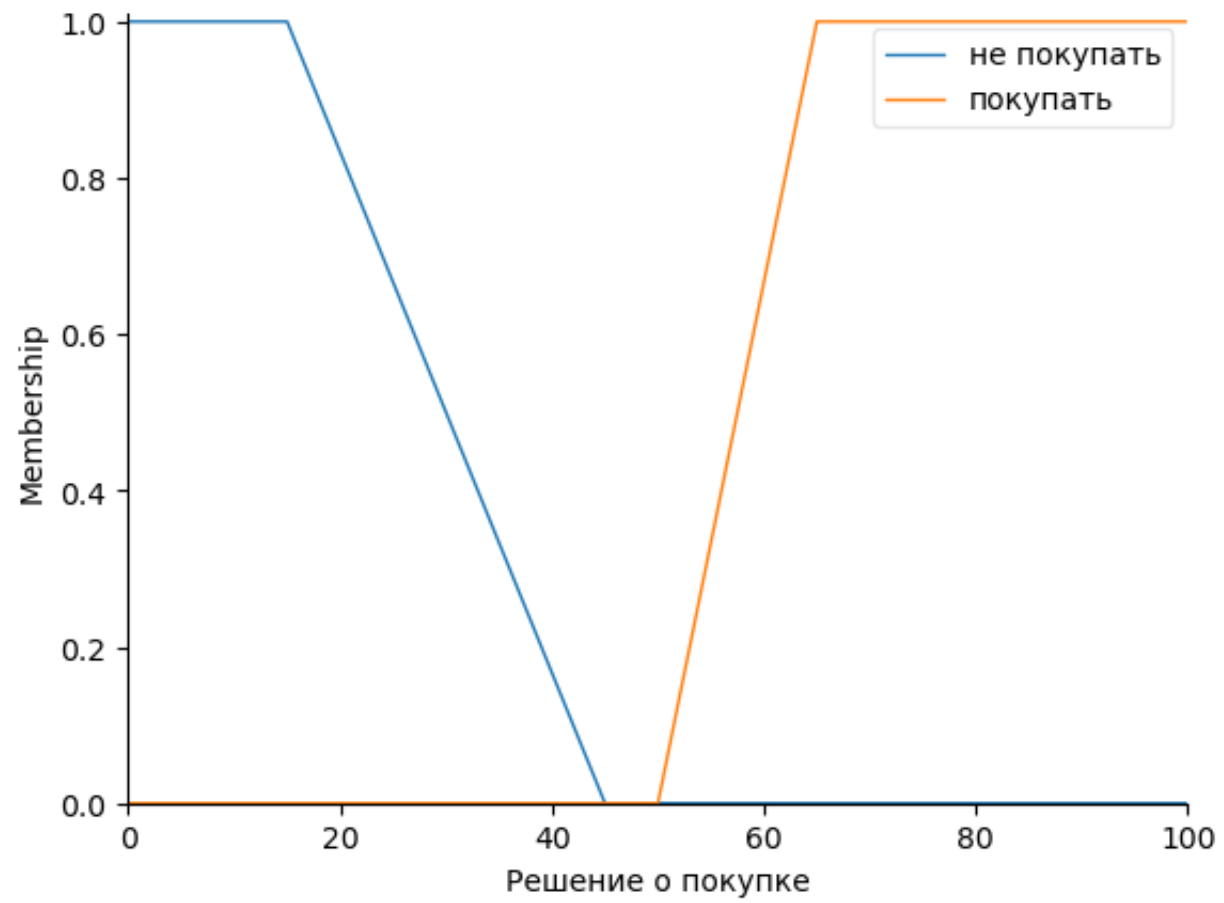
```
In [7]: country_index.view()
```



```
In [8]: date_of_dividends.view()
```



```
In [9]: purchase_decision.view()
```



```
In [10]: #Определение правил
rule1 = ctrl.Rule(price["высокая"] | company_index["не надежная"],
                  purchase_decision["не покупать"])

rule2 = ctrl.Rule(company_index["не надежная"] & country_index["не надежная"],
                  purchase_decision["не покупать"])

rule3 = ctrl.Rule(date_of_dividends["скоро"] & price["высокая"] ,
                  purchase_decision["не покупать"])

rule4 = ctrl.Rule(date_of_dividends["скоро"] & price["низкая"],
                  purchase_decision["покупать"])

rule5 = ctrl.Rule(country_index["надежная"] | company_index["надежная"],
                  purchase_decision["покупать"])

rule6 = ctrl.Rule(country_index["средняя"] & company_index["средняя"] & price["низкая"],
                  purchase_decision["покупать"])

rule7 = ctrl.Rule(price["нормальная"] | country_index["надежная"] | company_index["надежная"],
                  purchase_decision["покупать"])

