

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра прикладной математики и процессов управления.

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Системное программирование в Linux»
на тему «Шахматы»

Студент гр. 22.Б16-пу

Поляков Л.С.

Преподаватель

Киямов Ж. У.

Санкт-Петербург
2023 г.

Оглавление

1. Введение.....	3
2. Цель работы.....	3
3. Задача.....	3
4. Описание программы.....	3
5. Рекомендации пользователю.....	4
6. Рекомендации программисту.....	4
7. Контрольный пример.....	5
8. Вывод.....	7

1. Введение

Целью данной работы является создание программы, которая генерирует начальное положение на шахматной доске и вычисляет три лучших хода в данной позиции. Программа написана на языке C++ и использует набор шахматных фигур, представленных в виде объектов класса Piece.

2. Цель работы

Целью данной работы является разработка программы, которая:

1. Генерирует случайное начальное положение на шахматной доске и находит три лучших хода в данной позиции.
2. Каждый из трёх лучших ходов дополняется еще двумя (или менее, если обнаружен шах) дополнительными ходами, развивая данную позицию дальше.

3. Задачи:

Для достижения поставленной цели были решены следующие задачи:

- Реализация генерации начальной позиции на доске.
- Реализация проверок на допустимость ходов шахматных фигур.
- Вычисление оценки каждого возможного хода.
- Выбор трех лучших ходов на основе оценок.

4. Описание программы

class Piece {}	Класс, отвечающий за фигуры на доске
void generateRandomPosition()	Генерирует случайную расстановку
void printBoard()	Выводит шахматную доску на экран
bool isValidCell()	Проверяет можно ли сходить на определенную клетку
bool isValidPawnMove()	Определяет правильность хода пешки
bool isValidKingMove()	Определяет правильность хода короля
bool isValidBishopMove()	Определяет правильность хода слона
bool isValidRookMove()	Определяет правильность хода ладьи
bool isValidKnightMove()	Определяет правильность хода коня
bool isValidMove()	Определяет правильность хода фигуры

	на данной клетке
<code>bool isCheck()</code>	Проверка на шах
<code>bool isCheckMate()</code>	Проверка на мат
<code>bool canCapturePiece()</code>	Проверяет можно ли съесть фигуру
<code>bool canSavePiece()</code>	Проверяет можно ли спасти фигуру
<code>int getPieceValue()</code>	Возвращает ценность фигуры
<code>bool isValidCoordinate()</code>	Проверяет правильность координат
<code>int evaluateMove()</code>	Вычисляет ценность хода
<code>std::vector<std::pair<int, int>></code> <code>findBestMoves()</code>	Вычисляет лучшие ходы из данной позиции
<code>void printMoves()</code>	Выводит найденные ходы
<code>int main()</code>	Главная функция, ответственная за запуск всех функций и ввод входных данных

5. Рекомендации пользователю

Для успешной компиляции и запуска программы рекомендуется использовать компилятор, поддерживающий стандарты C++ 17-20. Также необходимо иметь доступ к стандартным библиотекам C++, таким как `<iostream>`, `<ctime>`, `<cstdlib>` и другим.

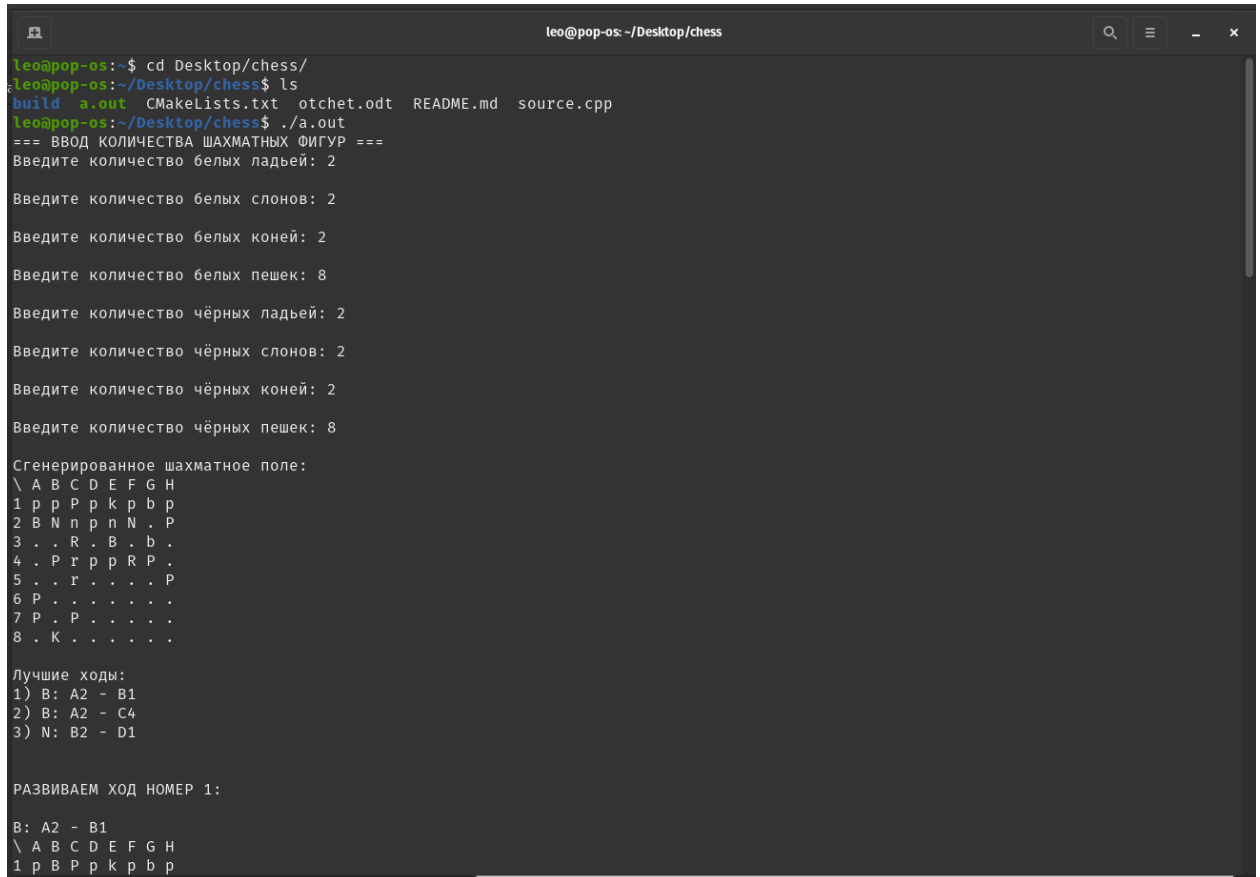
6. Рекомендации программисту

Для дальнейшего развития программы и расширения её функциональности можно рассмотреть следующие направления:

- Реализация ввода и вывода данных через консоль или файлы.
- Доработка алгоритма определения лучших ходов для более точной оценки ситуации.
- Реализация игрового интерфейса для взаимодействия с пользователем.

Входные данные программы представляют собой количество фигур каждого цвета, которое указывает пользователь вручную, что в последствии помогает создать начальное положение на шахматной доске, которое генерируется функцией `generateRandomPosition`. Выходные данные - это три лучших хода, вычисленных функцией `findBestMoves`, которые дополняются еще 1 или 2 ходами, а затем выводятся на экран с помощью `printBoard`.

7. Контрольный пример



```
leo@pop-os: ~/Desktop/chess
leo@pop-os:~$ cd Desktop/chess/
leo@pop-os:~/Desktop/chess$ ls
build  a.out  CMakeLists.txt  otchet.odt  README.md  source.cpp
leo@pop-os:~/Desktop/chess$ ./a.out
=== ВВОД КОЛИЧЕСТВА ШАХМАТНЫХ ФИГУР ===
Введите количество белых ладьей: 2

Введите количество белых слонов: 2

Введите количество белых коней: 2

Введите количество белых пешек: 8

Введите количество чёрных ладьей: 2

Введите количество чёрных слонов: 2

Введите количество чёрных коней: 2

Введите количество чёрных пешек: 8

Сгенерированное шахматное поле:
\ A B C D E F G H
1 p p P p k p b p
2 B N n p n N . P
3 . . R . B . b .
4 . P r p r R P .
5 . . r . . . . P
6 P . . . . . . .
7 P . P . . . . .
8 . K . . . . . .

Лучшие ходы:
1) B: A2 - B1
2) B: A2 - C4
3) N: B2 - D1

РАЗВИВАЕМ ХОД НОМЕР 1:

B: A2 - B1
\ A B C D E F G H
1 p B P p k p b p
```

Рис. 1 (Для начала необходимо зайти в директорию с программой. Затем запустить файл a.out. После запуска программы пользователю необходимо ввести количество каждой из фигур (белых и черных). Вывод программы показан на данном рисунке)

```
leo@pop-os: ~/Desktop/chess
8 . K . . . . .
.
РАЗВИВАЕМ ХОД НОМЕР 3:
N: B2 - D1
\ A B C D E F G H
1 p p P N k p b p
2 B . n p n N . P
3 . . R . B . b .
4 . P r p p R P .
5 . . r . . . . P
6 P . . . . .
7 P . P . . . .
8 . K . . . . .

N: F2 - H1
\ A B C D E F G H
1 p p P N k p b N
2 B . n p n . . P
3 . . R . B . b .
4 . P r p p R P .
5 . . r . . . . P
6 P . . . . .
7 P . P . . . .
8 . K . . . . .

N: H1 - G3
\ A B C D E F G H
1 p p P N k p b .
2 B . n p n . . P
3 . . R . B . N .
4 . P r p p R P .
5 . . r . . . . P
6 P . . . . .
7 P . P . . . .
8 . K . . . . .

leo@pop-os:~/Desktop/chess$
```

Рис. 2 (Таким образом, программа выводит 3 лучших хода и для каждого из этих ходов выводит 1 или 2 следующих за ними лучших хода.)

8. Вывод

В результате выполнения данной работы была разработана программа, способная генерировать начальную расстановку шахматных фигур и вычислять три лучших хода в заданной позиции.

Ссылка на GitHub: <https://github.com/p0le0n/chessHW>