Министерство образования Новосибирской области

ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галущака»

**Разработка справочного приложения для игры в шахматы с элементами обучения**

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения  
для компьютерных систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.202000.010.000ПЗ

Разработал:

Некрасов Я. А.

**Содержание**

[Введение 3](#_Toc165052019)

[Исследовательский раздел 4](#_Toc165052020)

[1.1 Описание предметной области 4](#_Toc165052021)

[1.2 Образ клиента 4](#_Toc165052022)

[1.3 Сценарии использования 5](#_Toc165052023)

[1.4 Сбор и анализ прототипов 5](#_Toc165052024)

[1.4.1 Основные негативные отзывы от пользователей приложений 7](#_Toc165052025)

[Проектирование приложения 8](#_Toc165052026)

[1.5 UI/UX дизайн приложения 8](#_Toc165052027)

[2.2 Выбор технологии, языка и среды программирования 13](#_Toc165052028)

[Разработка мобильного приложения 14](#_Toc165052029)

[2.3 Разработка базы данных 14](#_Toc165052030)

[2.4 Разработка мультимедийного контента 15](#_Toc165052031)

[2.5 Описание используемых плагинов 16](#_Toc165052032)

[2.6 Описание разработанных процедур и функций 17](#_Toc165052033)

[Тестирование 25](#_Toc165052034)

[2.7 Протокол тестирования дизайна приложения 25](#_Toc165052035)

[Заключение 28](#_Toc165052036)

[Библиография 29](#_Toc165052037)

[Приложение Б 37](#_Toc165052038)

[Приложение В 38](#_Toc165052039)

Введение

Шахматы являются популярной темой во всем мире. С конца 20 века компьютеры значительно повлияли на шахматное сообщество. На шахматном сервисе «Lichess.org» происходит около 90 миллионов партий между пользователями сервиса ежемесячно. Создание шахматного приложения может повысить доступность этой игры для развития у пользователя логического мышления, навыков планирования.

Курсовой проект по созданию справочного приложения с элементами обучения позволит пользователю осуществлять игру в шахматы с искусственным противником и получать знания о названиях шахматных позиций для дальнейшего самостоятельного изучеия.

Целью разработки является создание шахматного приложения с элементами обучения.

Задачами курсового проекта в связи с указанной целью являются:

* изучение предметной области;
* рассмотрение приложения с точки зрения пользователя для выявления функций приложения;
* разработка и реализация дизайна приложения;
* написание кода приложения;
* тестирование полученного продукта.

Описание предметной области

Многим людям, имеющим свободное время приходила идея об отдыхе, и для исполнения данного желания имеется множество видов досуга, в число которых входит времяпровождение за настольными играми. Некоторые из настольных игр не только снимают стресс, но также развивают мышление.

Данное явление не обошло стороной популярную настольную игру «шахматы», соревновательная игра между двумя игроками, действие которой происходят на игровом поле, с разметкой из клеток 8 на 8, целью которой является с помощью специального набора фигур таких как: пешка, слон, конь, ладья, ферзь, король, поставить «шах и мат королю оппонента».

Проведение партий шахмат подразделяют по контролю времени, и самыми популярными и общее принятыми, являются классика, рапид и блиц.

При классическом контроле времени на обдумывание каждому игроку дается не мене часа на партию. Наиболее популярный контроль 1.30 часа на 40 ходов. Далее возможны вариант: 30 ходов до конца. 20 минут с добавлением секунд за каждый сделанный ход и некоторые другие.

Рапид или быстрые шахматы – такой формат игры, в котором каждому из игроков дается время на обдумывания более 10, но менее 60 минут. Контроль может быть как с добавлением секунд на каждый сделанный ход, так и без добавления сегодня самый популярный контроль времени рапиде – минут 15 + 10 секунд добавление на каждый ход.

В блице контроль времени: меньше или равно 10 минут каждому до окончания партии.

Выбор технологии, языка и среды программирования

Проект подразумевает использование базы данных, разработки интерфейса, алгоритма по оценке позиции. Для реализации этих компонентов подходит язык программирования Python. Это высокоуровневый язык программирования общего назначения, который подходит для разработки игр и других приложений благодаря своей простоте, читаемости и большому количеству доступных библиотек. Он широко используется как начинающими, так и опытными разработчиками. На нем активно пишется «задние» части сервисов, такие как взаимодействие с базами данных, работа с нейронными сетями и так далее. Использование этого языка упростит разработку текущего решения и позволит использовать его части в дальнейших проектах.

Средой программирования при разработке проекта служила Visual Studio Code, так как она не предъявляет высоких требований к рабочей станции, позволяет расширение функционала при помощи расширений, которые изменяют функционал IDE.

