**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЕТ**

**по лабораторной работе №4**

**по дисциплине «Алгоритмы и структуры данных»**

**на тему «Исследование генетического алгоритма. Изучение различных кодировок генотипа»**

**Вариант – 5**

**Студент гр. 23Б16-пу**

**Горыцнцев Р.Н.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

Оглавление

1. [Цель работы 3](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013329)
2. [Описание задачи (формализация задачи) 3](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013330)
3. [Теоретическая часть 4](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013331)
4. [Основные шаги программы 7](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013332)
5. [Блок схема программы 9](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013333)
6. [Описание программы 10](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013334)
7. [Рекомендации пользователя 11](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013335)
8. [Контрольный пример 11](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013338)
9. [Анализ результатов работы 13](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013339)
10. [Вывод 15](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013340)
11. [Листинг 16](file:///C:\Users\Romchik\Downloads\Telegram%20Desktop\Отчет.docx#_Toc180013340)

# **Цель работы**

Цель данной лабораторной работы — исследовать два основных способа кодирования генотипа хромосом в генетическом алгоритме и проверить их эффективность.

# **Описание задачи (формализация задачи)**

1. Изучение особенностей кодирования генетических алгоритмов:
   * Изучить основные способы кодирования генотипа в генетических алгоритмах
   * Ознакомиться с особенностями каждого способа кодирования, их преимуществами и недостатками.
2. Написание программы поиска минимума функции:
   * Разработать программу на выбранном языке программирования, реализующую генетический алгоритм для поиска минимума выбранной функции.
3. Тестирование программы:
   * Протестировать программу на выбранной тестовой функции для вариантов кодирования генотипа.
   * Сравнить результаты работы алгоритма для разных способов кодирования.
4. Анализ результатов и модернизация алгоритма:
   * Проанализировать результаты работы алгоритма для разных способов кодирования.
   * Описать и выполнить модернизацию части алгоритма: кроссинговер
   * Провести отладку алгоритма, комментарии и выводы.

# **Теоретическая часть**

**Генетические алгоритмы** (ГА) — это эвристические алгоритмы поиска, используемые для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию. ГА основаны на принципах естественного отбора и генетики.

**Основные этапы ГА:**

1. Инициализация популяции: Создание начальной популяции особей (решений), каждая из которых представляет собой возможное решение задачи.
2. Оценка приспособленности: Вычисление значения целевой функции (приспособленности) для каждой особи в популяции.
3. Выбор родителей: Выбор особей для скрещивания (кроссинговера) на основе их приспособленности.
4. Скрещивание (кроссинговер): Обмен генетической информацией между родителями для создания потомков.
5. Мутация: Случайное изменение генов потомков для внесения разнообразия в популяцию.
6. Формирование новой популяции: Замена старых особей на новых (потомков) для следующей итерации.
7. Проверка условия остановки: Повторение шагов 2-6 до выполнения условия остановки (например, достижение максимального числа итераций или достаточной точности решения).

**Кодирование генотипа**

Генотип — это представление решения задачи в виде хромосомы, которая состоит из генов. Существуют различные способы кодирования генотипа, но два основных способа — это бинарное кодирование и кодирование с помощью действительных чисел.

1) Бинарное кодирование

Бинарное кодирование — это наиболее распространенный способ представления генотипа в генетических алгоритмах. В этом случае каждый ген представляет собой бит (0 или 1), а вся хромосома — это последовательность битов.

Преимущества:

* Простота реализации.
* Эффективное использование памяти.
* Легкость применения операторов кроссинговера и мутации.

Недостатки:

* Низкая точность для задач с непрерывными переменными.
* Необходимость преобразования бинарного кода в реальные значения переменных.

2) Кодирование с помощью действительных чисел (вещественное)

Кодирование с помощью действительных чисел предполагает, что каждый ген представляет собой действительное число из заданного диапазона. Вся хромосома — это набор действительных чисел.

Преимущества:

* Высокая точность для задач с непрерывными переменными.
* Не требуется преобразование кода в реальные значения.

Недостатки:

* Сложность реализации операторов кроссинговера и мутации.
* Более высокие требования к памяти.

**Скрещивание**

Кроссинговер (скрещивание) — это ключевой оператор генетических алгоритмов, отвечающий за обмен генетической информацией между родителями и создание новых особей (потомков). Существуют различные типы кроссинговера, рассмотрим 2 из них: одноточечный и однородный.

В **одноточечном** кроссинговере случайным образом выбирается точка разделения (разреза) на хромосомах родителей. Далее происходит обмен частями хромосом до и после точки разделения между родителями, в результате чего формируются две новые хромосомы-потомки.

* Преимущества:
  + Простота реализации.
  + Эффективен для задач, где важна сохранность целостности генов.
* Недостатки:
  + Может привести к недостаточному разнообразию потомков.
  + Неэффективен для задач с сильной взаимосвязью генов.

