**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«Национальный исследовательский технологический университет**

**«МИСИС»**

**Институт информационных технологий и компьютерных наук**

**Кафедра инженерной кибернетики**

**Прикладная математика**

**КУРСОВАЯ РАБОТА**

**по дисциплине «Объектно-ориентированное программирование»**

**Консольное приложение «Игры»**

Выполнили:

Студентки 1 курса группы БПМ-24-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К.Е. Тимофеева

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Г. Тестова

Преподаватель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Еременко

**Москва 2025**

## Содержание

1. Введение…………………………………………………………………...
2. Цель работы……………………………………………………………….
3. Задачи……………………………………………………………………...
4. Общая структура проекта…………………………………………………
5. Нейронные сети и глубинное обучение……..…………………………..
   1. Виды нейронных сетей…………………………………………….
   2. Применение нейронных сетей…………………………………….
   3. Глубинное обучение……………………………………………….
6. Свёрточные сети………………………………………………………......
7. План обучения модели……………………………………………………
   1. Дополнительные функции………………………………………...
8. Тестирование бота………………………………………………………...
9. Заключение…………………………………………………………..........
10. Список используемой литературы………………………………………
11. Приложения……………………………………………………………….

**Введение**

В рамках данной курсовой работы применяются накопленные знания из дисциплины «Объектно-ориентированное программирование» при разработке консольного приложения «Игры». Программа предназначена для развлекательных целей.

Консольное приложение "Игры" представляет собой программу, написанную на языке C++, которая включает в себя две классические игры:

* "Угадай число" – игра, в которой пользователь должен угадать загаданное компьютером число.
* "Змейка" – классическая аркадная игра, где игрок управляет змейкой, собирая еду и избегая столкновений с границами поля и собственным телом.

Приложение разработано с использованием стандартной библиотеки C++ и работает в консольном режиме.

**Цель работы**

Создать консольное приложение, демонстрирующее реализацию двух игр ("Угадай число" и "Змейка") с использованием возможностей языка C++ и объектно-ориентированного программирования.

**Задачи**

1. Изучить материалы, необходимые для написания программы
2. Разработать структуру приложения, продумать логику работы
3. Реализовать игру "Угадай число"
4. Реализовать игру "Змейка" с управлением через клавиши
5. Обеспечить обработку пользовательского ввода и валидацию данных
6. Организовать систему команд, понятных программе

**Общая структура проекта**

project/ (корневая директория проекта)

│

├── game1/ (директория первой игры - "Угадай число")

│ ├── CMakeLists.txt (файл сборки для первой игры)

│ ├── game1.cpp (реализация логики игры)

│ ├── game1.hpp (заголовочный файл)

│ └── game1\_rules.txt (правила игры)

│

├── game2/ (директория второй игры - "Змейка")

│ ├── CMakeLists.txt (файл сборки для второй игры)

│ ├── game2.cpp (реализация логики игры)

│ └── game2.hpp (заголовочный файл)

│ └── game2\_rules.txt (правила игры)

│

├── CMakeLists.txt (главный файл сборки проекта)

├── main.cpp (основной исполняемый файл)

├── help\_message.txt (файл со справочной информацией)

├── start\_message228.txt (стартовое сообщение)

├── .gitignore (файл исключений Git)

└── README.md (документация проекта)

Файл main.cpp является основным исполняемым компонентом. Он содержит точку входа программы и обрабатывает основной пользовательский ввод.

Папка game1 содержит реализацию первой игры («Угадай число»). Заголовочный файл game1.hpp содержит объявление класса и функций для первой игры, game1.cpp – реализацию методов класса и игровой логики в целом. В отдельный файл вынесены правила игры.

Аналогично организована реализация игры «Змейка». Заголовочный файл game2.hpp содержит объявление класса и функций для первой игры, game2.cpp содержит реализацию методов класса и игрового процееса. В файле game2\_rules.txt содержатся правила игры.

Также в папке project лежат вспомогательные файлы start\_message.txt и help\_message.txt, которые содержат стартовое сообщение и справочную информацию по приложению соответственно. Текст вынесен в отдельные файлы для удобства редактирования.

В проекте используется система сборки CMake, с помощью которой все файлы объединяются в общую систему с четкими зависимостями и простым процессом сборки на любой поддерживаемой платформе. Преимущество использования CMake в том, что каждая игра собирается в отдельную библиотеку, возможна независимая компиляция объектов и мы можем с легкостью добавлять новые игры.

**Описание работы**

Для создания