Используемые расширения для среды программирования:

* Error Lens подсвечивает ошибки и предупреждения прямо в строке кода. Делает код более читаемым благодаря инлайн-подсказкам и объяснениям. Экономит время на поиск и исправление ошибок;
* vscode-icons добавляет иконки для файлов и папок в проводнике VS Code. Упрощает визуальную навигацию по проекту, так как файлы Python, JSON, изображения и другие ресурсы отображаются с соответствующими иконками. Улучшает общий пользовательский опыт благодаря красивому и организованному интерфейсу;
* Python — официальное расширение для поддержки языка Python. **Подсветка синтаксиса** делает код более читабельным. **Встроенный терминал** позволяет запускать Python-скрипты непосредственно из VS Code;
* Python Debugger предоставляет мощные инструменты отладки Python-кода;
* Code Runner упрощает запуск программы, избавляя программиста от необходимости прописывать команды в терминал.
* Draw.io Integration позволяет составлять схемы и диаграммы в среде программирования

Анализ и уточнение требований к программному продукту

* 1. Выбор методов и разработка основных алгоритмов решения задачи

При проектировании программного продукта были разработаны следующие схемы:

* диаграмма потоков данных;
* диаграмма «сущность-связь»;
* алгоритм работы приложения.
* Диаграмма прицедентов

Диаграмма потоков данных (DFD, Data Flow Diagram) — это инструмент, который используется для визуализации того, как данные перемещаются и обрабатываются в системе. Ее цель — показать, как информация поступает, как она преобразуется и как она выходит из системы или проходит через различные ее части. На DFD изображаются все основные элементы, которые взаимодействуют с данными, такие как источники данных, процессы, хранилища и потоки данных. Диаграмма потоков данных для проекта указана на рисунке 3.1:

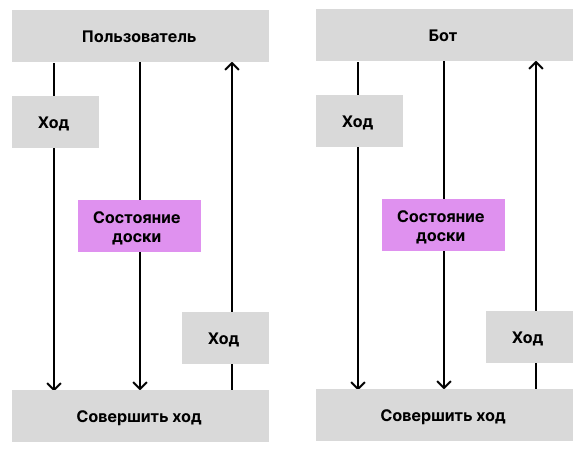


Рисунок 3.1 – Диаграмма потоков данных

На диаграмме изображен общий принцип взаимодействия пользователя с шахматным ботом. Пользователь делает ход, и получившаяся позиция передается боту. Со стороны бота возвращается ход и измененная шахматная позиция.

ERD (Entity-Relationship Diagram) – это диаграмма, используемая для моделирования структуры базы данных. Она помогает визуализировать сущности (таблицы) и их взаимосвязи, что облегчает понимание и проектирование базы данных. Диаграмма «сущность-связь» показана   
на рисунках 3.2, 3.3. В проекте используется несколько баз данных в 1 и 3 нормативных формах. Это позволяет

* использовать базу данных в 1 нормативной форме в качестве резервной копии;
* использовать более простой код для заполнения базы данных, так как для этого используется база данных в 1 нормативной форме.

Для перевода базы данных из 1 нормативной формы в 3 реализован  
код миграции.

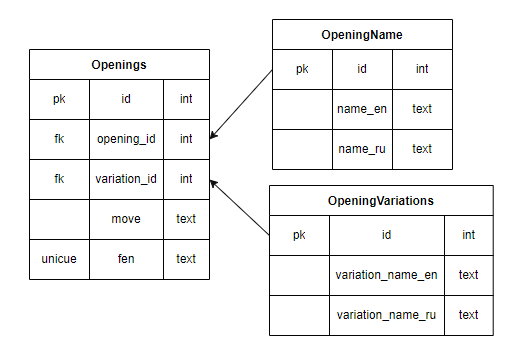


Рисунок 3.2 – Диаграмма «сущность-связь» для базы данных в 3 нормативной форме



Рисунок 3.3 – Диаграмма «сущность-связь» для базы данных в 1 нормативной форме

Алгоритм работы приложения – это последовательность шагов, которые выполняются для достижения определенной цели или функции приложения.

Алгоритм работы приложения представлен в приложении А. В контексте работы шахматного приложения алгоритм работы представляет собой основной цикл, не прекращающийся до окончания партии.

Разработка структурной схемы программного продукта

* 1. Описание используемых процедур и библиотечных функций

Библиотеки, используемые в программном решении представлены в  
таблице 4.1

Таблица 4.1 – Список библиотек, используемых в программе

|  |  |
| --- | --- |
| Название | Назначение |
| Pygame | библиотека для разработки компьютерных игр. Она предоставляет инструменты для работы с графикой, звуком и вводом-выводом. Необходима в проекте для реализации отображения доски и фигур, пользовательского интерфейса и декоративных звуков в процессе партии |
| Chess | для работы с шахматами, которая предоставляет мощный инструментарий для анализа и управления шахматными партиями. Необходима для реализации комплексных шахмат, возможности интеграции сторонних шахматных ботов |
| Sqlite3 | для доступа к СУБД SQLite с помощью python |
| Os | для работы с файлами, папками, путями к файлам. |

Названия функций, используемых в решении и их назначение представлены в таблице 4.2.