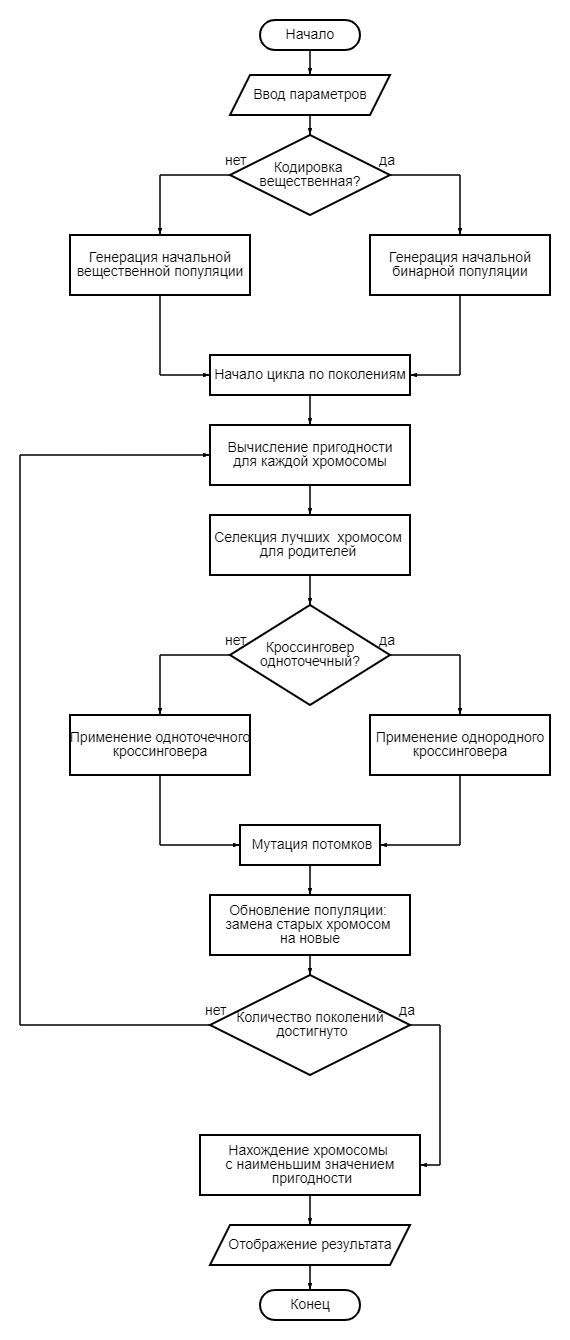
  В **однородном** кроссинговере каждый ген потомка формируется случайным образом, независимо от других генов, путем выбора соответствующего гена одного из родителей.

* Преимущества:
  + Позволяет получить большее разнообразие потомков.
  + Эффективен для задач с сильной взаимосвязью генов.
* Недостатки:
  + Сложность реализации.
  + Может привести к потере полезных комбинаций генов.

# **Основные шаги программы**

1. **Генерация начальной популяции:**
   * Вещественная кодировка: Генерация начальной популяции с использованием равномерного распределения в заданном диапазоне.
   * Бинарная кодировка: Генерация начальной популяции с использованием случайных бит.
2. **Декодирование бинарной хромосомы:**
   * Преобразование бинарной хромосомы в вещественные значения переменных x1 и x2 с использованием заданного диапазона.
3. **Вычисление пригодности каждой хромосомы:**
   * Для каждой хромосомы в популяции вычисляется значение тестовой функции (пригодность).
   * В случае бинарной кодировки сначала происходит декодирование хромосомы в вещественные значения.
4. **Селекция:**
   * Выбор родительских хромосом для скрещивания на основе их пригодности.
   * Используется метод отбора лучших особей.
5. **Кроссинговер:**
   * Одноточечный кроссинговер: Разделение родительских хромосом в случайной точке и обмен частями для создания потомков.
   * Однородный кроссинговер: Каждый ген потомка наследуется от одного из родителей с определенной вероятностью.
6. **Мутация:**
   * Случайное изменение генов потомков с заданной вероятностью.
   * В случае вещественной кодировки гены изменяются на случайное значение в пределах заданного диапазона.
   * В случае бинарной кодировки гены инвертируются (0 на 1 или 1 на 0).
7. **Генетический алгоритм:**
   * Инициализация начальной популяции в зависимости от выбранного типа кодировки.
   * Повторение процесса селекции, кроссинговера и мутации для заданного числа поколений.
   * Выбор лучшей хромосомы по итогам всех поколений.
8. **Функция для расчета хромосом:**
   * Получение параметров генетического алгоритма от пользователя через графический интерфейс.
   * Вызов функции генетического алгоритма с заданными параметрами.
   * Отображение лучшего решения и значения функции в графическом интерфейсе.
9. **Запуск программы**

# **Блок схема программы**



*Рис 1. Блок схема программы*

# **Описание программы**

Программа реализована на языке Python 3.12 с использованием следующих пакетов: tkinter, numpy, random. В генетическом алгоритме используются 12 функций. В таблицах представлено описание функций генетического алгоритма.

