Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Системы искусственного интеллекта**

**Лабораторная работа №2**

**«Генетические алгоритмы»**

**Вариант №4**

Выполнил

студент группы ИВТАПбд-41

Евкарпиева Полина Александровна

Проверил

Преподаватель кафедры ВТ

Хайруллин И.Д.

Ульяновск

2025

**Цель работы:**

Необходимо на языке Python разработать скрипт, позволяющий выполнить операцию пересечения заданных пользователем нечетких множеств с трапециевидными функциями принадлежности. Входными данными будут параметры функций принадлежности и четкие объекты для каждого из множеств. Выходными – пересечение данных нечетких множеств.

Предметная область: финансовый риск

* Кредитный рейтинг: низкий, средний, высокий, премиальный
* Уровень риска: безопасный, приемлемый, рискованный, опасный

**Контрольные вопросы по лабораторной работе:**

1. Дайте определение нечеткому множеству.
2. Какие способы задания функций принадлежности вы знаете?
3. Какую операцию вы реализовали в своей лабораторной работе?

**Теоретический материал:**

Объекты – лингвистические переменные. Таким образом, операндами и результатом интеллектуальных операций являются значения особого вида – *нечеткие множества.*

*Нечёткие множества (fuzzy sets) в машинном обучении (ML) — это наборы элементов с разной степенью принадлежности.* В отличие от традиционных множеств с чёткими границами, нечёткие множества позволяют элементам принадлежать к группе в разной степени. Каждый элемент имеет значение принадлежности от 0 до 1, указывающее на его степень принадлежности к набору.

Основные интеллектуальные операции строятся с помощью *операций нечеткой логики.*

Алгоритмы вычисления нечетких значений предназначены для манипулирования со значениями, представленными нечеткими множествами на основе операций нечеткой логики, поэтому они классифицируются как нечеткие системы логического вывода. Часто используют сокращенную форму обозначенного класса моделей – ***нечеткие модели или нечеткие системы.***

Некоторые особенности теории:

* **Функции принадлежности** — определяют, каким образом каждому элементу нечёткого множества присваивается значение принадлежности. Распространённые типы — треугольные, трапециевидные и гауссовы функции.
* **Лингвистические переменные** — переменные, описываемые естественным языком, такие как «высокий», «средний» или «низкий».

**Способы задания функции принадлежности:**

* Структурный способ
* Функциональный способ

**Пересечение нечетких множеств** A и B определяется как операция, результатом которой является новое нечеткое множество C = A ∩ B с функцией принадлежности:



Эта операция соответствует логической операции "И" и находит минимальную степень принадлежности для каждого элемента.

**Описание реализации:**

Архитектура программы включает несколько ключевых компонентов: модуль математических вычислений для работы с нечеткими множествами, модуль взаимодействия с пользователем и модуль визуализации результатов.

Программа начинается с инициализации необходимых структур данных и параметров, затем переходит к взаимодействию с пользователем, после чего выполняет вычисления и завершается визуализацией результатов.

Для работы с трапециевидными функциями принадлежности была разработана специализированная функция trapezoidal\_mf(), которая принимает четыре параметра, определяющих форму трапеции. Особое внимание было уделено обработке граничных условий и предотвращению ошибок деления на ноль. В коде добавлена малая константа 1e-6 к знаменателям, что обеспечивает численную устойчивость при вычислениях.

Листинг 1. Трапециевидная функция принадлежности

|  |
| --- |
| def trapezoidal\_mf(x, a, b, c, d):  left\_slope = np.clip((x - a) / (b - a + 1e-6), 0, 1)  right\_slope = np.clip((d - x) / (d - c + 1e-6), 0, 1)  return np.maximum(0, np.minimum(left\_slope, right\_slope)) |

Были определены два универсума: для кредитного рейтинга (0-1000) и уровня риска (0-100%). Для каждого универсума создано по четыре нечетких множества с перекрывающимися зонами, что позволяет обеспечить плавные переходы между лингвистическими категориями.

В области финансового риска параметры трапециевидных функций были подобраны на основе экспертных представлений о взаимосвязи между кредитным рейтингом и уровнем риска. Для кредитного рейтинга установлены следующие диапазоны:

* Низкий рейтинг охватывает диапазон 0-400 баллов с ядром 0-300
* Средний рейтинг занимает 350-650 баллов с ядром 450-550
* Высокий рейтинг определен в интервале 600-900 баллов
* Премиальный рейтинг начинается с 850 баллов

Аналогичным образом настроены параметры для уровня риска, где каждая категория имеет свою зону влияния с перекрытиями для обеспечения непрерывности шкалы оценок.