Таблица 4.1 – Список функций, используемых в программе

|  |  |
| --- | --- |
| Название | Назначение |
| main.getWindowSize() | Расчет размеров окна приложения в зависимости от размеров экрана пользователя. |
| main. loadSetting() | Получение настроек приложения из соответствующего файла |
| main() | Запуск приложения |
| src.game.choose\_promotoion() | Открытие окна для выбора, в какую фигуру превратится пешка |
| src.game.choose\_color() | Открытие окна для выбора цвета фигур перед началом партии |
| src.game. handle\_mouse\_button\_down() | Обработка нажатия кнопки мыши по модели игрового поля |
| src.game.handle\_mouse\_button\_up() | Обработка отпускания кнопки мыши от модели игрового поля |
| src.game.execute\_move() | Выполнение хода, полученного от пользователя путем перетаскивания фигуры по модели игрового поля |
| src.game.update\_opening() | Обновление названия позиции, отображенной на модели доски |
| src.game.display\_opeing\_name() | Отображение названия позиции, отображенной на модели доски |
| src.game.undo\_move() | Отмена последнего хода в партии по нажатии соответствующей клавиши с клавиатуры |
| src.game.redo\_move() | Возврат последнего отмененного хода в партии по нажатии соответствующей клавиши с клавиатуры |
| src.bot.chessBotWrapper.find\_best\_move() | Получение лучшего хода по оценке искусственного оппонента |
| src.windows.board.load\_images() | Загрузка изображений для отображения их на месте соответствующих фигур на модели доски |
| src.windows.board.draw\_board() | Отрисовка модели доски на экране пользователя с учетом соответствующих настроек и условностей |
| src.windows.color\_window.open() | Отображение страницы выбора цвета фигур с учетом соответствующих настроек |
| src.windows.promotion\_window.open() | Отображение страницы выбора фигруры, в которую превратится пешка в соответствии с правилами игры и учетом соответствующих настроек |
| src.ai.chessbot\_find\_best\_move() | Возвращает лучший ход по оценке искусственного оппонента |
| src.ai.chessbot.minimax() | Реализация алгоритма минимакс для расчета ходов искусственным оппонентом |
| src.ai.chessbot.store\_transponition() | Сохранение позиции в позиции транспозиции для ускорения расчетов искусственного оппонента |
| src.ai.chessbot.sort\_moves() | Сортировка ходов искусственным оппонентом по приоритету. |
| src.ai.chessbot.thretens\_mate() | Проверка, создает ли ход угрозу мата |
| src.ai.chessbot.evaluate\_opening\_principles() | Оценка позиции на соответствие дебютным принципам |
| src.ai.chessbot.evaluate\_board() | Оценка позиции по нескольким параметрам |
| src.ai.chessbot.spatial\_advantage() | Оценка пространственного преимущества в партии |
| src.ai.chessbot.control\_of\_open\_lines() | Оценка контроля над открытыми линиями |
| src.ai.chessbot.calculate\_isolated\_pawns | Оценка пешек не имеющих пешек на соседних вертикалях |
| src.ai.chessbot.calculate\_doubled\_pawns | Оценка 2 и более пешек на 1 вертикали |
| src.ai.chessbot.calculate\_pawn\_islands | Вычисление «пешечных островов». Пешечный остров – несколько пешек на соседних вертикалях |
| src.ai.chessbot.center\_control | Вычисление контроля над центром в позиции. |
| src.db.new.database.create\_tables | Создание таблиц базы данных в соответствующем файле |
| src.db.new.database.execute\_query | Вспомогательный метод для выполнения SQL запросов для базы данных |
| src.db.new.database.create\_opening | Создание записи о шахматном дебюте в базе данных |
| src.db.new.database.read\_openings | Получение всех записей о дебютах из базы данных |
| src.db.new.database.update\_opening | Обновление записи о дебюте в базе данных |
| src.db.new.database.delete\_opening | Удаление записи о дебюте в базе данных |
| src.db.new.database. get\_full\_opening\_name\_by\_fen | Получение полного названия дебюта по позиции, записанной в формате FEN |
| src.db.new.database.close | Закрытие соединения к базе данных |
| src.db.migration.migrate\_data | Миграция данных из 1 нормативной формы в 3 нормативную форму |
| src.db.migration.main | Запуск кода миграции базы данных из 1 нормативной формы в 3 |
| src.db.old.database.create\_tables | Создание таблиц базы данных в 1 нормативной форме в соответствующем файле |
| src.db.old.database.execute\_query | Вспомогательный метод для выполнения SQL запросов для базы данных в 1 нормативной форме |
| src.db.old.database.create\_opening | Создание записи о шахматном дебюте в базе данных в 1 нормативной форме |
| src.db.old.database.read\_openings | Получение всех записей о дебютах из базы данных в 1 нормативной форме |
| src.db.old.database.update\_opening | Обновление записи о дебюте в базе данных в 1 нормативной форме |
| src.db.old.database.delete\_opening | Удаление записи о дебюте в базе данных в 1 нормативной форме |
| src.db.old.database. get\_opening\_name\_by\_fen | Получение полного названия дебюта по позиции, записанной в формате FEN |

* 1. Спецификация программы

Файлы программного продукта представлены в таблице 4.3.

|  |  |
| --- | --- |
| Наименование | Назначение |
| assets | Папка для хранения исходных файлов продукта, такие как изображения фигур, декоративные звуки и.т.д |
| assets/fonts | Папка для хранения шрифтов для программного продукта |
| assets/images | Папка для хранения изображений для программного продукта |
| assets/images/docs-images | Папка для хранения изображений для файла README.md |
| assets/images/icons | Папка для хранения иконок для программного продукта |
| assets/images/pieces | Папка для хранения изображений фигур для программного продукта |
| assets/sounds | Папка для хранения звуковых файлов для программного продукта |
| data | Папка для хранения баз данных |
| docs | Папка для хранения документов по программному продукту, таких как: диаграммы, пояснительная записка и.т.д. |
| DEPENDENCIES.txt | Список зависимостей для программного продукта |
| LIECENSE | Текст условий распространения программного продукта |
| README.md | Файл, рекомендуемый к прочтению пользователем перед использованием программного продукта |
| User\_settings.txt | Файл, хранящий пользовательские настройки программного продукта, такие как: Размер окна, настройки искусственного оппонента, язык интерфейса и.т.д |
| main.py | Запускаемый файл для программного продукта |
| src | Папка, хранящая код программного продукта |
| src/ai | Папка, хранящая код искусственного оппонента |
| src/db | Папка, хранящая код баз данных |
| src/db/new | Папка, хранящая код базы данных в 3 нормативной форме |
| src/db/old | Папка, хранящая код базы данных в 1 нормативной форме |
| src/db/migration.py | Файл запуска миграции базы данных из 1 в 3 нормативной форме |
| src/windows | Папка, хранящая код окон программного продукта |