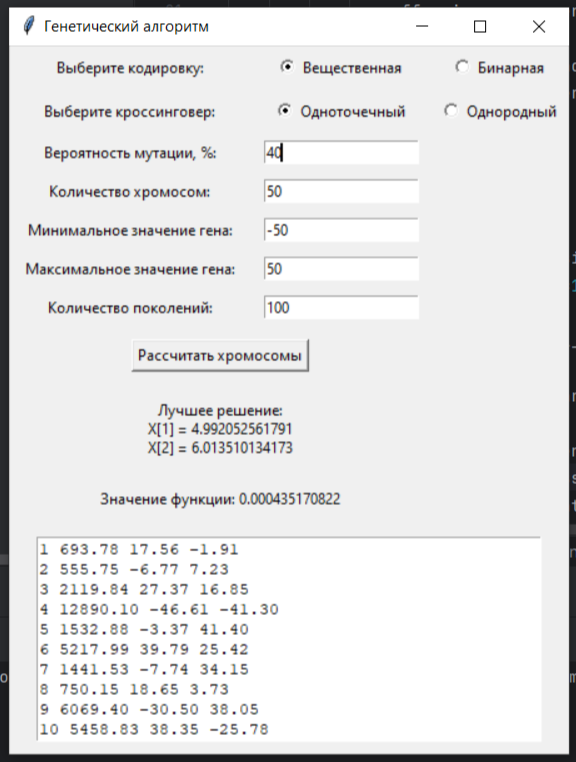
*Таблица 1. lab\_4.py*

|  |  |  |
| --- | --- | --- |
| Функция | Описание | Возвращаемое значение |
| test\_function | Определяет тестовую функцию, минимум которой будет искать генетический алгоритм. | float |
| generate\_real\_population | Генерирует начальную популяцию с использованием вещественной кодировки. | np.ndarray |
| generate\_binary\_population | Генерирует начальную популяцию с использованием бинарной кодировки. | np.ndarray |
| decode\_binary\_chromosome | Преобразует бинарную хромосому в вещественные значения x1 и x2. | tuple |
| calculate\_fitness | Вычисляет пригодность каждой хромосомы в популяции. | np.ndarray |
| selection | Выбирает родительские хромосомы для скрещивания на основе их пригодности. | np.ndarray |
| single\_point\_crossover | Выполняет одноточечный кроссинговер для создания потомков. | np.ndarray |
| uniform\_crossover | Выполняет однородный кроссинговер для создания потомков. | np.ndarray |
| mutation | Выполняет мутацию потомков с заданной вероятностью. | np.ndarray |
| genetic\_algorithm | Реализует основной цикл генетического алгоритма. | tuple |
| calculate\_chromosomes | Функция для расчета хромосом и отображения результатов в графическом интерфейсе. | None |
| display\_chromosomes | Функция для отображения таблицы с хромосомами и их пригодностью. | None |

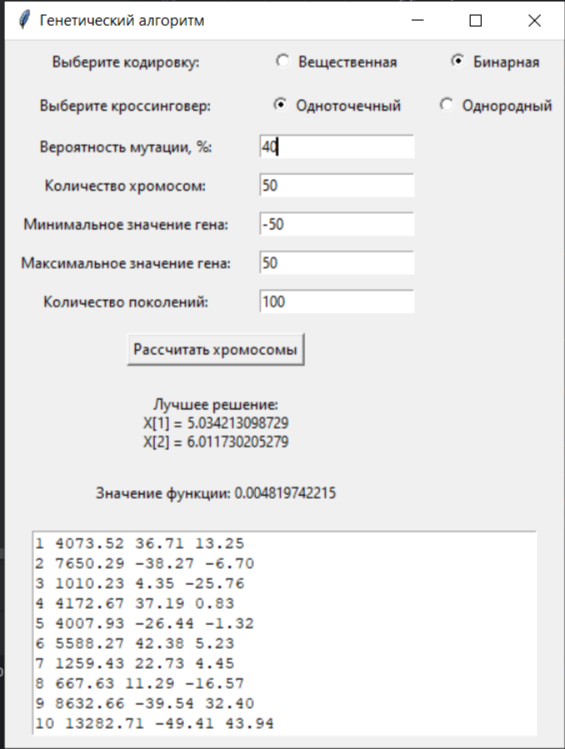
# **Рекомендации пользователя**

Установите необходимые библиотеки: tkinter, numpy, random. Перед запуском программы убедитесь, что все параметры заданы корректно: вероятность мутации, количество хромосом, минимальное и максимальное значения генов, количество поколений. Выберите кодировку (вещественная, бинарная) и тип кроссинговера (одноточечный, однородный). После запуска проверьте результаты в полях "Лучшее решение" и "Значение функции", а также таблицу с хромосомами на корректность данных.

# **Контрольный пример**



*Рис 2: пример окна программы с вещественной кодировкой*



*Рис 3: пример окна программы с бинарной кодировкой*

# **Анализ результатов работы:**

Анализ проводится при 40% вероятности мутации и значении гена от -50 до 50.

