МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Вятский государственный университет»**

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Допущено к защите

Руководитель проекта

\_\_\_\_\_\_\_\_/Исупов К.С./

(подпись) (Ф.И.О)

«\_\_»\_\_\_\_\_\_\_\_\_\_202\_г.

Разработка приложения «Аналитика сезона NBA 2019-2020»

Пояснительная записка курсового проекта по дисциплине

«Комплекс знаний бакалавра в области программного и аппаратного обеспечения вычислительной техники»

ТПЖА.09.03.01.723 ПЗ

Разработал студент группы ИВТ-23 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Кудяшев Я.Ю./

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Исупов К.С./

Консультант \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Коржавина А.С./

Работа защищена с оценкой «\_\_\_\_\_\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (дата)

Члены комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_/

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись)

Киров 2020

Реферат

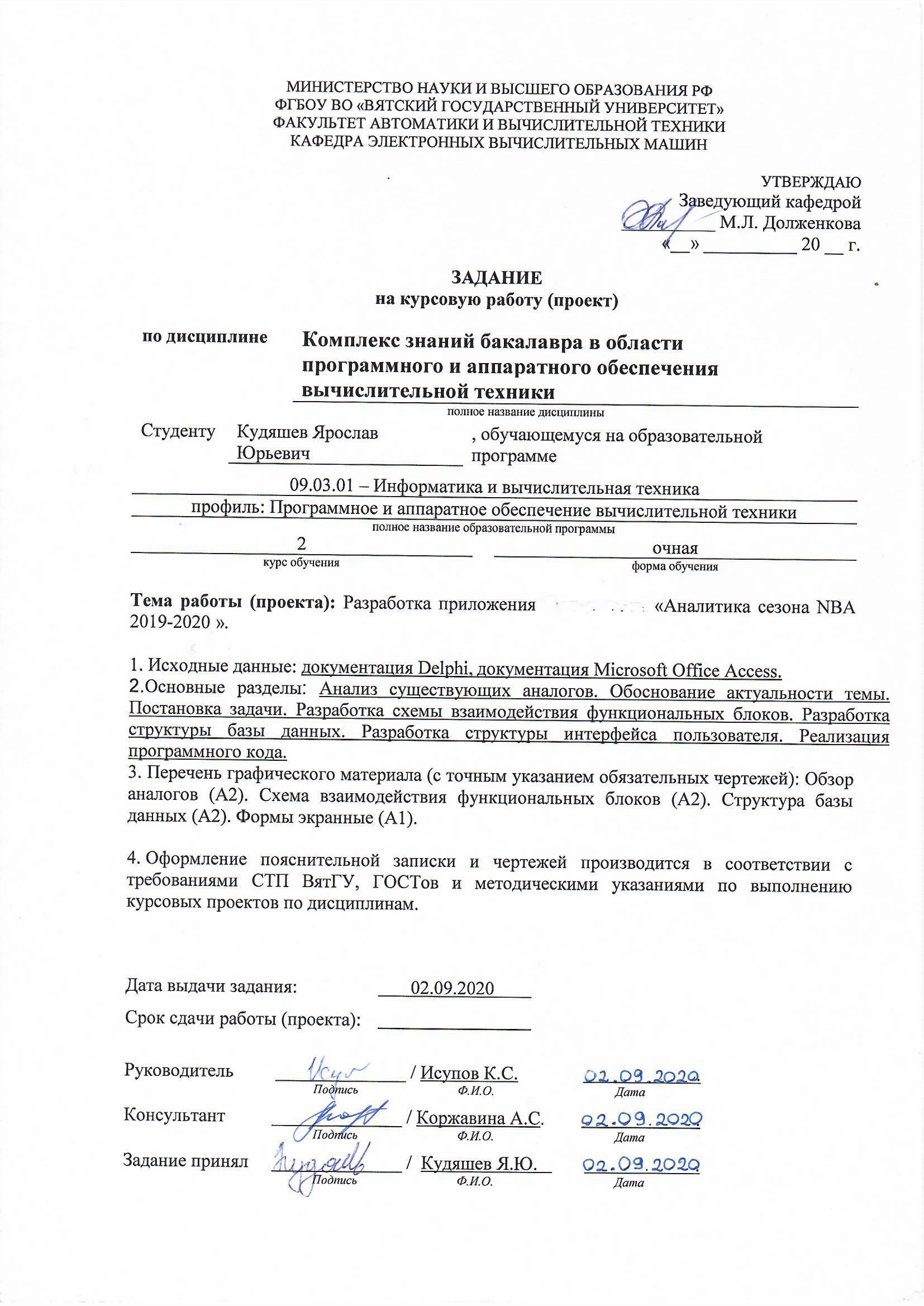
Кудяшев Я.Ю.. Разработка приложения «Аналитика сезона NBA 2019-2020». ТПЖА.090301.021 ПЗ: Курс. проект / ВятГУ, каф. ЭВМ; рук. Исупов К.С.. - Киров, 2020. – ПЗ 34с, 12 рис., 4 прил.

Объект курсового проекта – аналитические показатели в мире спорта.

Предмет курсового проекта – аналитика игрового сезона в области национальной баскетбольной ассоциации.

Цель курсового проекта – облегчить работу аналитикам при исследовании статистики сезона NBA.

Результатом выполнения курсового проекта является приложение для аналитики сезона НБА 2019-2020.



Содержание

[Введение 4](#_Toc60005385)

[1. Обзор предметной области 5](#_Toc60005386)

[2. Разработка расширенного технического задания 10](#_Toc60005387)

[2.1. Основание разработки 10](#_Toc60005388)

[2.2. Назначение 10](#_Toc60005389)

[2.3. Требования к программе 10](#_Toc60005390)

[2.3.1. Требования к функциональным характеристикам 10](#_Toc60005391)

[2.3.2. Требования к информационной и программной совместимости 11](#_Toc60005392)

[2.4. Требования к программной документации 11](#_Toc60005393)

[2.5. Требования к технической документации 11](#_Toc60005394)

[3. Разработка алгоритмов функционирования 12](#_Toc60005395)

[3.1. Основной алгоритм 12](#_Toc60005396)

[3.2. Алгоритм работы с базой данных 14](#_Toc60005397)

[3.3. Алгоритм заполнения базы данных 14](#_Toc60005398)

[4. Реализация программного продукта 16](#_Toc60005399)

[4.1. Выбор языка программирования и среды разработки 16](#_Toc60005400)

[4.2. Тестирование 17](#_Toc60005401)

[4.3. Руководство пользователя 19](#_Toc60005403)

[4.4. Руководство программиста 20](#_Toc60005404)

[Заключение 21](#_Toc60005405)

[Приложение А 22](#_Toc60005406)

[Приложение Б 23](#_Toc60005407)

[Приложение В 24](#_Toc60005409)

[Приложение Г 26](#_Toc60005410)

Введение

В настоящее время представить работу спортивного аналитика без каких-либо сподручных средств практически невозможно. Человек способен наблюдать за ходом спортивного матча и мысленно анализировать действия игроков: выделять слабые и сильные стороны как в защите, так и в нападении.