rule8 = ctrl.Rule(date_of_dividends["скоро"] & country_index["не надежная"] & price["низкая"],
                  purchase_decision["покупать"])

rule9 = ctrl.Rule(date_of_dividends["скоро"] & price["нормальная"],
                  purchase_decision["покупать"])

rule10 = ctrl.Rule(price["высокая"] & company_index["надежная"] & country_index["надежная"],
                   purchase_decision["покупать"])
```

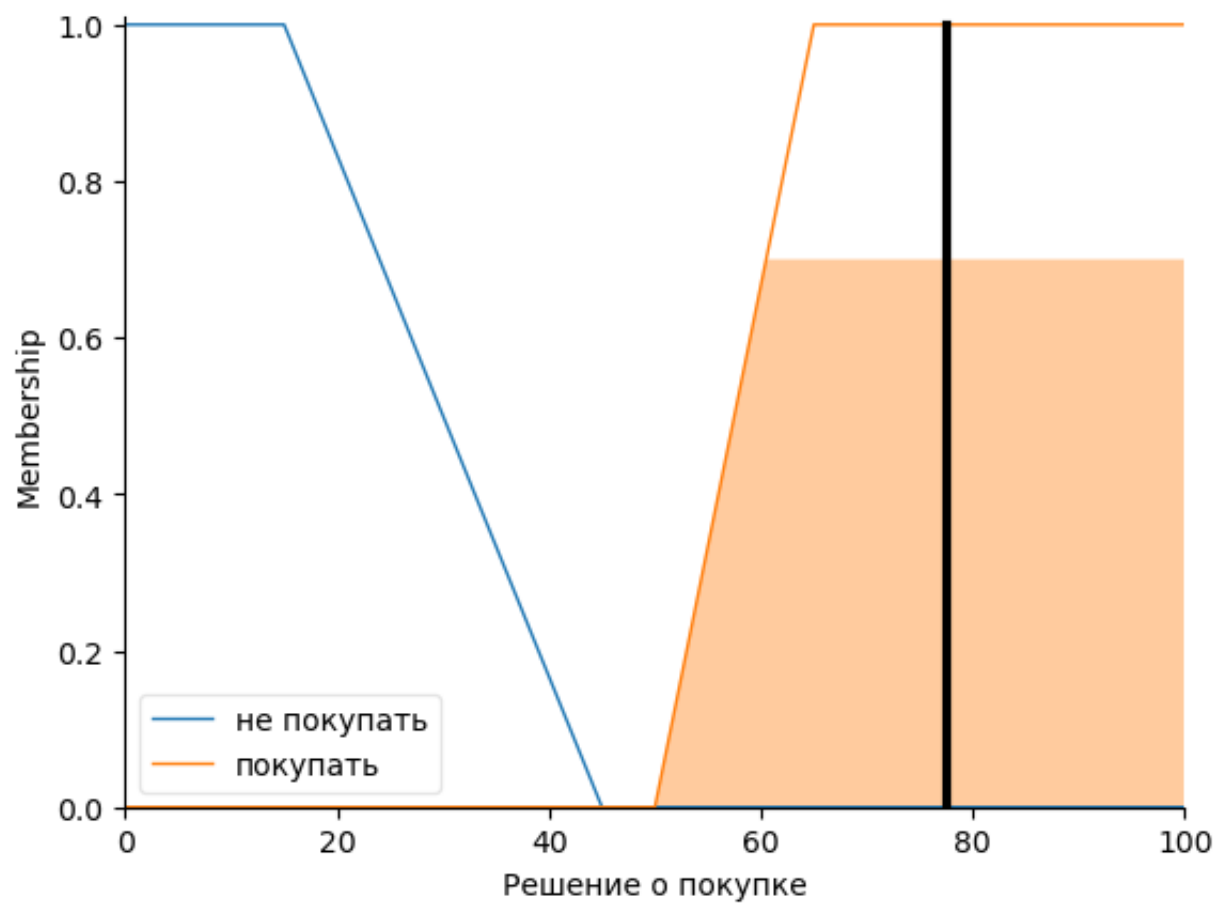
[illegible]

```
In [12]: #Создаем модель расчета  
decision = ctrl.ControlSystemSimulation(decision_ctrl)  
  
#Подаем на вход четкие числа  
decision.input["Цена за акцию"] = 23  
decision.input["Индекс надежности компании"] = 7.7  
decision.input["Индекс надежности страны"] = 2.1  
decision.input["Близость даты выплаты дивидендов"] = 8  
  
#Запускаем расчет  
decision.compute()
```

```
In [13]: #печатаем результат  
print(decision.output["Решение о покупке"])  
  
77.52234538820413
```



```
In [14]: #выводим результат в виде графика  
purchase_decision.view(sim=purchase_decision)
```



```
In [ ]:
```