*Таблица 2: тесты вещественной кодировки с одноточечным кроссинговером*

|  |  |  |  |
| --- | --- | --- | --- |
| Количество хромосом | Количество поколений | Лучшее решение | Значение функции |
| 10 | 100 | (4.989240244612, 6.050807184050) | 0.003044459295 |
| 80 | 100 | (4.991510158384, 5.984000612772) | 0.000544290034 |
| 100 | 10 | (4.963885679917, 6.039196064618) | 0.006753307942 |
| 100 | 80 | (5.008934047098, 6.026257760822) | 0.001008738794 |

*Таблица 3: тесты бинарной кодировки с одноточечным кроссинговером*

|  |  |  |  |
| --- | --- | --- | --- |
| Количество хромосом | Количество поколений | Лучшее решение | Значение функции |
| 10 | 100 | (6.304985337243, 6.011730205279) | 6.812084519397 |
| 80 | 100 | (5.034213098729, 6.011730205279) | 0.004819742215 |
| 100 | 10 | (4.838709677419, 5.425219941349) | 0.434430388456 |
| 100 | 80 | (5.034213098729, 6.304985337243) | 0.097698200432 |

*Таблица 4: тесты вещественной кодировки с однородным кроссинговером*

|  |  |  |  |
| --- | --- | --- | --- |
| Количество хромосом | Количество поколений | Лучшее решение | Значение функции |
| 10 | 100 | (4.971571800168, 7.800656741787) | 3.245597351927 |
| 80 | 100 | (5.000740425022, 5.978622434621) | 0.000459193218 |
| 100 | 10 | (4.479204164526, 5.292174908397) | 1.585929569291 |
| 100 | 80 | (5.010872761865, 6.008322649438) | 0.000542134296 |

*Таблица 5: тесты бинарной кодировки с однородным кроссинговером*

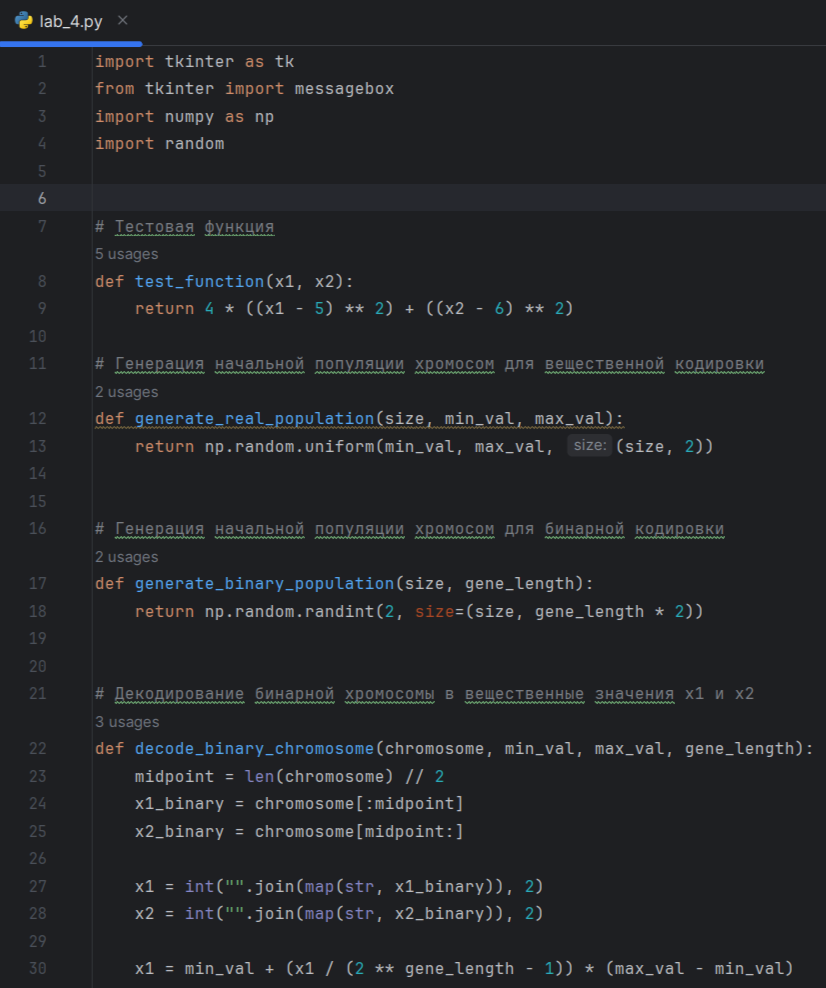
|  |  |  |  |
| --- | --- | --- | --- |
| Количество хромосом | Количество поколений | Лучшее решение | Значение функции |
| 10 | 100 | (5.034213098729, -0.048875855327) | 36.593581257662 |
| 80 | 100 | (5.034213098729, 6.304985337243) | 0.097698200432 |
| 100 | 10 | (5.034213098729, 6.011730205279) | 0.004819742215 |
| 100 | 80 | (5.034213098729, 6.011730205279) | 0.004819742215 |

1. **Вещественная кодировка:**
   * Вещественная кодировка, как правило, показывает лучшие результаты по сравнению с бинарной кодировкой.
   * Она обеспечивает более точные решения и быструю сходимость.
2. **Бинарная кодировка:**
   * Бинарная кодировка может давать приемлемые результаты, но часто требует большего количества поколений для сходимости.
   * Она может страдать от проблем с точностью, особенно при больших диапазонах значений.
3. **Одноточечный кроссинговер:**
   * Одноточечный кроссинговер показывает стабильные результаты для обоих типов кодировки.
   * Он хорошо работает с вещественной кодировкой, обеспечивая быструю сходимость и точные решения.
4. **Однородный кроссинговер:**
   * Однородный кроссинговер может давать более разнообразные решения, но иногда приводит к большему разбросу результатов.
   * Он может быть полезен для задач, где требуется исследование большего пространства поиска.