Операция пересечения реализована через функцию fuzzy\_intersection(), которая использует поэлементный минимум между двумя массивами значений принадлежности. Для работы с конкретными значениями, вводимыми пользователем, применяется встроенная функция min() для пар значений принадлежности.

Листинг 2. Пересечение множеств

|  |
| --- |
| def fuzzy\_intersection(set1, set2):  return np.minimum(set1, set2) |

Особенностью реализации является то, что пересечение вычисляется для всех возможных комбинаций категорий кредитного рейтинга и уровня риска, что дает полную картину взаимосвязи между этими факторами.

Интерфейс вывода результатов организован в виде структурированных таблиц, где последовательно отображаются степени принадлежности для каждого нечеткого множества и результаты операции пересечения для всех комбинаций.

Модуль визуализации создает комплексное представление результатов через систему из четырех графиков. На первых двух графиках отображаются трапециевидные функции принадлежности для кредитного рейтинга и уровня риска с отметками введенных пользователем значений. Это позволяет наглядно увидеть, как конкретные числовые значения соотносятся с лингвистическими категориями.

На третьем и четвертом графиках представлены столбчатые диаграммы, которые количественно отображают степени принадлежности для всех категорий. Каждый столбец подписан числовым значением степени принадлежности, что облегчает точную интерпретацию результатов.

Вычислительный модуль программы использует интерполяцию для определения степеней принадлежности конкретных значений. Для каждого введенного пользователем значения программа находит соответствующие степени принадлежности ко всем определенным нечетким множествам.

Операция пересечения выполняется как для визуализации на универсумах, так и для конкретных пар значений. Это позволяет одновременно получить как общее представление о взаимосвязи множеств, так и конкретные числовые результаты для анализируемого случая.

**Тестирование:**

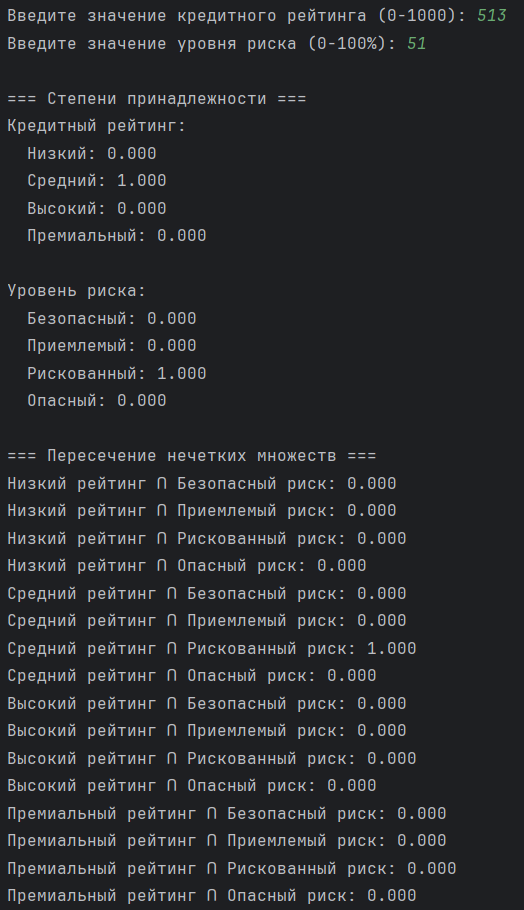


Рис. 1. Результат 1 теста в консоли

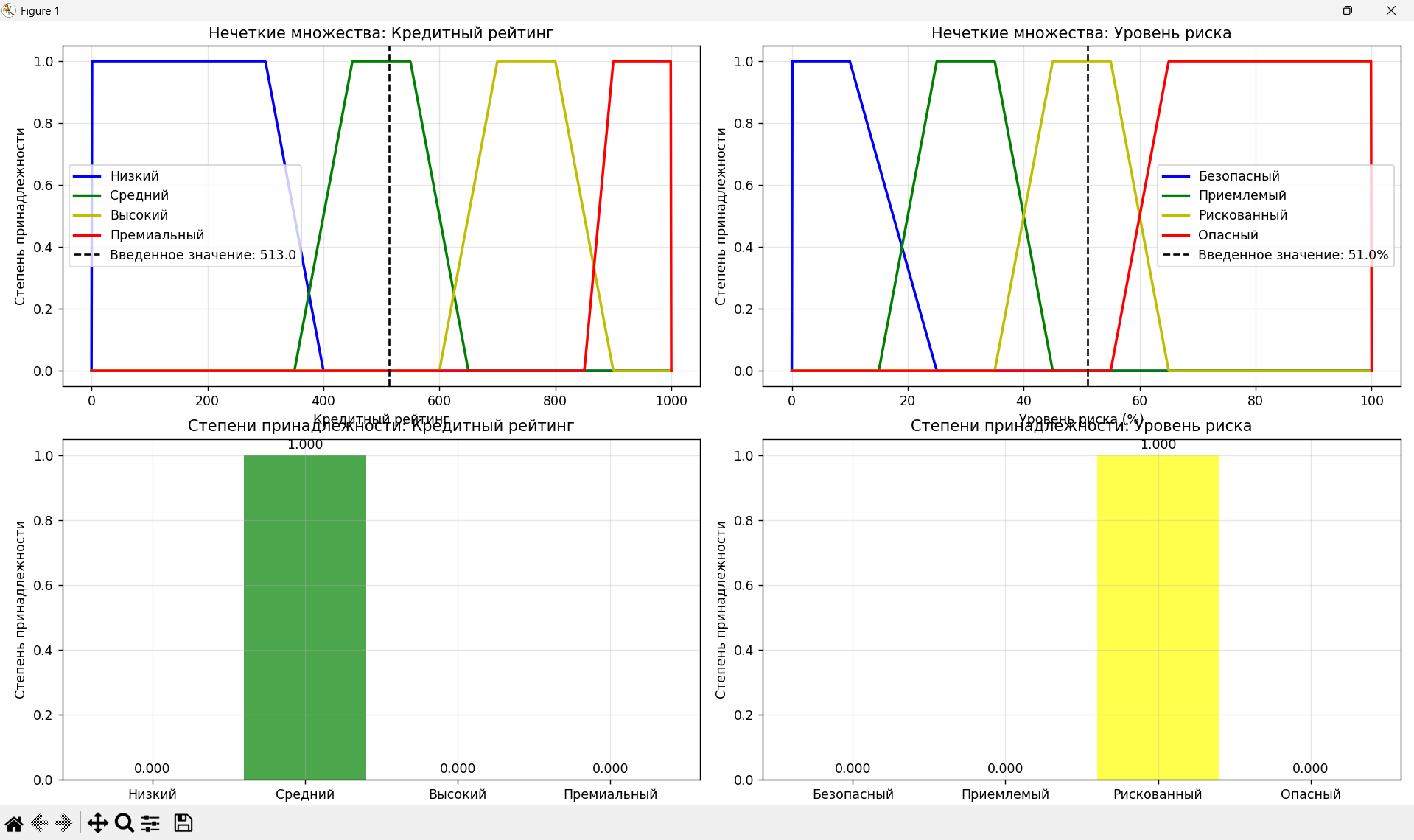


Рис. 2. Графики 1 теста

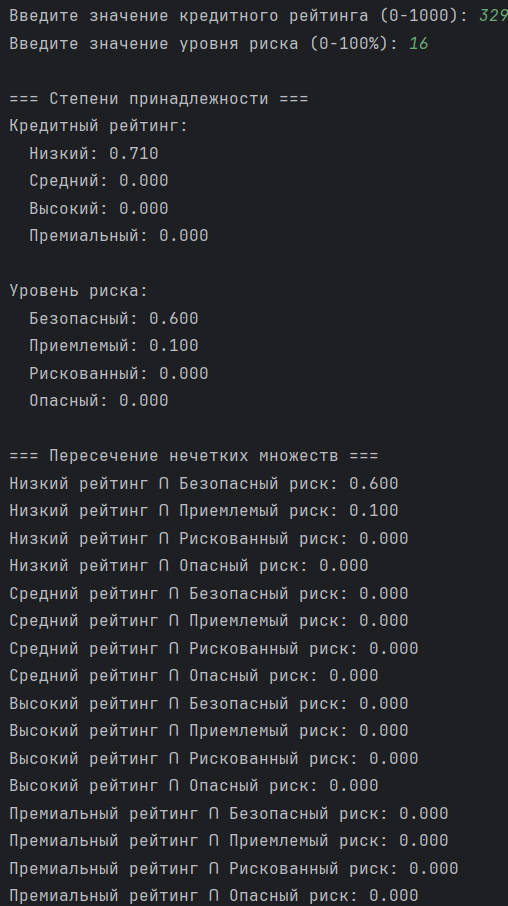


Рис. 3. Результат 2 теста в консоли

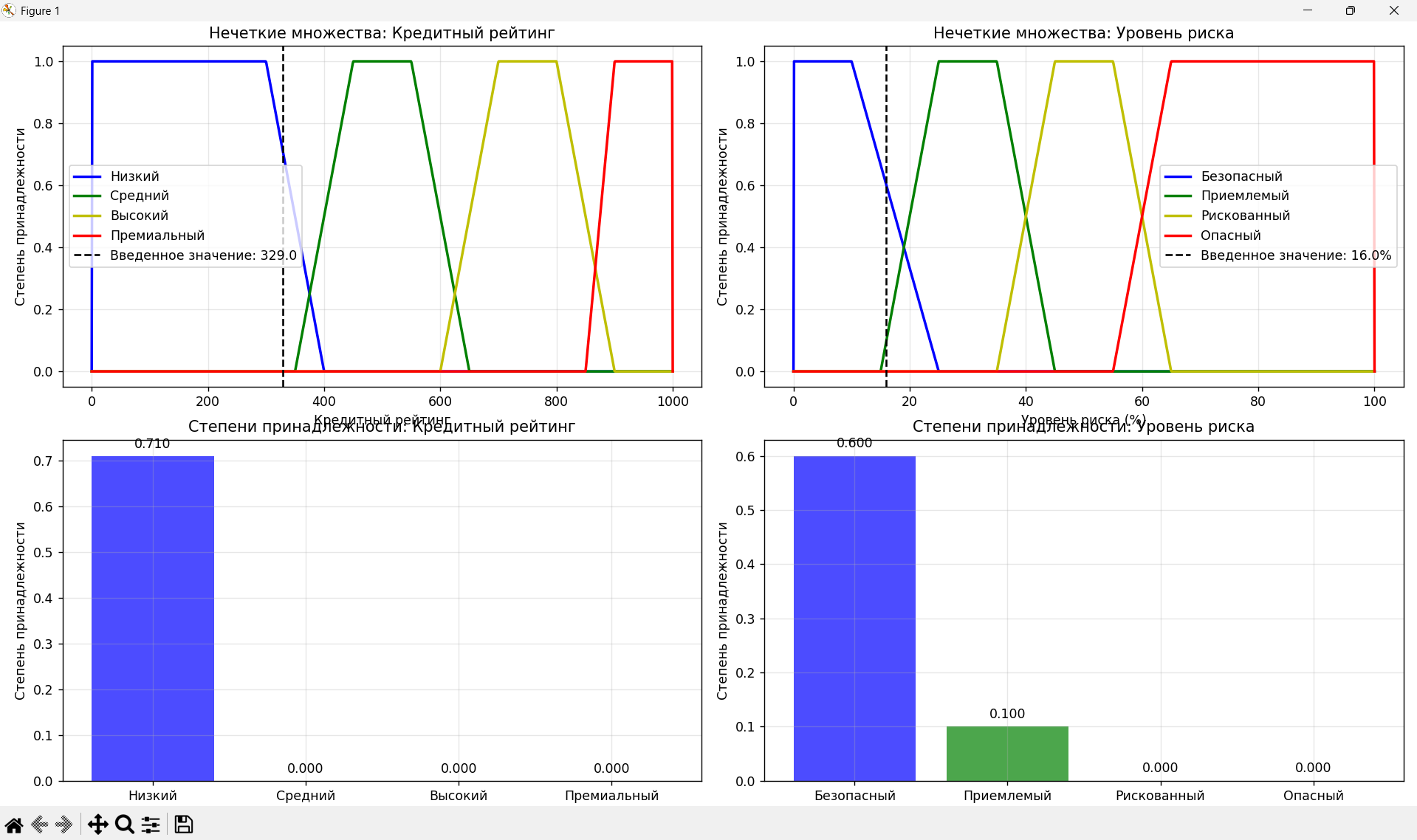


Рис. 4. Графики 2 теста

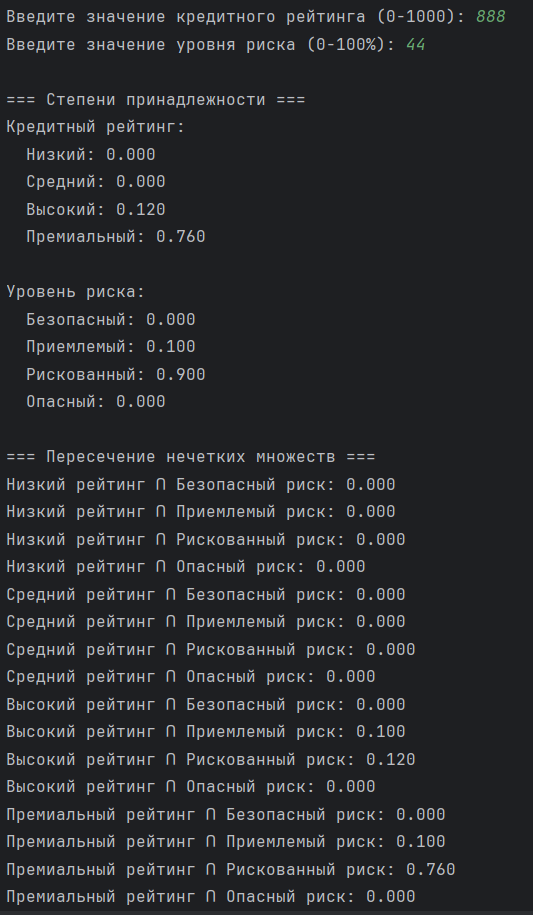


Рис. 5. Результат 3 теста в консоли

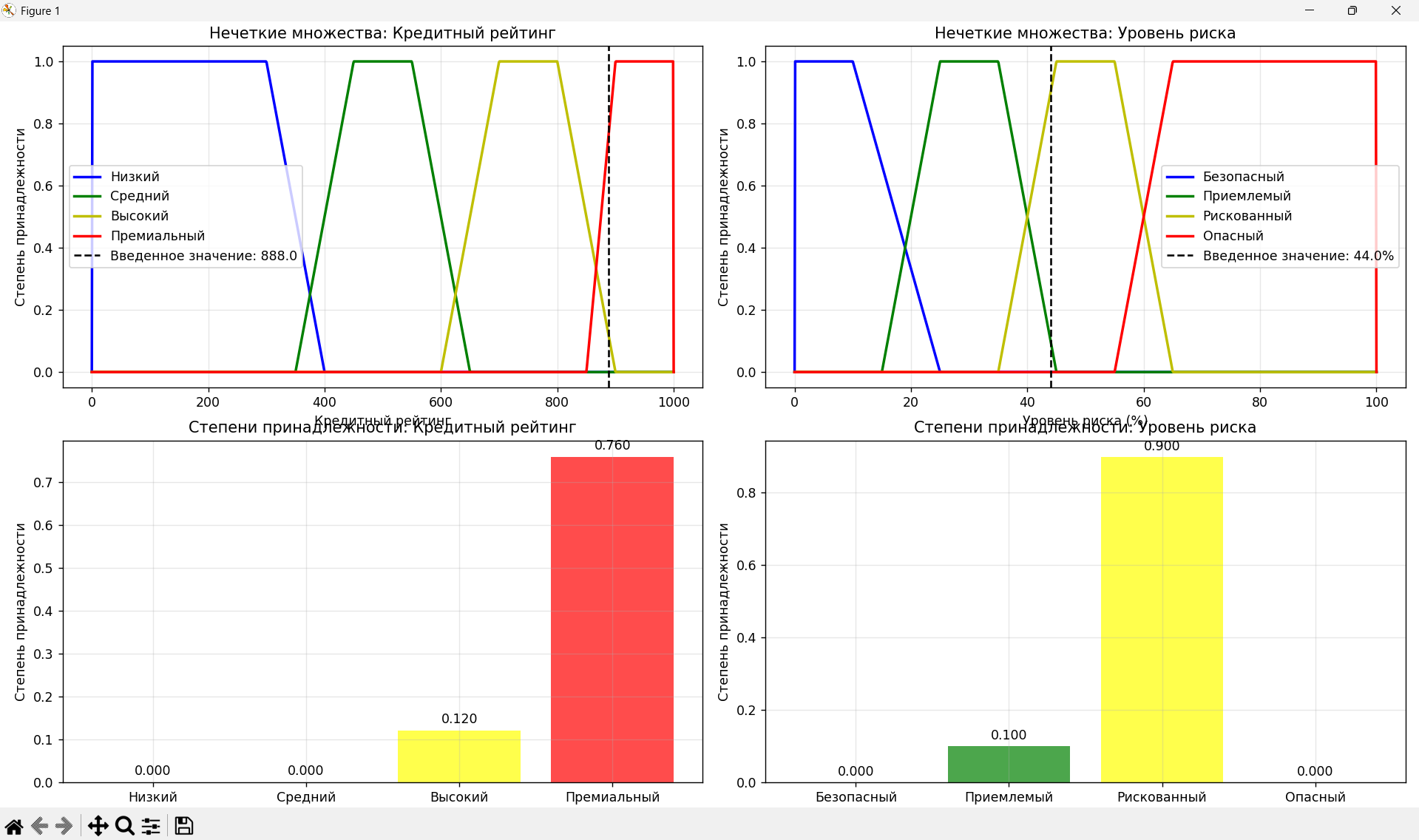


Рис. 6. Графики 3 теста

**Контрольные вопросы:**

*1. Дайте определение нечеткому множеству*

**Нечеткое множество** — это расширение понятия классического множества, в котором элементы могут иметь **степень принадлежности** к множеству, принимающую значения из интервала [0,1], а не только бинарные значения 0 или 1.

Формально, нечеткое множество A на универсуме X определяется как множество упорядоченных пар:

A = {(x, μₐ(x)) | x ∈ X}

где:

* X - универсум (область определения)
* μₐ: X → [0,1] - функция принадлежности
* μₐ(x) - степень принадлежности элемента x к нечеткому множеству A

**Отличия от классического множества:**

* В классической теории: элемент либо принадлежит (1), либо не принадлежит (0) множеству
* В нечеткой логике: элемент может принадлежать множеству частично (0.3, 0.7 и т.д.)

*2. Какие способы задания функций принадлежности вы знаете?*

Существует несколько основных способов задания функций принадлежности:

* 1. Параметрические функции
     + Треугольная: задается тремя параметрами (a, b, c)
     + Трапециевидная: задается четырьмя параметрами (a, b, c, d)
     + Гауссова
     + Сигмоидальная
  2. Непараметрические методы
     + Статистические: на основе гистограмм и распределений
     + Экспертные оценки: опрос специалистов в предметной области
     + Адаптивные алгоритмы: нейронные сети, генетические алгоритмы
  3. Эвристические методы
     + Относительные предпочтения: попарные сравнения элементов
     + Интервальные оценки: задание функций через α-срезы
  4. Аналитические методы
     + На основе регрессионного анализа
     + Метод главных компонент

*3. Какую операцию вы реализовали в своей лабораторной работе?*

В лабораторной работе была реализована **операция пересечения нечетких множеств**.

Для двух нечетких множеств A и B с функциями принадлежности μₐ(x) и μ\_B(x) пересечение C = A ∩ B определяется как:

μ\_C(x) = min(μₐ(x), μ\_B(x)) для всех x ∈ X

Было выполнено пересечение множеств:

* **Кредитный рейтинг** = {Низкий, Средний, Высокий, Премиальный}
* **Уровень риска** = {Безопасный, Приемлемый, Рискованный, Опасный}

Результат: 16 комбинаций

Операция пересечения соответствует логической операции **"И"**:

* "Если кредитный рейтинг Средний **И** риск Приемлемый, то..."
* Результирующая степень принадлежности показывает **наихудший** (минимальный) сценарий
* Обеспечивает консервативную оценку финансового риска

Операция пересечения была выбрана как наиболее соответствующая требованиям варианта задания и обеспечивающая надежную оценку финансовых рисков.

**Вывод:**

В ходе выполнения лабораторной работы была успешно разработана и реализована программа для операции пересечения нечетких множеств с трапециевидными функциями принадлежности в предметной области финансового риска. Программа позволяет анализировать взаимосвязь между кредитным рейтингом и уровнем риска, вычисляя степени принадлежности для различных лингвистических категорий и определяя их пересечение через операцию минимума.

**Приложение: код программы**

|  |
| --- |
| import numpy as np import matplotlib.pyplot as plt   # Функция для определения трапециевидной функции принадлежности def trapezoidal\_mf(x, a, b, c, d):  *"""  Трапециевидная функция принадлежности.  :param x: Точки, для которых вычисляется функция принадлежности.  :param a: Левая граница начала возрастания функции.  :param b: Левая верхняя граница (где функция равна 1).  :param c: Правая верхняя граница (где функция равна 1).  :param d: Правая граница окончания убывания функции.  :return: Значение функции принадлежности в точках x.  """* left\_slope = np.clip((x - a) / (b - a + 1e-6), 0, 1)  right\_slope = np.clip((d - x) / (d - c + 1e-6), 0, 1)  return np.maximum(0, np.minimum(left\_slope, right\_slope))   # Операция пересечения нечетких множеств (минимум) def fuzzy\_intersection(set1, set2):  return np.minimum(set1, set2)   # Основная программа def main():  print("=== Пересечение нечетких множеств для финансового риска ===\n")   # Универсум для кредитного рейтинга (0-1000)  x\_credit = np.linspace(0, 1000, 1000)   # Универсум для уровня риска (0-100%)  x\_risk = np.linspace(0, 100, 1000)   # Параметры для кредитного рейтинга  print("Параметры трапециевидных функций для кредитного рейтинга:")  print("Низкий: a=0, b=0, c=300, d=400")  print("Средний: a=350, b=450, c=550, d=650")  print("Высокий: a=600, b=700, c=800, d=900")  print("Премиальный: a=850, b=900, c=1000, d=1000")   # Определение функций принадлежности для кредитного рейтинга  credit\_low = trapezoidal\_mf(x\_credit, 0, 0, 300, 400)  credit\_medium = trapezoidal\_mf(x\_credit, 350, 450, 550, 650)  credit\_high = trapezoidal\_mf(x\_credit, 600, 700, 800, 900)  credit\_premium = trapezoidal\_mf(x\_credit, 850, 900, 1000, 1000)   # Параметры для уровня риска  print("\nПараметры трапециевидных функций для уровня риска:")  print("Безопасный: a=0, b=0, c=10, d=25")  print("Приемлемый: a=15, b=25, c=35, d=45")  print("Рискованный: a=35, b=45, c=55, d=65")  print("Опасный: a=55, b=65, c=100, d=100")   # Определение функций принадлежности для уровня риска  risk\_safe = trapezoidal\_mf(x\_risk, 0, 0, 10, 25)  risk\_acceptable = trapezoidal\_mf(x\_risk, 15, 25, 35, 45)  risk\_risky = trapezoidal\_mf(x\_risk, 35, 45, 55, 65)  risk\_dangerous = trapezoidal\_mf(x\_risk, 55, 65, 100, 100)   # Ввод значений от пользователя  try:  credit\_score = float(input("\nВведите значение кредитного рейтинга (0-1000): "))  risk\_level = float(input("Введите значение уровня риска (0-100%): "))   # Проверка корректности ввода  if credit\_score < 0 or credit\_score > 1000:  print("Ошибка: кредитный рейтинг должен быть в диапазоне 0-1000")  return  if risk\_level < 0 or risk\_level > 100:  print("Ошибка: уровень риска должен быть в диапазоне 0-100")  return   except ValueError:  print("Ошибка: введите числовые значения")  return   # Вычисление степеней принадлежности для кредитного рейтинга  credit\_memberships = {  'Низкий': np.interp(credit\_score, x\_credit, credit\_low),  'Средний': np.interp(credit\_score, x\_credit, credit\_medium),  'Высокий': np.interp(credit\_score, x\_credit, credit\_high),  'Премиальный': np.interp(credit\_score, x\_credit, credit\_premium)  }   # Вычисление степеней принадлежности для уровня риска  risk\_memberships = {  'Безопасный': np.interp(risk\_level, x\_risk, risk\_safe),  'Приемлемый': np.interp(risk\_level, x\_risk, risk\_acceptable),  'Рискованный': np.interp(risk\_level, x\_risk, risk\_risky),  'Опасный': np.interp(risk\_level, x\_risk, risk\_dangerous)  }   # Вывод степеней принадлежности  print("\n=== Степени принадлежности ===")  print("Кредитный рейтинг:")  for category, membership in credit\_memberships.items():  print(f" {category}: {membership:.3f}")   print("\nУровень риска:")  for category, membership in risk\_memberships.items():  print(f" {category}: {membership:.3f}")   # Выполнение пересечения для всех комбинаций  print("\n=== Пересечение нечетких множеств ===")  intersection\_results = {}   for credit\_cat, credit\_mem in credit\_memberships.items():  for risk\_cat, risk\_mem in risk\_memberships.items():  intersection = min(credit\_mem, risk\_mem)  intersection\_results[f"{credit\_cat} рейтинг ∩ {risk\_cat} риск"] = intersection  print(f"{credit\_cat} рейтинг ∩ {risk\_cat} риск: {intersection:.3f}")   # Визуализация  visualize\_results(x\_credit, x\_risk, credit\_low, credit\_medium, credit\_high, credit\_premium,  risk\_safe, risk\_acceptable, risk\_risky, risk\_dangerous,  credit\_score, risk\_level, credit\_memberships, risk\_memberships)   def visualize\_results(x\_credit, x\_risk, credit\_low, credit\_medium, credit\_high, credit\_premium,  risk\_safe, risk\_acceptable, risk\_risky, risk\_dangerous,  credit\_score, risk\_level, credit\_memberships, risk\_memberships):  # Создание графиков  fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(15, 12))   # График 1: Кредитный рейтинг  ax1.plot(x\_credit, credit\_low, 'b-', label='Низкий', linewidth=2)  ax1.plot(x\_credit, credit\_medium, 'g-', label='Средний', linewidth=2)  ax1.plot(x\_credit, credit\_high, 'y-', label='Высокий', linewidth=2)  ax1.plot(x\_credit, credit\_premium, 'r-', label='Премиальный', linewidth=2)  ax1.axvline(x=credit\_score, color='k', linestyle='--', label=f'Введенное значение: {credit\_score}')  ax1.set\_title('Нечеткие множества: Кредитный рейтинг')  ax1.set\_xlabel('Кредитный рейтинг')  ax1.set\_ylabel('Степень принадлежности')  ax1.legend()  ax1.grid(True, alpha=0.3)   # График 2: Уровень риска  ax2.plot(x\_risk, risk\_safe, 'b-', label='Безопасный', linewidth=2)  ax2.plot(x\_risk, risk\_acceptable, 'g-', label='Приемлемый', linewidth=2)  ax2.plot(x\_risk, risk\_risky, 'y-', label='Рискованный', linewidth=2)  ax2.plot(x\_risk, risk\_dangerous, 'r-', label='Опасный', linewidth=2)  ax2.axvline(x=risk\_level, color='k', linestyle='--', label=f'Введенное значение: {risk\_level}%')  ax2.set\_title('Нечеткие множества: Уровень риска')  ax2.set\_xlabel('Уровень риска (%)')  ax2.set\_ylabel('Степень принадлежности')  ax2.legend()  ax2.grid(True, alpha=0.3)   # График 3: Степени принадлежности для кредитного рейтинга  credit\_categories = list(credit\_memberships.keys())  credit\_values = list(credit\_memberships.values())  bars1 = ax3.bar(credit\_categories, credit\_values, color=['blue', 'green', 'yellow', 'red'], alpha=0.7)  ax3.set\_title('Степени принадлежности: Кредитный рейтинг')  ax3.set\_ylabel('Степень принадлежности')  ax3.grid(True, alpha=0.3)   # Добавление значений на столбцы  for bar, value in zip(bars1, credit\_values):  ax3.text(bar.get\_x() + bar.get\_width() / 2, bar.get\_height() + 0.01,  f'{value:.3f}', ha='center', va='bottom')   # График 4: Степени принадлежности для уровня риска  risk\_categories = list(risk\_memberships.keys())  risk\_values = list(risk\_memberships.values())  bars2 = ax4.bar(risk\_categories, risk\_values, color=['blue', 'green', 'yellow', 'red'], alpha=0.7)  ax4.set\_title('Степени принадлежности: Уровень риска')  ax4.set\_ylabel('Степень принадлежности')  ax4.grid(True, alpha=0.3)   # Добавление значений на столбцы  for bar, value in zip(bars2, risk\_values):  ax4.text(bar.get\_x() + bar.get\_width() / 2, bar.get\_height() + 0.01,  f'{value:.3f}', ha='center', va='bottom')   plt.tight\_layout()  plt.show()   if \_\_name\_\_ == "\_\_main\_\_":  main() |