Конечно, обычный зритель, наблюдающий за игрой, вряд ли сможет запомнить всех игроков. В баскетболе, как правило, люди привыкли в первую очередь акцентировать внимание на количество очков, набираемых игроком, что является большой ошибкой. Они забывают про такие показатели, как эффективность игрока, его защитный рейтинг и остальную статистику, помимо очков. Лишь благодаря тщательной аналитике можно сказать, кто является наиболее ценным игроком команды и выделить тех ролевых игроков, которые оказывают огромное влияние на игру, набирая меньшее количество очков.

Решением этой проблемы может послужить база данных игроков NBA с их подробной статистикой. Благодаря ей можно проанализировать большинство аспектов игры какого-либо игрока. В базе отображены такие важные показатели, как очки, подборы, передачи, перехваты, блокшоты и другое. Таким образом, при помощи базы данных можно проанализировать практически все стороны игры того или иного игрока. Цифры игроков говорят о многом и дают представление об их слабых и сильных сторонах.

1. Обзор предметной области

Сегодняшний мир сложно представить без спорта и спортивных мероприятий. Спортивная индустрия насчитывает огромное количество различных видов спорта, таких как футбол, баскетбол, хоккей и другие. Данный проект посвящён баскетболу, а именно – сезону НБА 2019-2020. НБА – самая популярная мужская баскетбольная лига, расположенная в Северной Америке. Она бы не пользовалась такой популярностью, если бы не «звёздные» игроки и их достижения. Подсчётом статистики и дальнейшим прогнозированием будущего того или иного игрока занимаются аналитики. Однако было бы довольно сложно следить за игрой каждого из членов лиги без использования дополнительных средств.

Рассматривались разные способы решения проблемы аналитиков, но выбор пал на создание базы данных игроков НБА 2019-2020. Данное решение является наиболее рациональным с точки зрения полезности. Именно в базе данных отображена важнейшая статистика игроков НБА. Исходя из того, что лига состоит из 30 команд, было предпринято решение: разделить всю базу данных на 30 групп для более подробной аналитики по каждой команде, которая помогла бы выделить явных лидеров в том или ином аспекте игры.

На момент реализации проекта сезон НБА 2019-2020 подошёл к концу, поэтому база данных имеет наиболее актуальные цифры за весь сезон. Каждый игрок имеет 20 показателей, отображающих его эффективность игры в данном сезоне, и 6 полей, в которых отображена информация об их карьере и физических показателях, таких как рост, вест, опыт игры. Это необходимо для того, чтобы у человека, впервые столкнувшегося с каким-либо игроком из лиги, было хотя бы какое-то представление о нём: на какой позиции он выступает и какой стиль игры ему присущ.

Структура базы данных, состоящая из таких показателей как FGM, PTS, GP и т.п., считается как среднее арифметическое за весь сезон. Помимо цифр в программе пользователь сможет увидеть визуальное отображение показателей FG% и 3P% при помощи графиков. Именно эти параметры отвечают за процент попадания игроков с поля и с 3-очковой линии. Данная функция даст ясно понять, кто является «снайпером» в своей команде, и кому доверить право на бросок в решающий момент матча.

Далее представлено подробное описание каждого поля в базе данных. Поля Player, Weight, Height, Team, Position и Exp соответсвтенно, не зависят от сезона и являются личными данными каждого игрока. Они отвечают за имя и фамилию игрока, его вес и рост, а также позицию, опыт игры и команду, за которую он выступает. Вся остальные поля относятся к статистическим показателям игрока за 2019-2020 сезон НБА. Подробнее о каждом из них:

– GP – количесвто сыгранных игроком игр в этом сезоне;

– MPG – среднее количество минут, которое игрок проводил на поле за одну игру;

– PPG – среднее количество очков, набираемое игроков за матч;

– RPG – среднее количество подборов, набираемых игроком за матч;

– APG – среднее количество передач, раздаваемое игроком за матч;

– SPG – среднее количество перехватов, которое делает игрок за матч;

– BPG – среднее количество блокшотов, которое делает игрок за матч;

– TOV – среднее количество потерь, делаемое игроком за матч;

– PF – среднее количество фолов, получаемое игроком за матч;

– ORB – среднее количество подборов в атаке, делаемое игроком за матч;

– DRB – среднее количество подборов в защите, делаемое игроком за матч;

– FGM – среднее количество бросков за игру;

– FGA – среднее количество попыток броска за игру;

– FG% - средний процент попадания с игры;

– 3PM – среднее количество бросков с трёхочковой линии за игру;

– 3PA – среднее количество попыток броска с трёхочковой линии за игру;

– 3P% - средний процент попадания с трёхочковой линии за игру;

– FTM – среднее количество бросков со штрафной линии за игру;

– FTA – среднее количество попыток броска со штрафной линии за игру;

– FT% - средний процент попадания со штрафной линии за игру.

Данная программа имеет свои аналоги, которые можно найти в интернете. К примеру, на официальном сайте НБА nba.com можно найти полную сводку по статистике каждого игрока НБА. На данном сайте также можно найти личную информацию практически обо всех действующих игроках ассоциации. Помимо официального сайта можно найти ещё массу сайтов, где приведена полная статистика как игроков, так и команд. Такие сайты, как cbssports.com [1], realgm.com [2] и nba.com [3] дают возможность сортировать и искать игроков разных команд в нужном ключе. Каждому из них присуще своё оформление статистики игроков. Особенностью этих сайтов является то, что они «не стоят на месте». Каждый день они публикуют новости из лиги и результаты игр, прошедших за день. Статистика команд и игроков тоже обновляется каждый день, что позволяет в полной мере следить за событиями, происходящими вокруг лиги ежедневно. Кроме статистики игроков за текущий сезон, на сайте можно найти статистику за все сезоны НБА, как минимум этого десятилетия. Данная функция планируется в дальнейшей разработке программы. На рисунке 1 и рисунке 2 можно увидеть примеры отображения статистики игроков с аналоговых сайтов.