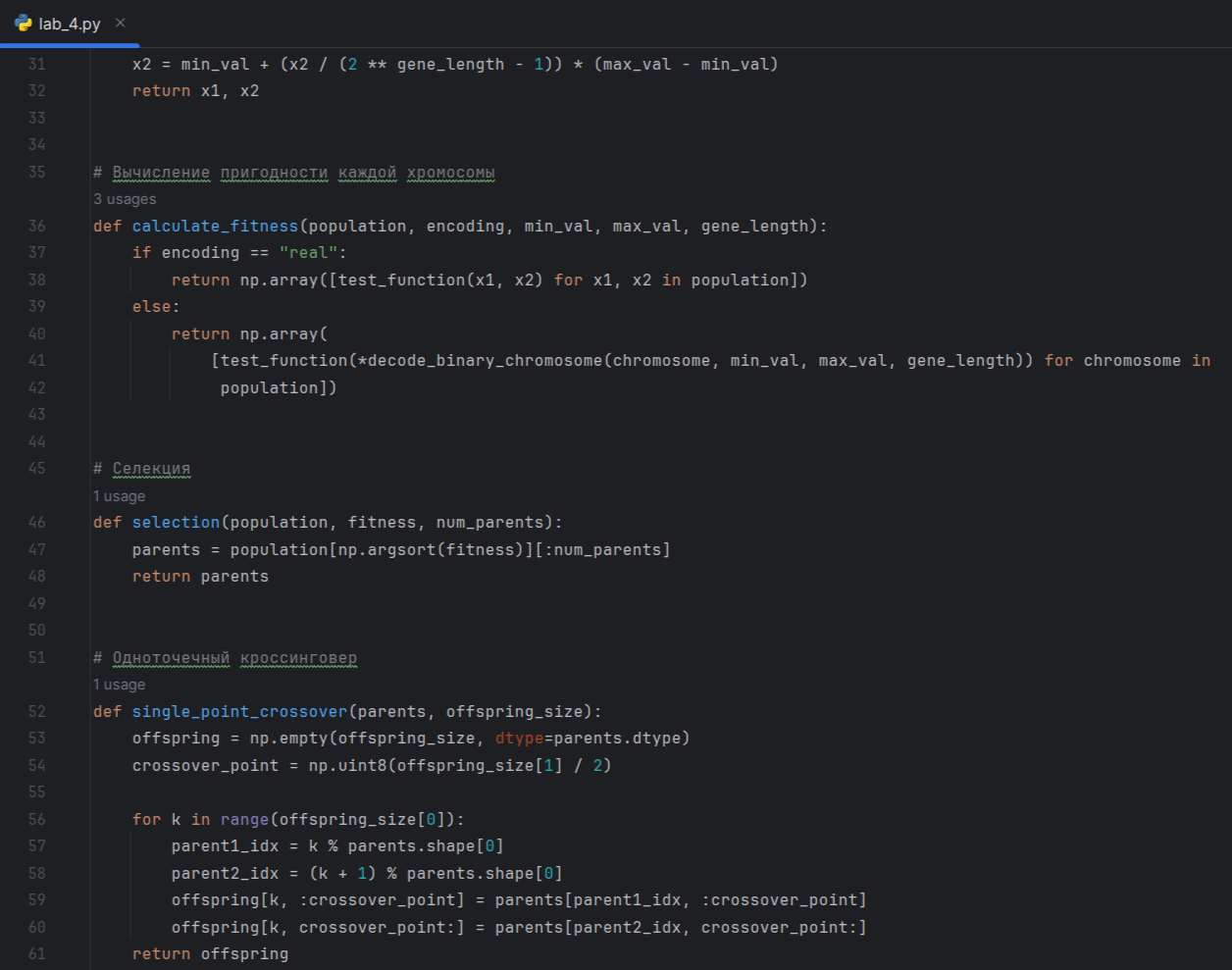
**Вывод:**

В ходе лабораторной работы было проведено исследование двух основных способов кодирования генотипа в генетическом алгоритме: вещественной и бинарной кодировки, а также двух типов кроссинговера: одноточечного и однородного. Результаты показали, что вещественная кодировка обеспечивает более высокую точность и быструю сходимость, особенно при использовании одноточечного кроссинговера. Бинарная кодировка, хотя и может давать приемлемые результаты, требует большего количества поколений для сходимости и страдает от проблем с точностью при больших диапазонах значений. Однородный кроссинговер в бинарной кодировке может приводить к большему разнообразию решений, но также и к большему разбросу результатов. В целом, выбор оптимального способа кодирования и типа кроссинговера зависит от специфики решаемой задачи и требований к точности и скорости сходимости алгоритма.