Проектирование интерфейса пользователя

* 1. Разработка форм ввода-вывода информации

Дизайн проекта был разработан в Дизайн проекта был разработан в программе Figma, где были определены основные экраны:

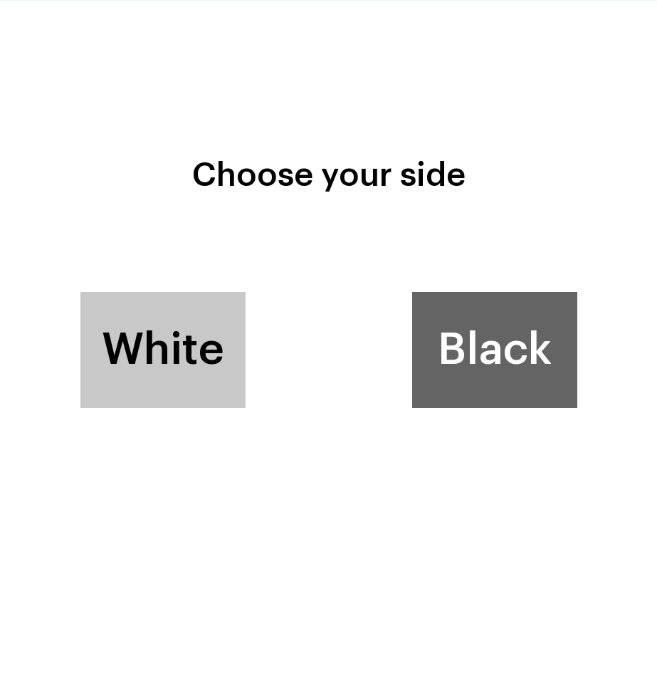
* экран выбора цвета фигур;
* экран партии и вспомогательной информации.