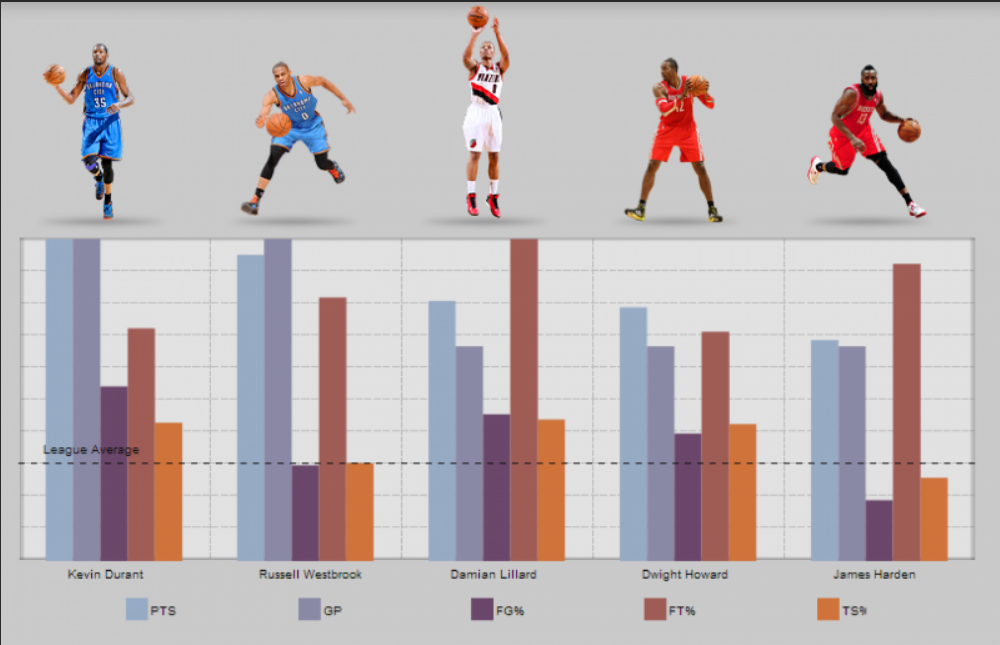


Рисунок 1 – Пример отображения статистики игроков НБА с сайта nba.com



Рисунок 2 – Пример оформления с сайта basketball.realgm.com

Разрабатываемая программа должна соответствовать следующим требованиям:

* + - программа должна отображать статистику для каждой из 30 команд НБА по отдельности;
    - наличие полной базы данных игроков НБА на сезон 2019-2020;
    - актуальная статистика на момент окончания сезона НБА 2019-2020;
    - возможность сортировки по главным показателям игроков;
    - наличие логотипов каждой из команд;
    - возможность поиска по базе данных;
    - отображение графиков эффективности игроков по командам.

2. Разработка расширенного технического задания

В данном разделе представлен процесс разработки расширенного технического задания.

2.1. Основание разработки

Программа разрабатывается на основе учебного плана кафедры «Электронные вычислительные машины» факультета автоматики и вычислительной техники (ФАВТ) Вятского государственного университета.

2.2. Назначение

Основным назначением разрабатываемой программы является облегчение работы спортивным аналитикам НБА.

2.3. Требования к программе

2.3.1. Требования к функциональным характеристикам

Программа должна обеспечивать возможность выполнения следующих функций:

– сортировка информации в базе данных по всем характеристикам игроков должна происходить не только в общей базе данных, но и в каждой команде по-отдельности;

– поиск игроков по базе данных;

– визуализация при помощи графиков эффективности попадания игроков с поля и из-за дуги;

– сортировка игроков по конференциям и дивизионам, к которым относится команда, за которую они выступают;

– сортировка игроков по классу «Rookie» для выявления самого прогрессирующего новичка этого года;

– отображение графиков эффективности игроков для каждой из команд.

2.3.2. Требования к информационной и программной совместимости

Исходные коды программы должны быть написанными на языке программирования высокого уровня и компилироваться на операционных системах семейства Windows и GNU/Linux.

2.4. Требования к программной документации

Состав документации должен включать в себя:

* техническое задание;
* пояснительная записка, содержащая описание разработки.

Программный модуль, написанный в ходе разработки устройства, должен быть самодокументирован, т.е. текст программы должен содержать все необходимые комментарии.

2.5. Требования к технической документации

Разрабатываемое программное обеспечение использует свободную модель распространения и разрабатывается с помощью бесплатных решений.

3. Разработка алгоритмов функционирования

3.1. Основной алгоритм

Основной алгоритм программы довольно прост. При нажатии на логотип любой команды НБА срабатывает событие OnImageClick, которое отвечает за событие на нажатие картинки. После нажатия происходит открытие таблицы с игроками из соответствующей команды. К тожу же, дополнительно происходит выделение выбранной команды. Этого удалось добиться при помощи события OnMouseMove. Помимо алгоритма отображения таблиц с данными в программе присутствует алгоритм построения графиков эффективности. При помощи компонента Delphi TChart удалось построить гистограммы, которые наглядно показывают проценты попадания игроков той или иной команды с поля и из-за дуги. Из двух текстовых файлов считываются проценты попадания каждого игрока, которые затем при помощи цикла отрисовываются на гистограмме для каждой команды. Отображения эффективности для целой лиги не реализовано, т.к. смысла в этом как такового нет. На рисунке 3 и рисунке 4 приведены схемы алгоритмов, реализованных в программе.

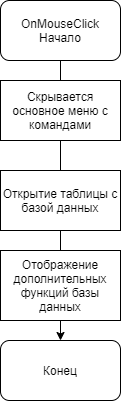


Рисунок 3 – Схема алгоритма отображения базы данных



Рисунок 4 – Схема заполнения графиков

3.2. Алгоритм работы с базой данных

Дальнейший алгоритм для работы с базами данных заключается в сортировках и поиску игроков по ней. При помощи запросов языка SQL удалось довольно просто довести работу программы до автоматизма. Для реализации сортировок и взаимодействия с базой данных были использованы основные команды языка SQL. При помощи оператора SELECT удаётся получить данные из выбранной таблицы. К примеру, комбинация SELECT \* FROM NBA даёт возможность отображения всех данных из таблицы. Ключевое слово WHERE в SELECT нужно для указания условий в запросе. В нём можно задавать следующие условия: сравнение текста, сравнение численных значений или логические операции AND, OR и NOT. Поиск по базе данных тоже осуществляется благодаря операторам SQL языка. Конструкция 'Select \* from NBA where Player like'+help2+' order by Player' позволяет найти любого игрока из базы данных. Оператор LIKE используется в WHERE, чтобы задать шаблон поиска похожего значения. Help2 – переменная, которой присваивается значения при вводе слова в компонент Edit. Подобные SQL выражения используются во всей программе при работе с базой данных. Именно это позволяет реализовать дополнительные возможности при анализе таблицы с данными.

3.3. Алгоритм заполнения базы данных

Пожалуй самым долгим этапом создания программы было заполнение таблиц в базе данных. На официальном сайте можно найти статистику игроков по каждой команде. Для начала пришлось перенести статистику каждой команды в отдельный текстовый файл. При помощи конструктора таблиц в Microsoft Office Access была составлена одна, общая для всех, структура, приведённая. Основную информацию про всех игроков удалось перенести в таблицу при помощи функции экспорта в таблицу из текстового файла. Особенностью таблицы является то, что каждый игрок прикреплён к определённой команде, за которую он выступает. Именно это позволяет с лёгкостью сортировать игроков по командам в самой программе при помощи SQL операторов, выбрав ключевое слово в WHERE. Статистика некоторых игроков отсутствовала на официальном сайте, из-за чего пришлось вручную заполнить около пары десятков игроков базы данных. После данных манипуляций база данных была полностью сформирована. На рисунке 5 можно увидеть структуру базы данных.