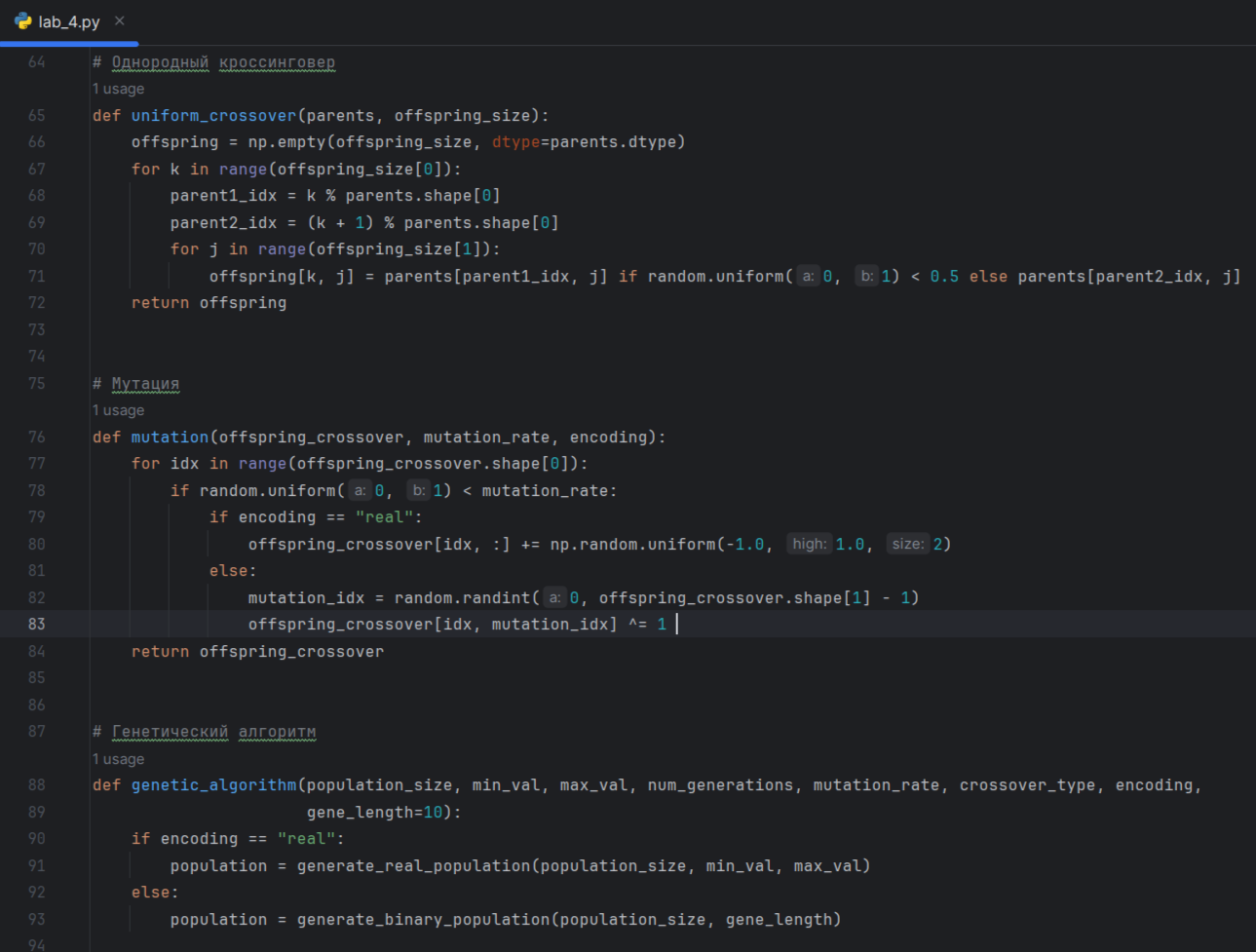
# **Листинг**



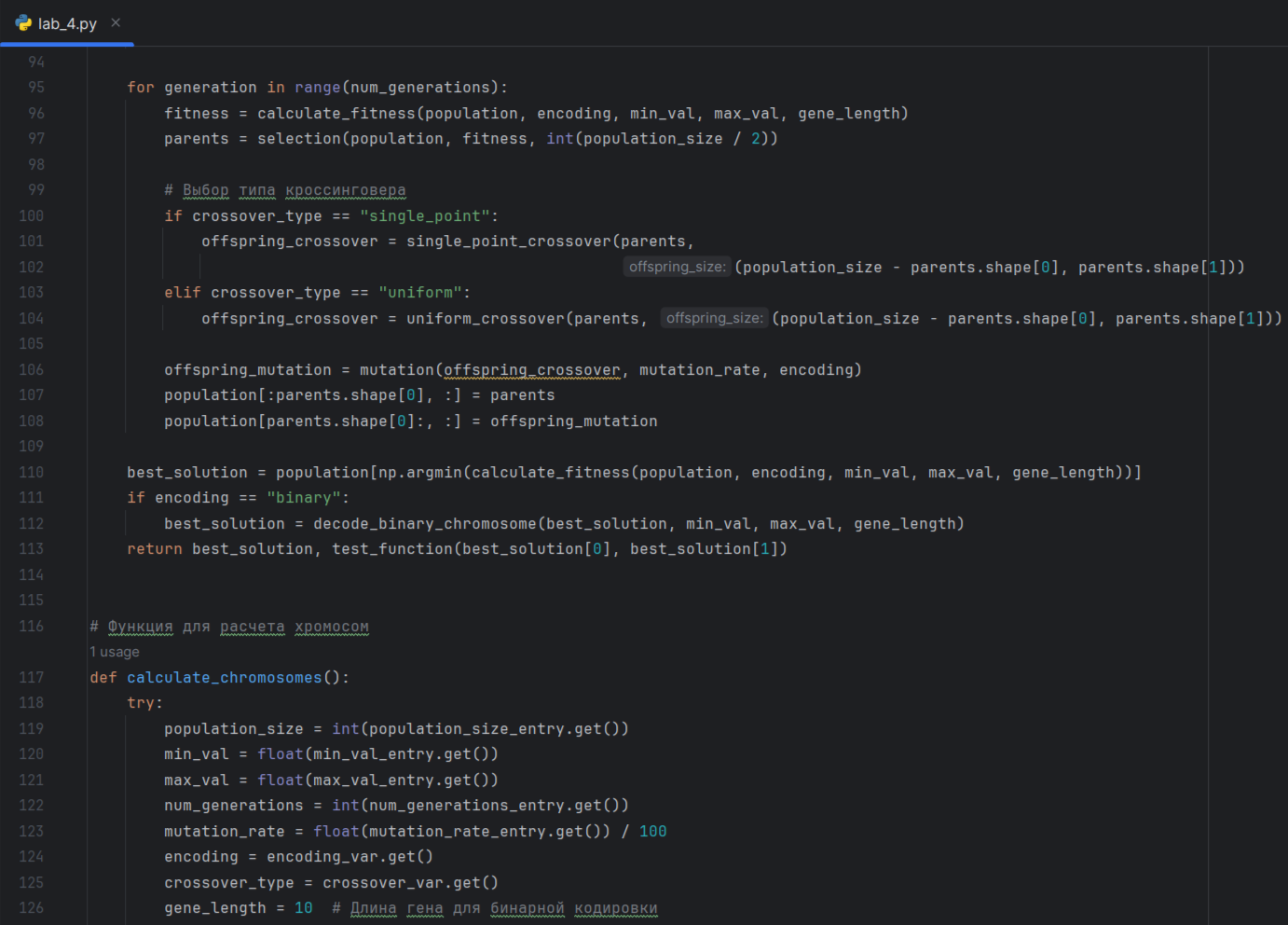
*Рис 4: Листинг*



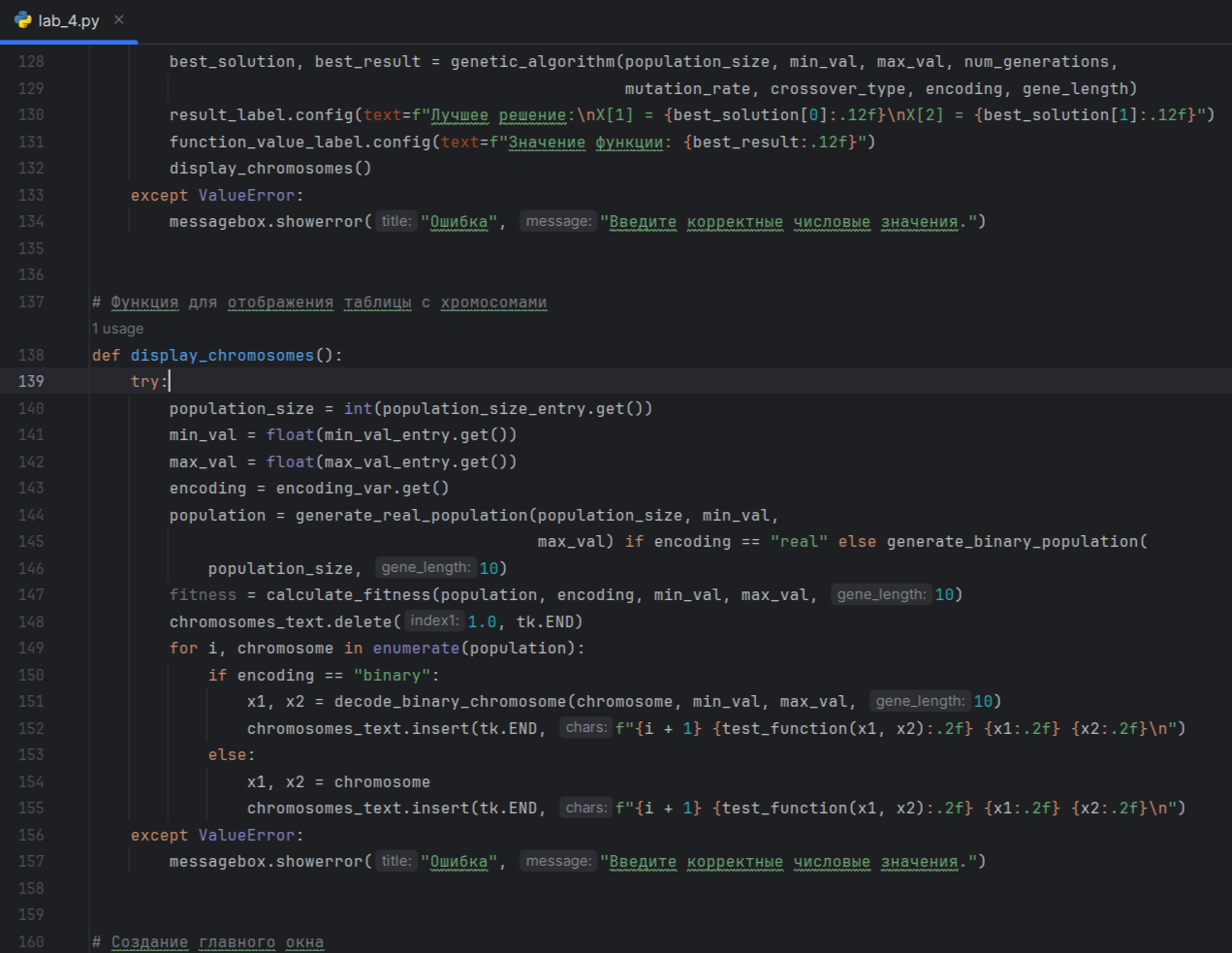
*Рис 5: Листинг*



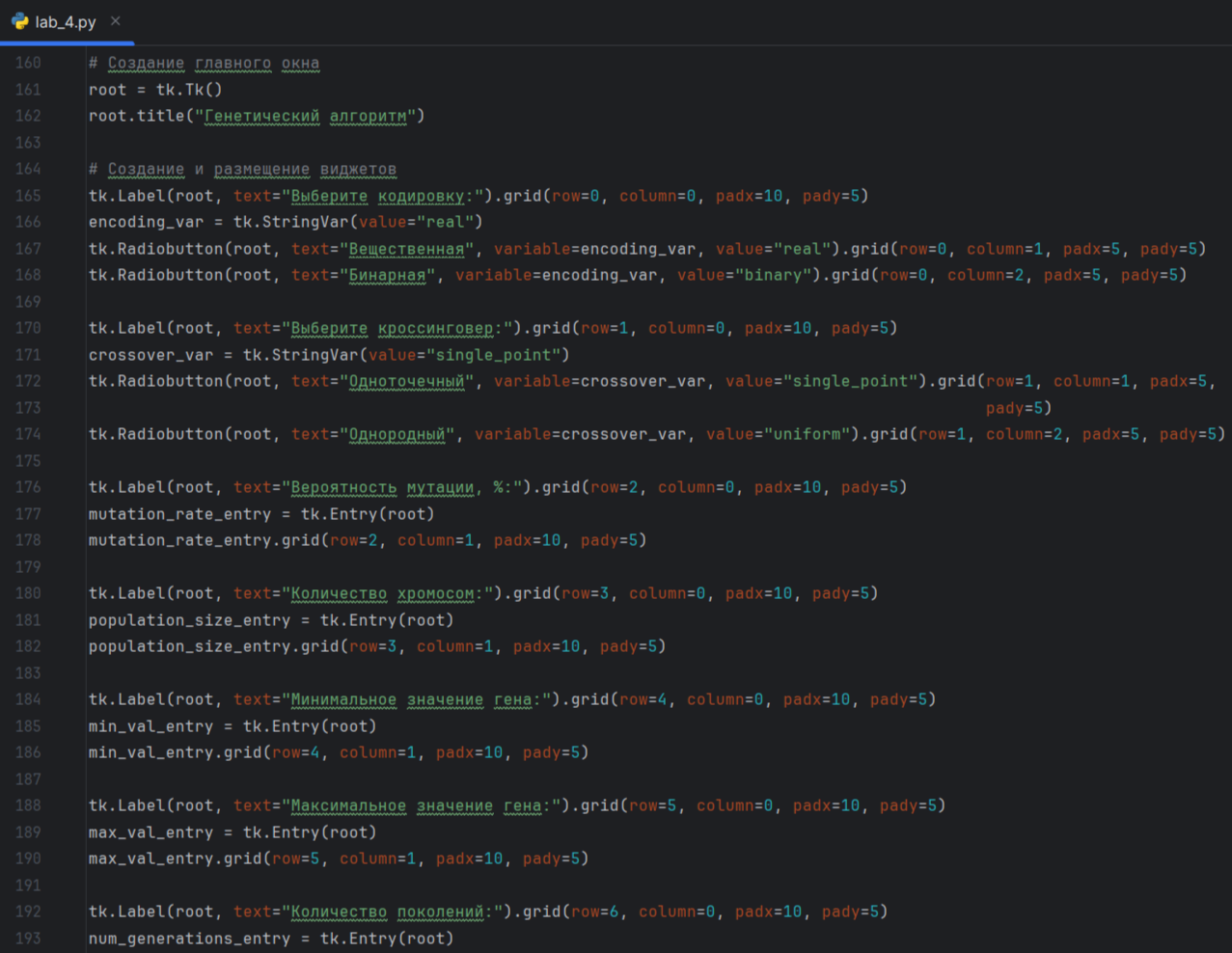
*Рис 6: Листинг*



*Рис 7: Листинг*



*Рис 8: Листинг*



*Рис 9: Листинг*



*Рис 10: Листинг*