Рисунок 5.1 – Экран для выбора цвета фигур на языках программного продукта

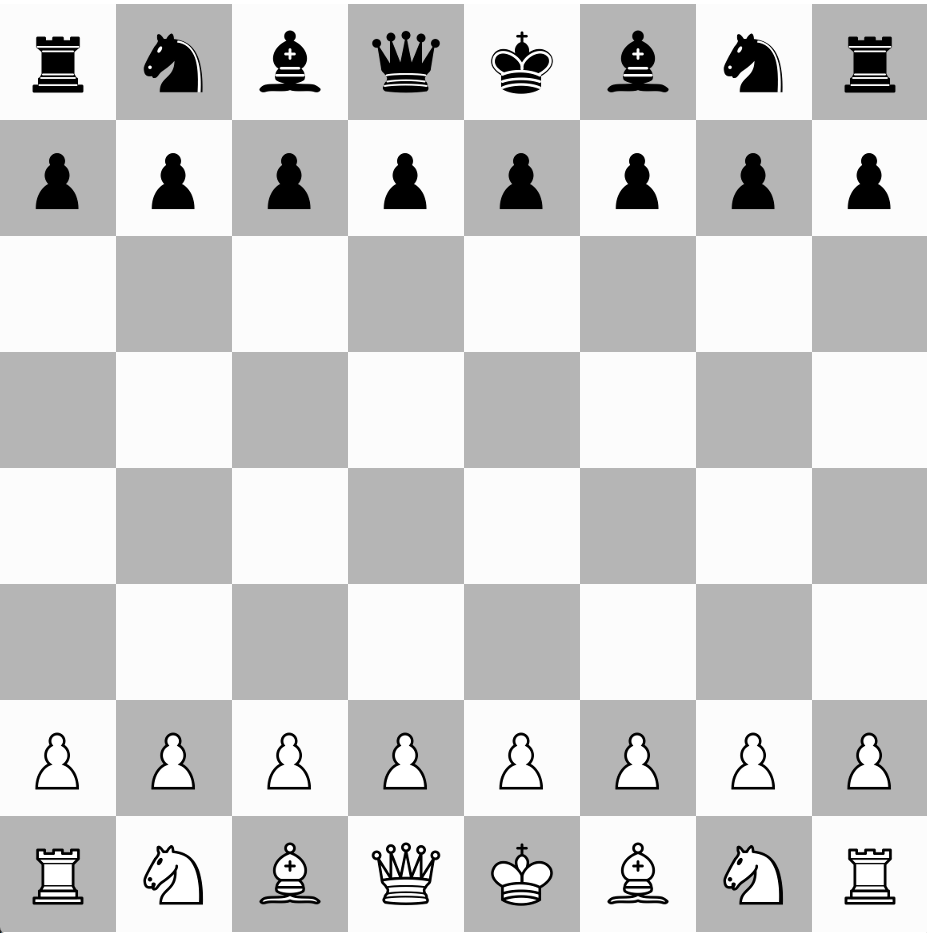
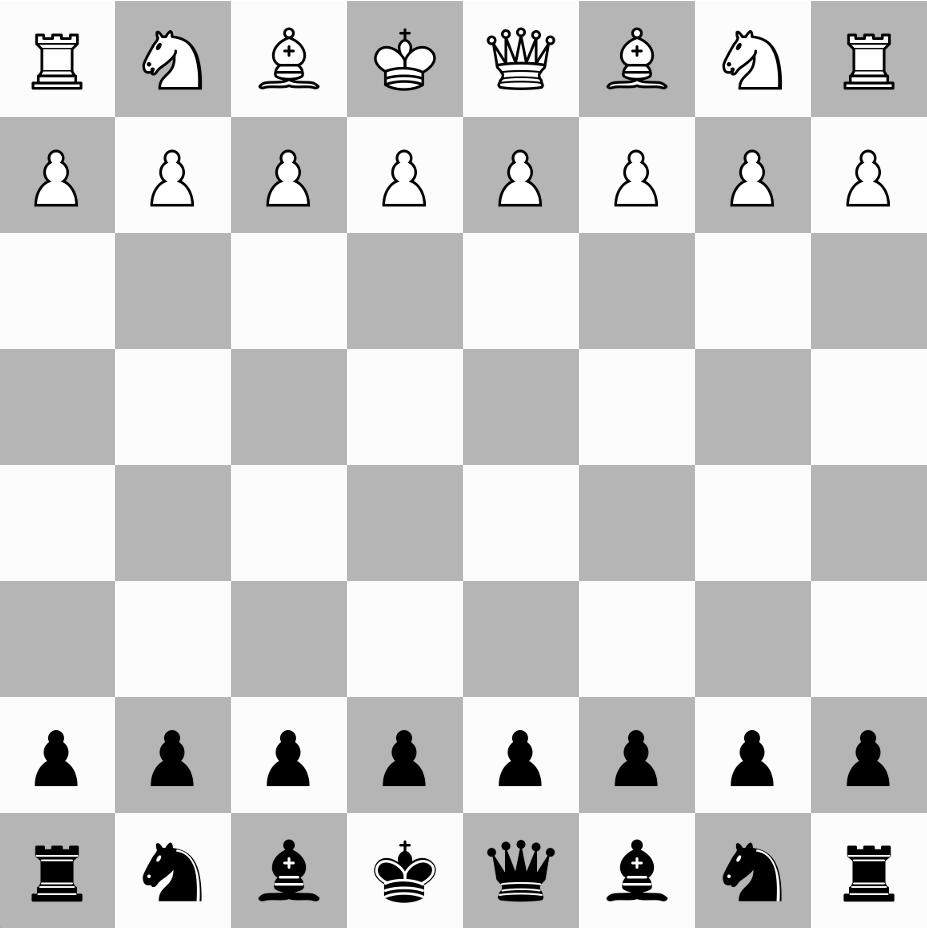
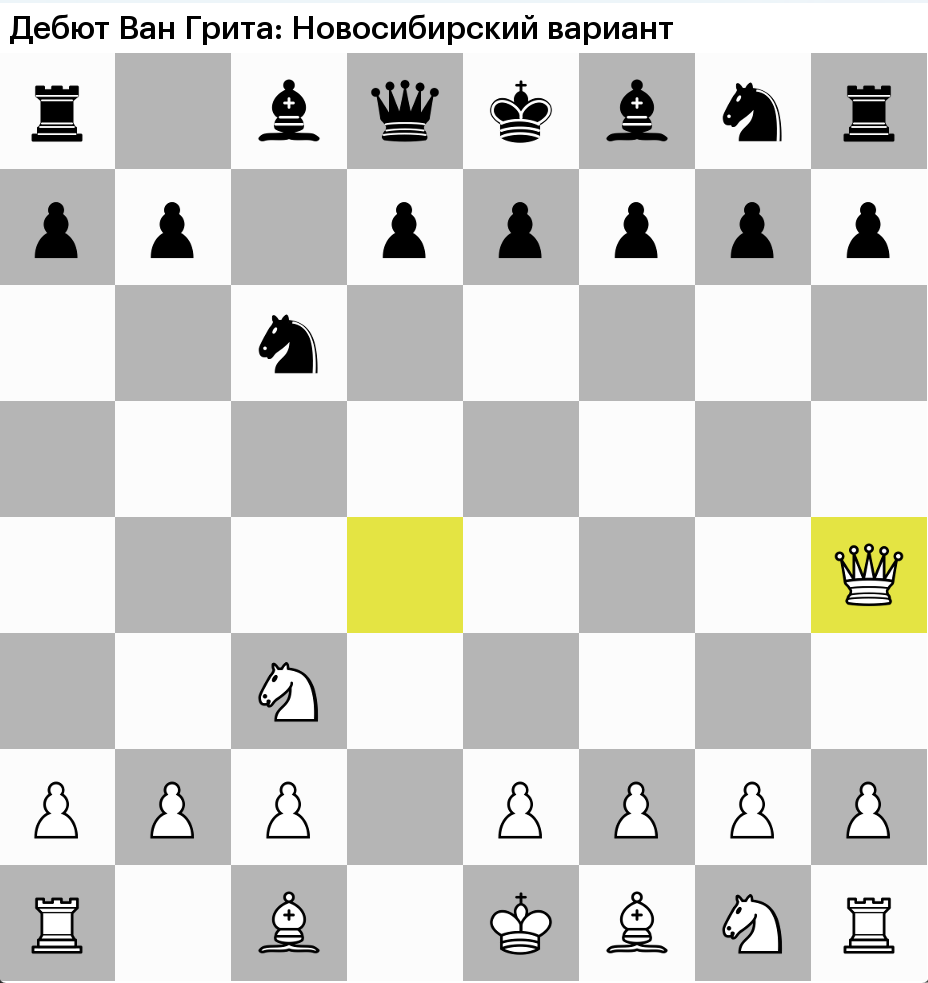
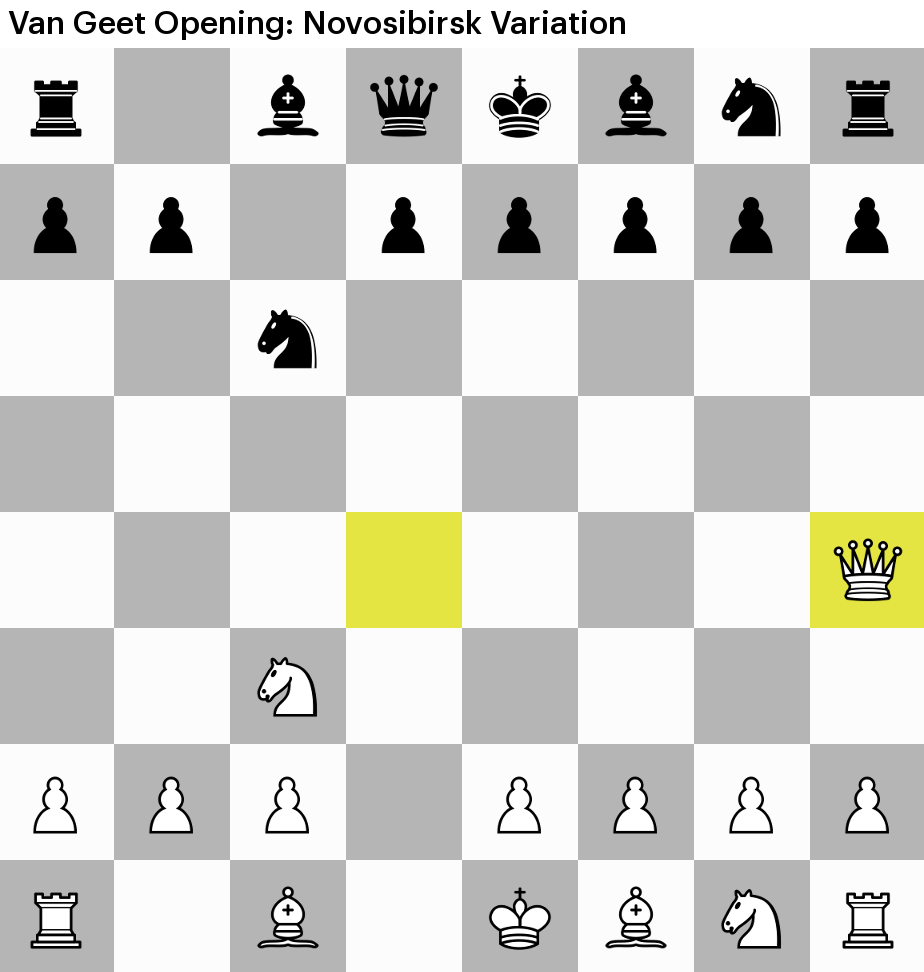
Рисунок 5.2 – Экран начала партии за белые и черные фигуры

Рисунок 5.3 – Экран ведомой партии на примере Новосибирского варианта дебюта Ван Грита на языках программного продута

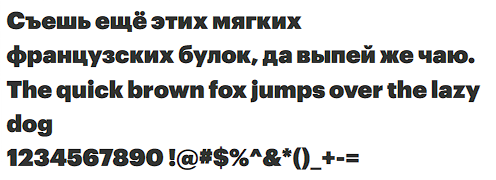
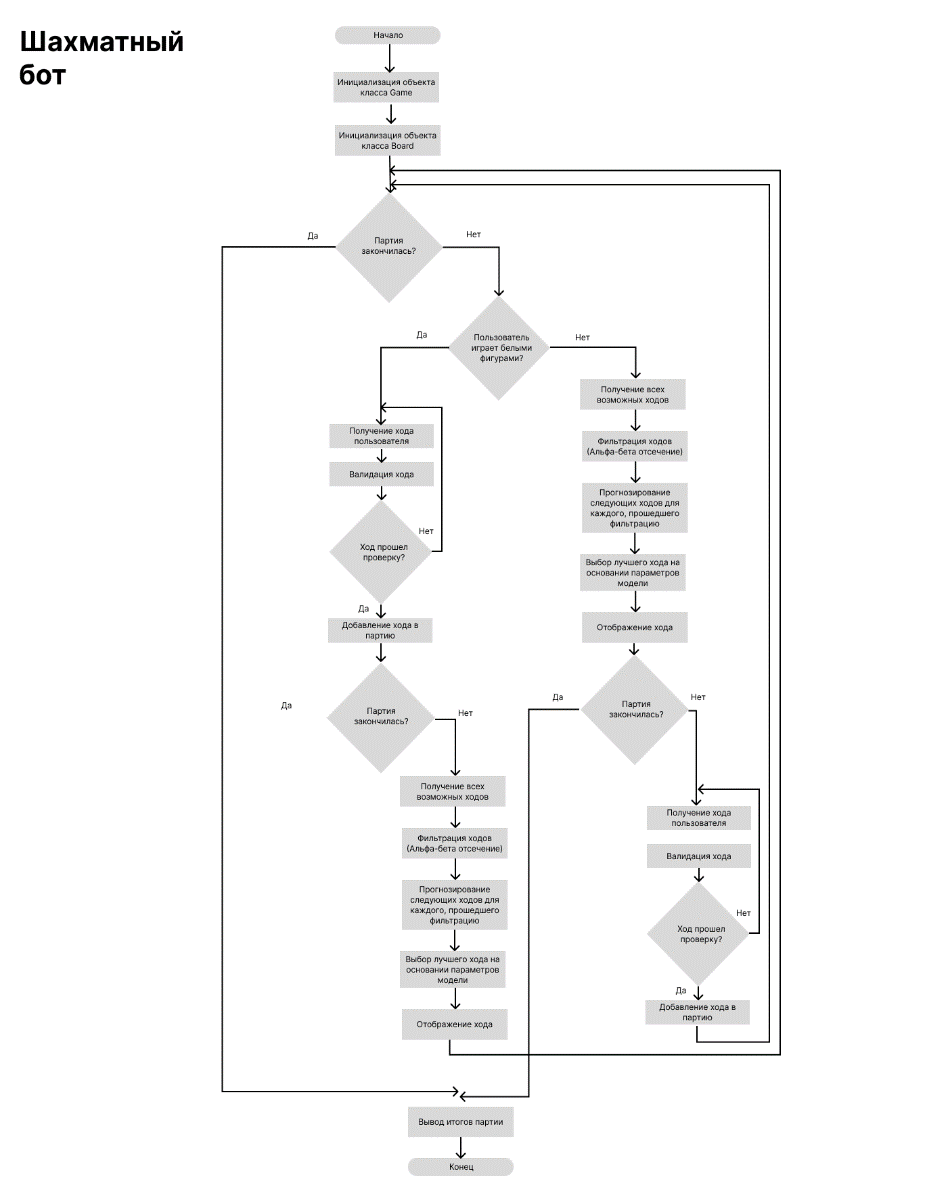
В проекте используется шрифт GraphikLCG, который изображенн на рисунке 5.4. Он имеет декоративное значение, поддерживает все языки, используемые в программном продукте, а также используется в проектах, рассчитанных на широкую аудиторию, приметы приведены на рисунках Б.1 – Б.4.

Рисунок 5.4 – Шрифт, используемый в проекте

Приложение А

(рекомендуемое)

Алгоритм работы приложения



Приложение Б

(информативное)

Примеры проектов, использующих шрифт, используемый в проекте

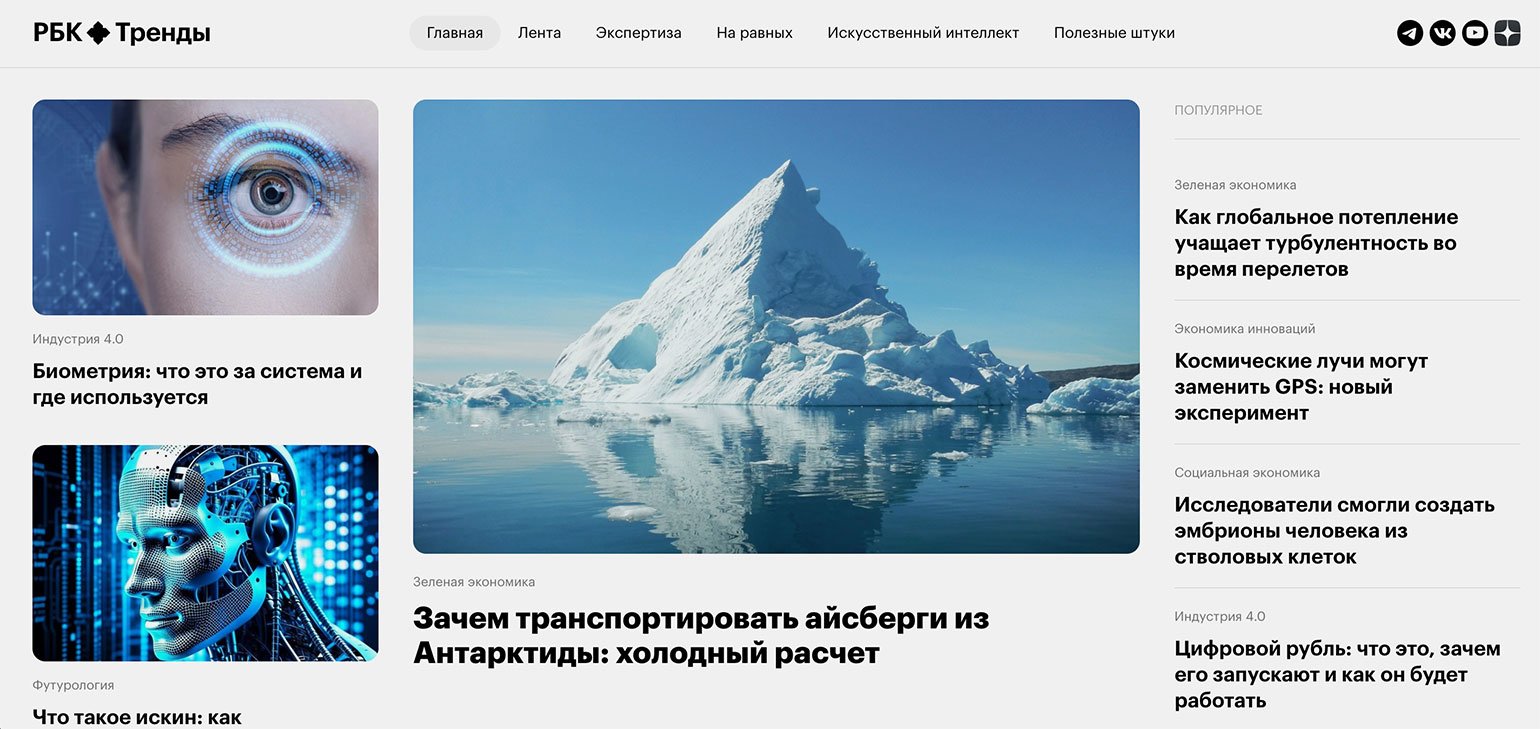


Рисунок Б.1 – медиахолдинг РБК.



Рисунок Б.2 – Сайт Большого театра

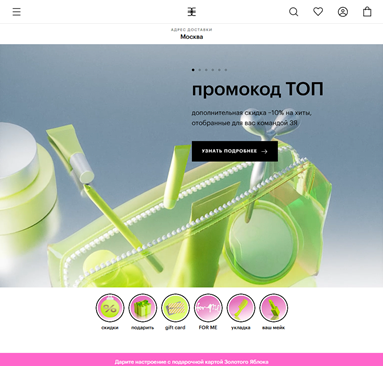


Рисунок Б.3 – Сайт магазина для красоты, жизни, дома «Золотое Яблоко»

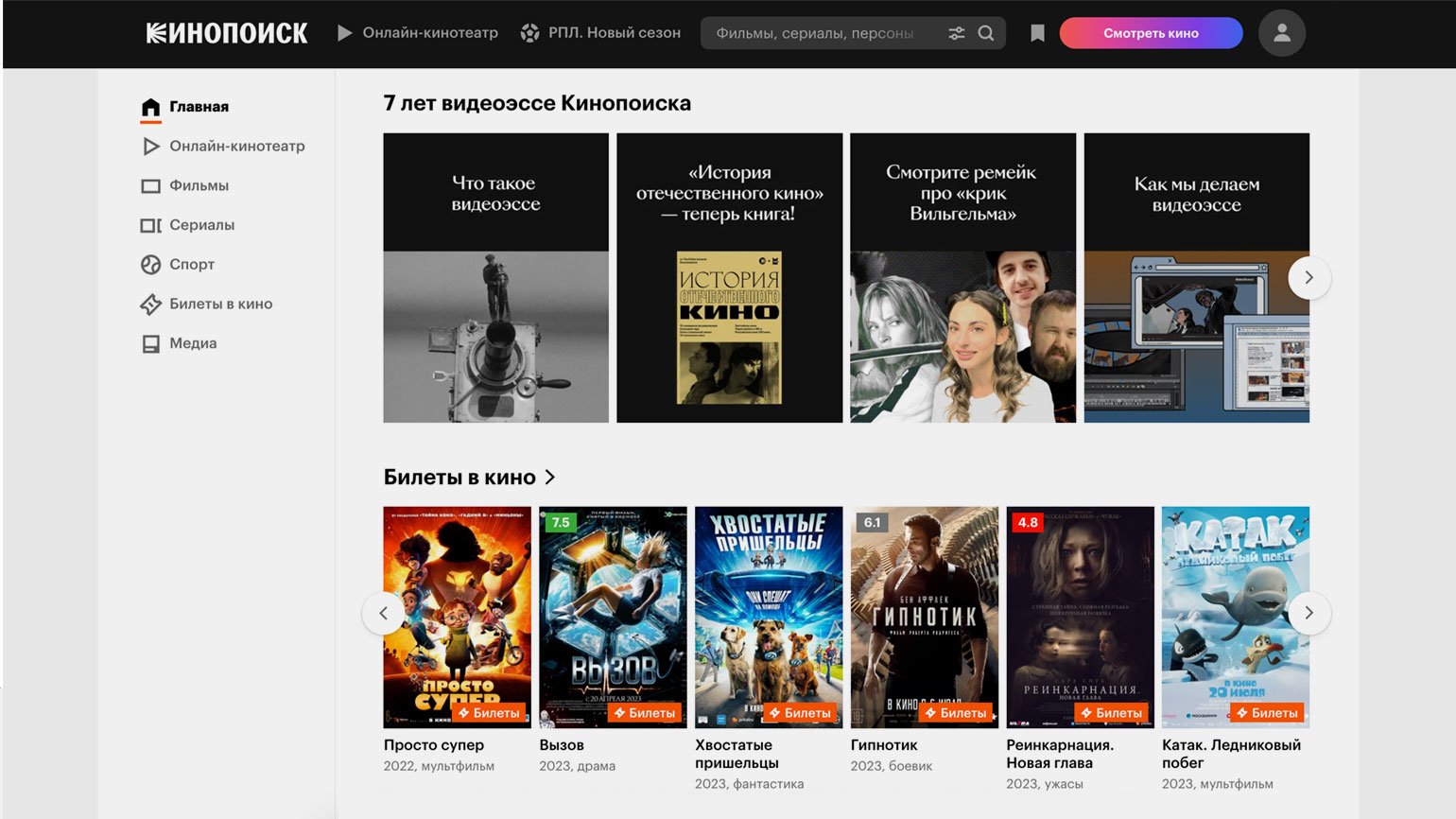


Рисунок Б.4 – Онлайн кинотеатр «Кинопоиск»