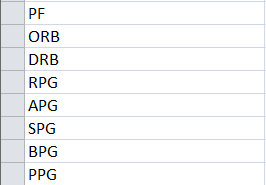


Рисунок 5 – Структура базы данных

4. Реализация программного продукта

4.1. Выбор языка программирования и среды разработки

На основании ранее разработанных этапов проектирования можно приступить к выбору среды проектирования и реализации программного кода. В качестве среды разработки была выбрана Delphi. Выбор пал именно на эту среду разработки, так как Delphi является диалектом языка Object Pascal, который хорошо известен с первого курса. Он является довольно простым в реализации и в изучении. Данная среда программирования позволяется работать с графическим интерфейсом, что даёт возможность красивого оформления курсового проекта. Помимо этого, Delphi имеет все нужные компоненты для работы с базами данных и построения графиков. Построение таблиц и заполнение самой базы данных было решено реализовать при помощи Microsoft Office Access. Данная полнофункциональная система для работы с базами данных любого типа довольна проста в использовании и, что немаловажно, позволяет экспортировать внешние данные из других источников в таблицы. Именно благодаря данной функции было сэкономлено огромное количество времени при заполнении базы данных.

В качестве макета интерфейса было принято решение взять за основу вступительное меню игры NBA2K20. При запуске программы пользователь сразу же может выбрать иконку команды, статистику игроков которой он хочет увидеть. Сбоку от каждой таблицы заинтересованный пользователь может прочитать краткую информацию про команду, которую он выбрал. Кнопки, расположенные на панели сверху, дают возможность построить графики эффективности для каждой команды, в которых будут отображены проценты попадания игроков с поля и из-за дуги, перейти в главное меню и выйти из самой программы. При помощи компонента TMaimMenu были реализованы пункты «Обозначения», где указаны расшифровки аббревиатур, и «Справка», в которой пользователь может ознакомиться с предназначением программы и узнать имя автора данного проекта. На рисунке 6 приведен макет интерфейса реализованной программы.

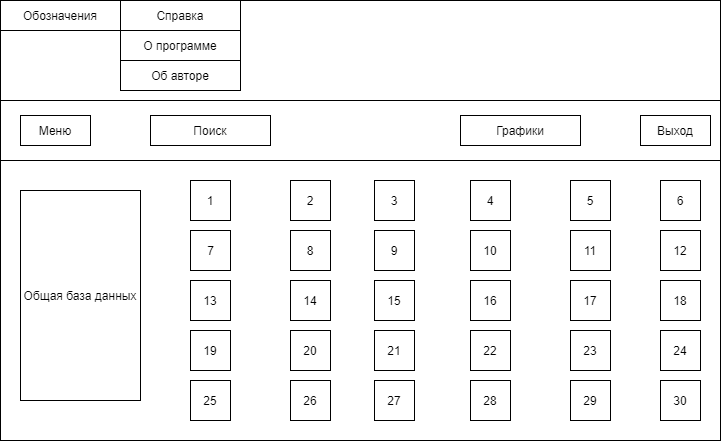


Рисунок 6 – Макет интерфейса программы

При реализации программного кода использовались схемы алгоритмов, представленные в приложении Б.

Исходный код представлен в приложении Г.

4.2. Тестирование

После написания программы было проведено собственноручное тестирование эффективности базы данных, благодаря которому удалось понять, что она действительно несёт пользу и с её помощью можно анализировать общую полезность какого-либо игрока на игровой площадке.

На официальном сайте НБА можно увидеть игроков, которые забрали главные награды сезона 2019-2020. На рисунке 7 приведен список игроков, которые взяли награды НБА сезона 2019-2020.



Рисунок 7 – Реальные награды НБА 2019/2020

Конечно, человек, ничего не понимающий в статистике, вряд ли сможет сделать верный прогноз и в правильном ключе проанализировать базу данных, но человек, который понимает, что значит каждый параметр, должен более менее точно определить: кто возьмёт ту или иную награду.

Так, благодаря программе удалось определить, что с большим шансом награду «Новичок года» заберёт игрок Ja Morant. Он хорошо показывал себя не только набирая очки, но и делая передачи членам своей команды. К тому же, он имеет неплохой процент попадания как из-за дуги, так и с поля. Точно такие же действия можно проделать и для других категорий игроков. Конечно, нельзя со стопроцентной вероятностью сказать, кто возьмёт ту или иную награду, т.к. это уже зависит непосредственно он настоящих аналитиков НБА. Данные, полученные в ходе анализа программной базы данных, представлены на рисунке 8.

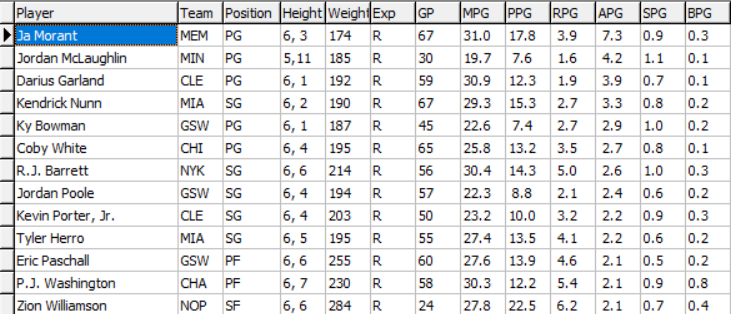


Рисунок 8 – Пример показателей к категории «Новички»

4.3. Руководство пользователя

Программа имеет довольно простой интерфейс. При запуске перед пользователем окажется интерфейс, содержащий логотипы 30 команд НБА, при нажатии на которые откроется таблица со статистикой команды. При нажатии на логотип самой НБА откроется таблица, содержащая статистику всех игроков лиги, с помощью которой можно выделять лидеров уже не только одной команды, но и всей лиги в целом.

Дополнительные кнопки, такие как «Menu» и «Efficiency», позволяют окрыть соответствующие их названию опции программы. Программа предусмотрена и для неподготовленных пользователей: во вкладке «Обозначения» можно увидеть расшифровку кратких обозначений таблицы, что безусловно поможет быстрее освоиться начинающему аналитику.

При помощи компонента RadioGroup можно выбирать различные виды сортировок базы данных, что значительно облегчает просмотр таблицы и её анализ по категориям.

4.4. Руководство программиста

Исходный текст программы самодокументирован, содержит все необходимые комментарии.

Чтобы добавить какие-либо другие сортировки достаточно просто при помощи SQL операторов отсортировать нужные поля в нужном ключе. Благодаря простой комбинации операторов SQL: 'Select \* FROM NBA WHERE Team = '''+'name''' – можно изменять ключ сортировки команд. Name – переменная, в которую записана команда, по которой нужно отсортировать базу данных.

Заключение

В процессе выполнения курсового проекта была разработана программа, позволяющая в полной мере проанализировать статистику игроков НБА за сезон 2019-2020.

В ходе выполнения проекта была рассмотрена предметная область, определены основные задачи, которые необходимо было реализовать, также была выбрана система для работы с базами данных, освоены стандартные SQL запросы и проведены работы над ошибками, возникшими в ходе тестирования.

В дальнейшем планируется увеличить базу данных, заполнив её не только сезоном 2019-2020, но и всеми остальными прошедшими. Это поможет сформировать базу данных содержащую информацию о всех игроках, когда либо игравших в НБА, и выделить тех личностей, которые внесли наибольший вклад в развитие игры. Помимо этого, хотелось бы сделать намного больше различных сортировок, которые будут нужны не только в аналитике, но и просто ради собственного интереса.

На сегодняшний день программа полностью соответствует требованиям, представленным ранее, и с её помощью можно проводить анлитику сезона НБА 2019-2020: выделять лучших и худших игроков на разных участках игрового поля.

Приложение А

(обязательное)

Список используемой литературы

1. cbssports.com [Электронный ресурс]. – Режим доступа:

<https://www.cbssports.com/nba/teams/>

2. realgm.com [Электронный ресурс]. – Режим доступа:

<https://basketball.realgm.com/nba/teams/Washington-Wizards/30/stats/2020/Averages/All/points/All/desc/1/Regular_Season>

3.nba.com [Электронный ресурс].– Режим доступа: <https://www.nba.com/?50>

Приложение Б

(обязательное)

Схемы алгоритмов

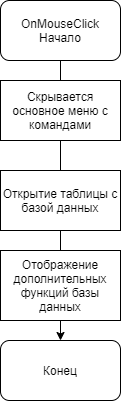


Рисунок 9 – Схемы алгоритмов

Приложение В

(обязательное)

Экранные формы

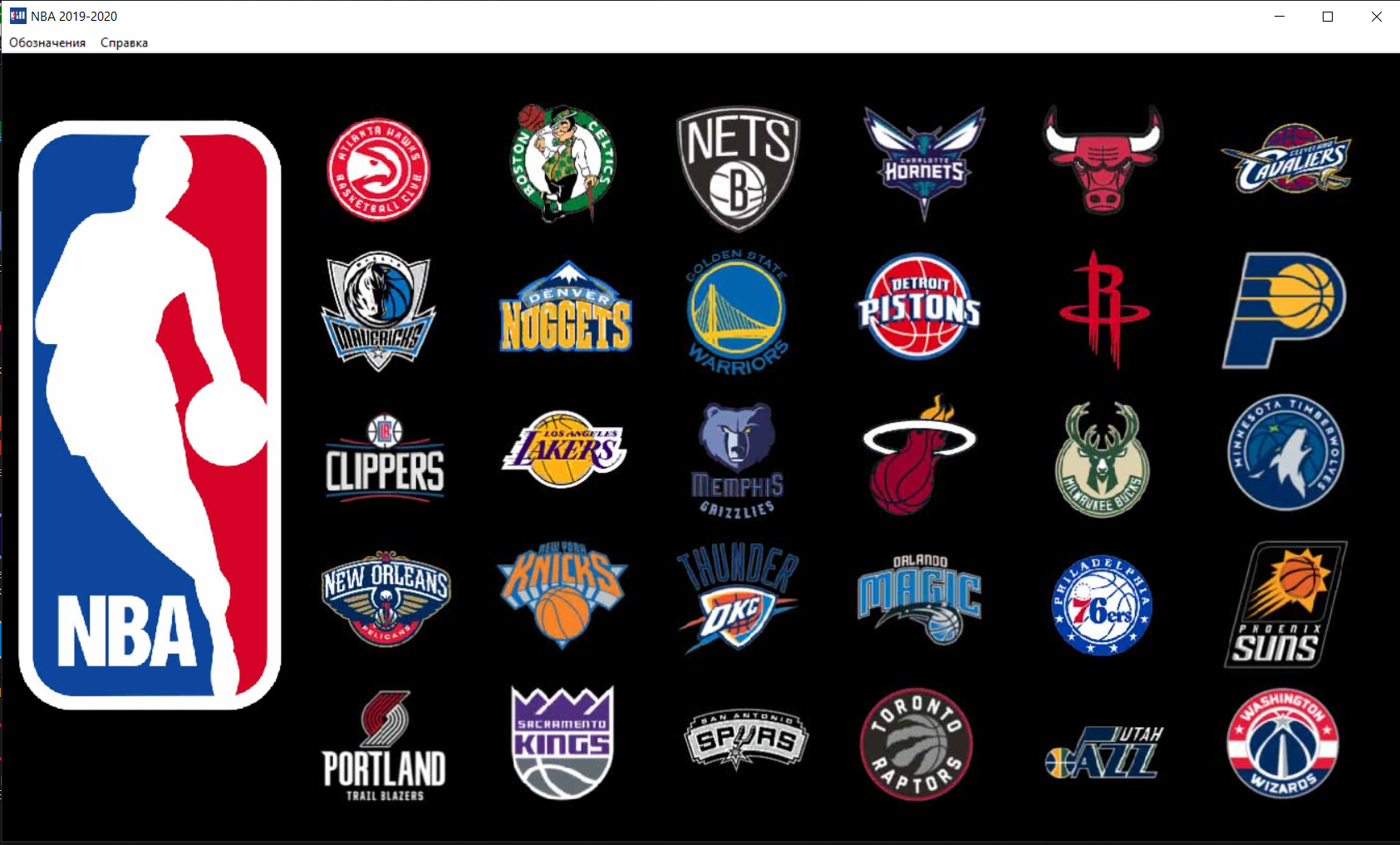


Рисунок 10 – Главное меню программы

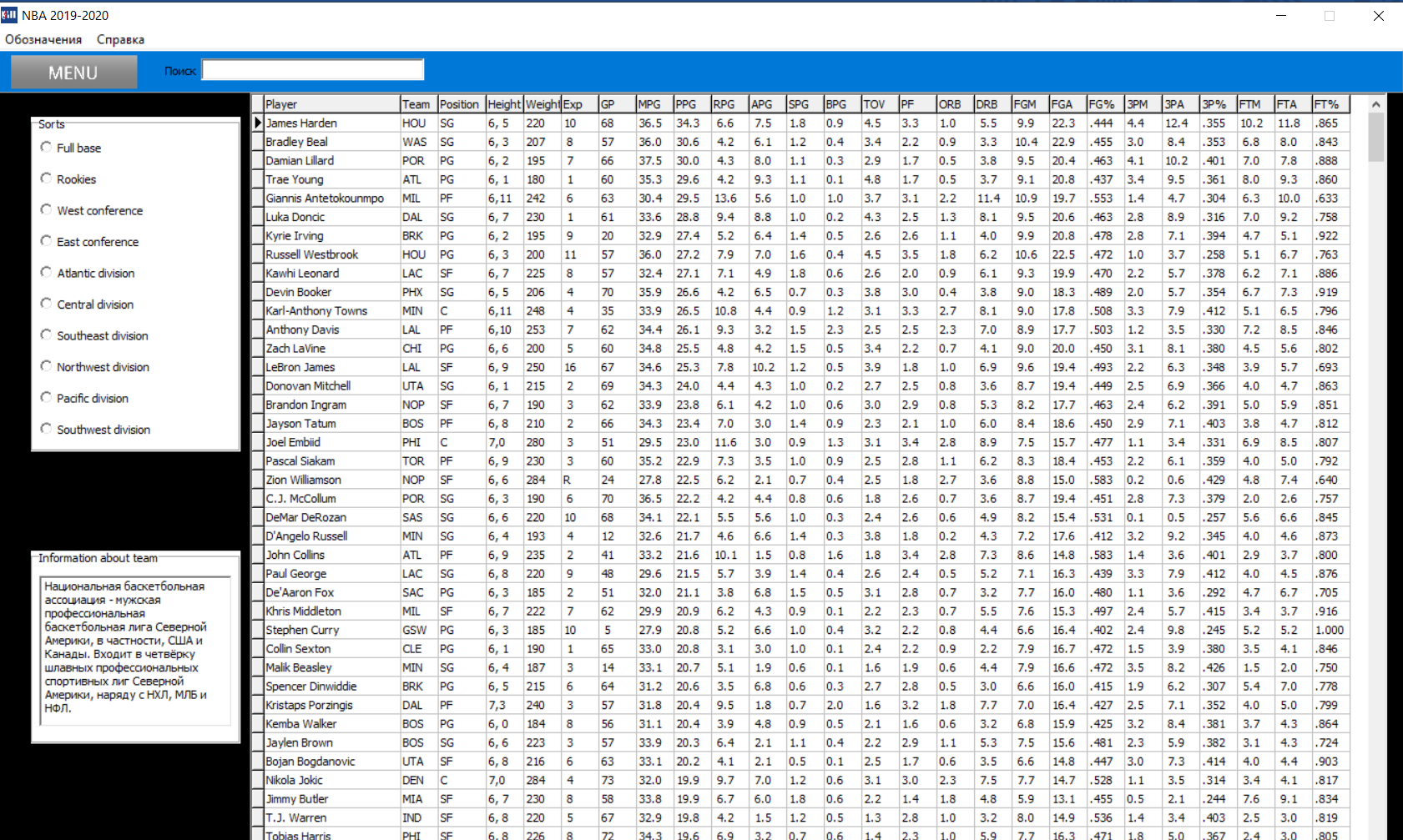


Рисунок 11 – Основная база данных

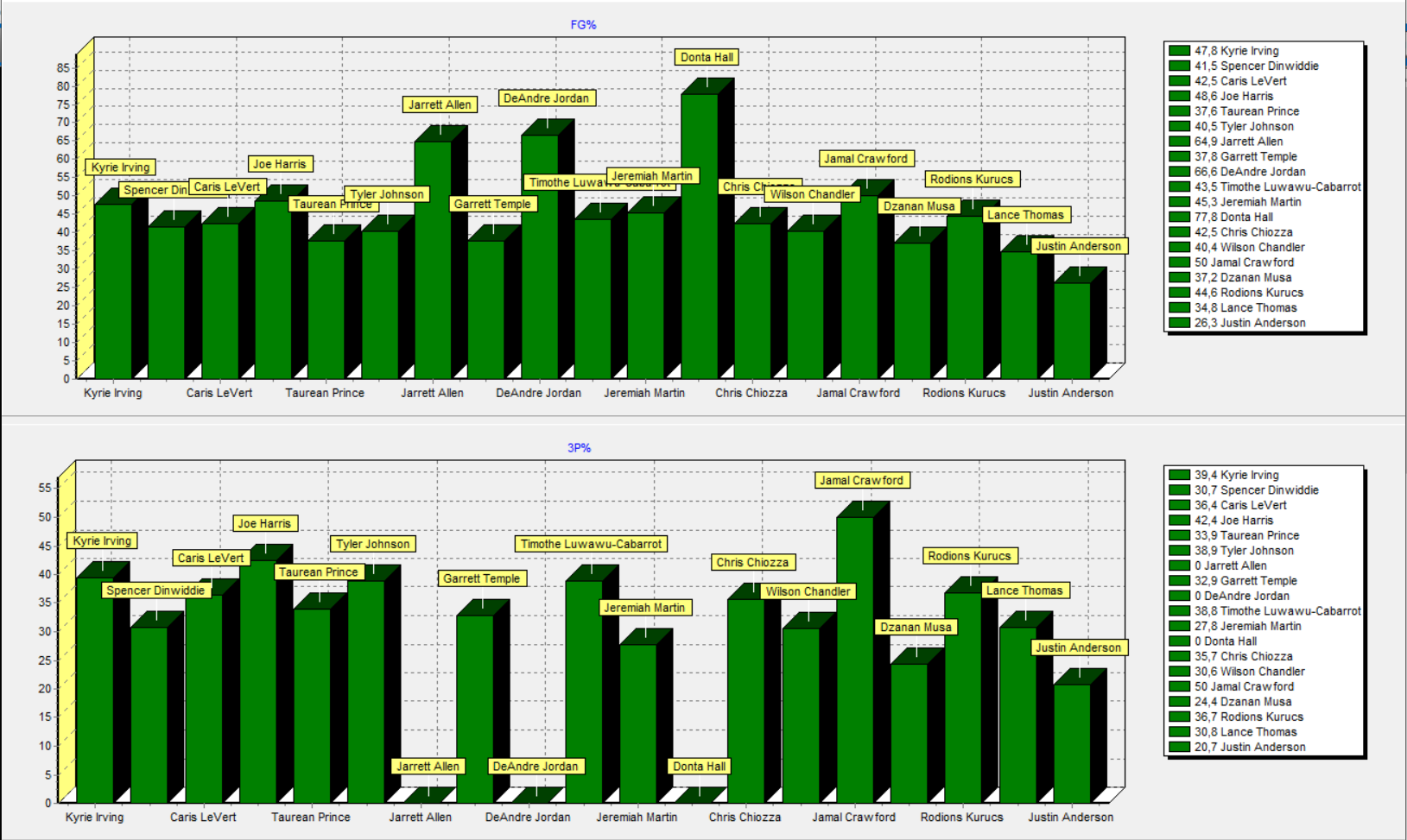


Рисунок 12 – Графики эффективности игроков команды Brooklyn Nets

Приложение Г

(обязательное)

Листинг программы

**procedure** TForm1.edt1Change(Sender: TObject);

**var** help1,help2:string;

kek,kok:string;

**begin**

help1:='%'+Form1.edt1.Text+'%';

help2:=QuotedStr(help1);

//kek:='%'+name+'%';

//kok:=QuotedStr(kek);

**with** Form1.qry1 **do**

**begin**

Close;

SQL.Clear;

SQL.Add('Select \* from NBA where Player like'+help2+' order by Player');

Open;

**end**;

**end**;

**procedure** TForm1.FormActivate(Sender: TObject);

**begin**

Form1.dbgrd1.hide;

Order:=True;

Field:=0;

**end**;

**procedure** TForm1.Clippers(Column: TColumn);

**var** new\_field:Integer;

**begin**

**begin**

new\_field:= Column.Index;

qry1.Close;

qry1.SQL.Clear;

**if** name='NBA' **then**

qry1.SQL.Add('Select \* from NBA ORDER BY ') **else**

**if** name = 'R' **then**

qry1.SQL.Add('Select \* from NBA where Exp = '''+name+'''ORDER BY ') **else**

**if** Name = 'West' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'LAL'+''''+'OR Team = '''+'LAC'+''''

+'OR Team = '''+'HOU'+''''+'OR Team = '''+'DAL'+''''+'OR Team = '''+'POR'

+''''+'OR Team = '''+'MIN'+''''+'OR Team = '''+'OKC'+''''+'OR Team = '''+'UTA'

+''''+'OR Team = '''+'DEN'+''''+'OR Team = '''+'MEM'+''''+'OR Team = '''+'NOP'

+''''+'OR Team = '''+'SAS'+''''+'OR Team = '''+'GSW'+''''+'OR Team = '''+'PHX'+''''+'OR Team = '''+'SAC'''+'ORDER BY ') **else**

**if** Name = 'East' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'BOS'+''''+'OR Team = '''+'NYK'+''''

+'OR Team = '''+'BRK'+''''+'OR Team = '''+'PHL'+''''+'OR Team = '''+'TOR'

+''''+'OR Team = '''+'ATL'+''''+'OR Team = '''+'CHA'+''''+'OR Team = '''+'MIA'

+''''+'OR Team = '''+'ORL'+''''+'OR Team = '''+'WAS'+''''+'OR Team = '''+'CHI'

+''''+'OR Team = '''+'CLE'+''''+'OR Team = '''+'DET'+''''+'OR Team = '''+'IND'+''''+'OR Team = '''+'MIL'''+'ORDER BY ') **else**

**if** Name = 'Atlantic' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'BOS'+''''+'OR Team = '''+'NYK'+''''

+'OR Team = '''+'BRK'+''''+'OR Team = '''+'PHL'+''''+'OR Team = '''+'TOR'''+'ORDER BY ') **else**

**if** Name = 'Central' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'IND'+''''+'OR Team = '''+'CLE'+''''

+'OR Team = '''+'MIL'+''''+'OR Team = '''+'DET'+''''+'OR Team = '''+'CHI'''+'ORDER BY ') **else**

**if** Name = 'Southeast' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'ATL'+''''+'OR Team = '''+'ORL'+''''

+'OR Team = '''+'MIA'+''''+'OR Team = '''+'WAS'+''''+'OR Team = '''+'CHA'''+'ORDER BY ') **else**

**if** Name = 'Northwest' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'UTA'+''''+'OR Team = '''+'MIN'+''''

+'OR Team = '''+'DEN'+''''+'OR Team = '''+'POR'+''''+'OR Team = '''+'OKC'''+'ORDER BY ') **else**

**if** Name = 'Pacific' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'LAC'+''''+'OR Team = '''+'PHX'+''''

+'OR Team = '''+'SAC'+''''+'OR Team = '''+'LAL'+''''+'OR Team = '''+'GSW'''+'ORDER BY ') **else**

**if** Name = 'Southwest' **then**

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'NOP'+''''+'OR Team = '''+'SAS'+''''

+'OR Team = '''+'DAL'+''''+'OR Team = '''+'MEM'+''''+'OR Team = '''+'HOU'''+'ORDER BY ') **else**

qry1.SQL.Add('Select \* from NBA where Team = '''+name+'''ORDER BY ');

**if** new\_field = field **then**

order := **not** order **else**

**begin**

field:=new\_field;

order :=true;

**end**;

**case** field **of**

0: qry1.SQL.Add(' [Player] ');

1: qry1.SQL.Add(' [Team] ');

2: qry1.SQL.Add(' [Position] ');

3: qry1.SQL.Add(' [Height] ');

4: qry1.SQL.Add(' [Weight] ');

5: qry1.SQL.Add(' [Exp] ');

6: qry1.SQL.Add(' [GP] ');

7: qry1.SQL.Add(' [MPG] ');

8: qry1.SQL.Add(' [PPG] ');

9: qry1.SQL.Add(' [RPG] ');

10: qry1.SQL.Add(' [APG] ');

11: qry1.SQL.Add(' [SPG] ');

12: qry1.SQL.Add(' [BPG] ');

//13: qry1.SQL.Add(' [Combined Statistics] ');

13: qry1.SQL.Add(' [TOV] ');

14: qry1.SQL.Add(' [PF] ');

15: qry1.SQL.Add(' [ORB] ');

16: qry1.SQL.Add(' [DRB] ');

17: qry1.SQL.Add(' [FGM] ');

18: qry1.SQL.Add(' [FGA] ');

19: qry1.SQL.Add(' [FG%] ');

20: qry1.SQL.Add(' [3PM] ');

21: qry1.SQL.Add(' [3PA] ');

22: qry1.SQL.Add(' [3P%] ');

23: qry1.SQL.Add(' [FTM] ');

24: qry1.SQL.Add(' [FTA] ');

25: qry1.SQL.Add(' [FT%] ');

**end**;

**if** order **then**

qry1.SQL.Add(' DESC');

qry1.Open;

**end**;

**end**;

**procedure** TForm1.LALakers1Click(Sender: TObject);

**begin**

name:='LAL';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'LAL''');

qry1.Active:=True;

**end**;

**procedure** TForm1.FormCreate(Sender: TObject);

**var** n:Integer;

**begin**

img2.Visible:=False;

Form1.dbgrd1.visible:=false;

Form1.pnl1.Hide;

name:='NBA';

form1.rg4.Visible:=false;

qry1.Open;

SetLength(player,qry1.RecordCount);

qry1.First;

**while not** qry1.Eof **do**

**begin**

**for** i:=0 **to** qry1.RecordCount-1 **do**

**begin**

player[i] := qry1.FieldByName('Player').AsString;

qry1.Next;

**end**;

**end**;

qry1.Open;

SetLength(team,qry1.RecordCount);

qry1.First;

**while not** qry1.Eof **do**

**begin**

**for** i:=0 **to** qry1.RecordCount-1 **do**

**begin**

team[i] := qry1.FieldByName('Team').AsString;

qry1.Next;

**end**;

**end**;

AssignFile(f,'FG%.txt');

Reset(f);

n:=0;

**while not** eof(f) **do**

**begin**

inc(n);

read(f,FG[n]);

FG[n]:=FG[n]\*100;

**end**;

closefile(f);

AssignFile(f,'3P%.txt');

Reset(f);

n:=0;

**while not** eof(f) **do**

**begin**

inc(n);

read(f,P[n]);

P[n]:=P[n]\*100;

**end**;

closefile(f);

**end**;

**procedure** TForm1.img1MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

**begin**

img2.Visible:=True;

**end**;

**procedure** TForm1.pnl1MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

**begin**

img2.Visible:=False;

**end**;

**procedure** TForm1.img2Click(Sender: TObject);

**begin**

Form1.dbgrd1.hide;

Form1.pnl1.Hide;

form1.rg4.Visible:=false;

Form1.img63.Visible:=true;

Form1.img64.visible:=true;

puk:=0;

**end**;

**procedure** TForm1.e(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

**begin**

Form1.img4.Visible:=True;

//Form1.img63.Visible:=true;

//Form1.img64.visible:=true;

**end**;

**procedure** TForm1.FormMouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

**begin**

Form1.img2.Visible:=False;

Form1.img4.Visible:=false;

Form1.img34.Visible:=false;

Form1.img35.Visible:=false;

Form1.img36.Visible:=false;

Form1.img37.Visible:=false;

Form1.img38.Visible:=false;

Form1.img39.Visible:=false;

Form1.img40.Visible:=false;

Form1.img41.Visible:=false;

Form1.img42.Visible:=false;

Form1.img43.Visible:=false;

Form1.img44.Visible:=false;

Form1.img45.Visible:=false;

Form1.img46.Visible:=false;

Form1.img47.Visible:=false;

Form1.img48.Visible:=false;

Form1.img49.Visible:=false;

Form1.img50.Visible:=false;

Form1.img51.Visible:=false;

Form1.img52.Visible:=false;

Form1.img53.Visible:=false;

Form1.img54.Visible:=false;

Form1.img55.Visible:=false;

Form1.img56.Visible:=false;

Form1.img57.Visible:=false;

Form1.img58.Visible:=false;

Form1.img59.Visible:=false;

Form1.img60.Visible:=false;

Form1.img61.Visible:=false;

Form1.img62.Visible:=false;

**if** puk=0 **then begin**

Form1.img64.Visible:=false;

Form1.img63.Visible:=True; **end**;

**end**;

**procedure** TForm1.img9MouseMove(Sender: TObject; Shift: TShiftState; X,

Y: Integer);

**begin**

Form1.img34.Visible:=True;

**end**;

**procedure** TForm1.img64Click(Sender: TObject);

**begin**

puk:=1;

name:='NBA';

img2.Visible:=true;

Form1.pnl1.show;

qry1.SQL.Clear;

qry1.SQL.add('Select \* from NBA ORDER BY PPG DESC');//('Select \* FROM NBA WHERE Team = '''+'LAL''');

qry1.Active:=True;

Form1.dbgrd1.Show;

form1.rg4.Visible:=true;

Form1.img63.visible:=false;

Form1.img64.visible:=false;

**end**;

**procedure** TForm1.rg4Click(Sender: TObject);

**begin**

**Case** rg4.ItemIndex **of**

0:**begin**

name:='NBA';

qry1.SQL.Clear;

qry1.SQL.add('Select \* from NBA ORDER BY PPG DESC');

qry1.Active:=True;

**end**;

1: **begin**

name:='R';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Exp = '''+ 'R''');

qry1.Active:=True;

**end**;

2: **begin**

name:='West';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'LAL'+''''+'OR Team = '''+'LAC'+''''

+'OR Team = '''+'HOU'+''''+'OR Team = '''+'DAL'+''''+'OR Team = '''+'POR'

+''''+'OR Team = '''+'MIN'+''''+'OR Team = '''+'OKC'+''''+'OR Team = '''+'UTA'

+''''+'OR Team = '''+'DEN'+''''+'OR Team = '''+'MEM'+''''+'OR Team = '''+'NOP'

+''''+'OR Team = '''+'SAS'+''''+'OR Team = '''+'GSW'+''''+'OR Team = '''+'PHX'+''''+'OR Team = '''+'SAC''');

qry1.Active:=True;

**end**;

3:**begin**

name:='East';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'BOS'+''''+'OR Team = '''+'NYK'+''''

+'OR Team = '''+'BRK'+''''+'OR Team = '''+'PHL'+''''+'OR Team = '''+'TOR'

+''''+'OR Team = '''+'ATL'+''''+'OR Team = '''+'CHA'+''''+'OR Team = '''+'MIA'

+''''+'OR Team = '''+'ORL'+''''+'OR Team = '''+'WAS'+''''+'OR Team = '''+'CHI'

+''''+'OR Team = '''+'CLE'+''''+'OR Team = '''+'DET'+''''+'OR Team = '''+'IND'+''''+'OR Team = '''+'MIL''');

qry1.Active:=True;

**end**;

4:**begin**

name:='Atlantic';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'BOS'+''''+'OR Team = '''+'NYK'+''''

+'OR Team = '''+'BRK'+''''+'OR Team = '''+'PHL'+''''+'OR Team = '''+'TOR''');

qry1.Active:=True;

**end**;

5:**begin**

name:='Central';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'IND'+''''+'OR Team = '''+'CLE'+''''

+'OR Team = '''+'MIL'+''''+'OR Team = '''+'DET'+''''+'OR Team = '''+'CHI''');

qry1.Active:=True;

**end**;

6:**begin**

name:='Southeast';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'ATL'+''''+'OR Team = '''+'ORL'+''''

+'OR Team = '''+'MIA'+''''+'OR Team = '''+'WAS'+''''+'OR Team = '''+'CHA''');

qry1.Active:=True;

**end**;

7:**begin**

name:='Northwest';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'UTA'+''''+'OR Team = '''+'MIN'+''''

+'OR Team = '''+'DEN'+''''+'OR Team = '''+'POR'+''''+'OR Team = '''+'OKC''');

qry1.Active:=True;

**end**;

8:**begin**

name:='Pacific';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'LAC'+''''+'OR Team = '''+'PHX'+''''

+'OR Team = '''+'SAC'+''''+'OR Team = '''+'LAL'+''''+'OR Team = '''+'GSW''');

qry1.Active:=True;

**end**;

9:**begin**

name:='Southwest';

qry1.SQL.Clear;

qry1.SQL.add('Select \* FROM NBA WHERE Team = '''+'NOP'+''''+'OR Team = '''+'SAS'+''''

+'OR Team = '''+'DAL'+''''+'OR Team = '''+'MEM'+''''+'OR Team = '''+'HOU''');

qry1.Active:=True;

**end**;

**end**;

**end**;

**procedure** TForm1.N1Click(Sender: TObject);

**begin**

ShowMessage('Player = Игрок'+ #13 + #10 +

'Team = Команда игрока'+ #13 + #10 +

'Position = Позиция игрока'+ #13 + #10 +

'Height = Рост'+ #13 + #10 +

'Weight = Вес'+ #13 + #10 +

'Exp = Опыт игры'+ #13 + #10 +

'GP = Игр сыграно'+ #13 + #10 +

'MPG = Среднее количество минут за матч'+ #13 + #10 +

'PPG = Среднее количество очков за матч'+ #13 + #10 +

'RPG = Среднее количество подборов за матч'+ #13 + #10 +

'APG = