|  |  |
| --- | --- |
| ***Вариант задания №7*** | |
| *Код программы:* | *Код оптимизированной программы:* |
| ***import numpy as np import matplotlib.pyplot as plt  # Параметры задачи left\_border = -2.0 right\_border = 20.0 n\_points\_list = [5, 10, 20, 50, 100]  # Создание фигуры для графиков plt.figure(figsize=(15, 10))  # Вывод результатов print("Результаты поиска интервала с минимумом функции f(x) = (x-8)^2") print("Метод: Пассивный поиск (базовые конструкции)")  # Цикл по количеству точек for i, n\_points in enumerate(n\_points\_list, 1):  # Инициализация переменных для хранения результатов  min\_x = 0  min\_value = float('inf')   # Создание массивов для хранения точек  x\_points = np.linspace(left\_border, right\_border, n\_points)  y\_points = (x\_points - 8) \*\* 2   # Поиск минимума  min\_index = np.argmin(y\_points)  min\_x = x\_points[min\_index]  min\_value = y\_points[min\_index]   # Вывод результатов для текущего количества точек  print(f"\nКоличество точек: {n\_points}")  print(f"Координата минимума: {min\_x:.4f}")  print(f"Значение минимума: {min\_value:.4f}")   # Создание subplot  plt.subplot(2, 3, i)   # Создание более гладкой кривой для отрисовки  x\_smooth = np.linspace(left\_border, right\_border, 200)  y\_smooth = (x\_smooth - 8) \*\* 2   # Построение графика функции  plt.plot(x\_smooth, y\_smooth, 'b-', label='f(x) = (x-8)²')   # Точки выборки  plt.scatter(x\_points, y\_points, color='red', label='Выборка')   # Точка минимума  plt.scatter(min\_x, min\_value, color='green', s=100, label='Минимум')   plt.title(f'Поиск минимума\n({n\_points} точек)')  plt.xlabel('x')  plt.ylabel('f(x)')  plt.legend()  plt.grid(True)  # Корректировка расположения субплотов plt.tight\_layout()  # Отображение графиков plt.show()*** | ***import numpy as np import matplotlib.pyplot as plt from typing import Tuple, List   def target\_function(x: float) -> float:  """  Целевая функция для поиска минимума.  f(x) = (x-8)^2  """  return (x - 8) \*\* 2   def find\_minimum\_in\_interval(  left\_border: float,  right\_border: float,  n\_points: int ) -> Tuple[float, float]:  """  Находит минимум функции в заданном интервале с использованием дискретной выборки.   Args:  left\_border (float): Левая граница интервала  right\_border (float): Правая граница интервала  n\_points (int): Количество точек для выборки   Returns:  Tuple[float, float]: Координата и значение минимума  """  # Создание массивов для хранения точек  x\_points = np.linspace(left\_border, right\_border, n\_points)  y\_points = target\_function(x\_points)   # Поиск минимума  min\_index = np.argmin(y\_points)  return x\_points[min\_index], y\_points[min\_index]   def plot\_function\_search(  left\_border: float,  right\_border: float,  n\_points: int,  subplot\_pos: int ):  """  Визуализация поиска минимума функции.   Args:  left\_border (float): Левая граница интервала  right\_border (float): Правая граница интервала  n\_points (int): Количество точек для выборки  subplot\_pos (int): Позиция subplot  """  # Поиск минимума  min\_x, min\_value = find\_minimum\_in\_interval(left\_border, right\_border, n\_points)   # Вывод результатов для текущего количества точек  print(f"\nКоличество точек: {n\_points}")  print(f"Координата минимума: {min\_x:.4f}")  print(f"Значение минимума: {min\_value:.4f}")   # Создание subplot  plt.subplot(2, 3, subplot\_pos)   # Создание более гладкой кривой для отрисовки  x\_smooth = np.linspace(left\_border, right\_border, 200)  y\_smooth = target\_function(x\_smooth)   # Создание массивов для хранения точек  x\_points = np.linspace(left\_border, right\_border, n\_points)  y\_points = target\_function(x\_points)   # Построение графика функции  plt.plot(x\_smooth, y\_smooth, 'b-', label='f(x) = (x-8)²')   # Точки выборки  plt.scatter(x\_points, y\_points, color='red', label='Выборка')   # Точка минимума  plt.scatter(min\_x, min\_value, color='green', s=100, label='Минимум')   plt.title(f'Поиск минимума\n({n\_points} точек)')  plt.xlabel('x')  plt.ylabel('f(x)')  plt.legend()  plt.grid(True)   def main():  """  Основная функция для выполнения поиска минимума и визуализации.  """  # Параметры задачи  left\_border = -2.0  right\_border = 20.0  n\_points\_list = [5, 10, 20, 50, 100]   # Создание фигуры для графиков  plt.figure(figsize=(15, 10))   # Вывод результатов  print("Результаты поиска интервала с минимумом функции f(x) = (x-8)^2")  print("Метод: Пассивный поиск (с использованием функций)")   # Визуализация для каждого количества точек  for i, n\_points in enumerate(n\_points\_list, 1):  plot\_function\_search(left\_border, right\_border, n\_points, i)   # Корректировка расположения субплотов  plt.tight\_layout()   # Отображение графиков  plt.show()   # Точка входа в программу if \_\_name\_\_ == "\_\_main\_\_":  main()*** |
| *Расчет метрики:* | *Расчет метрики:* |
|  | |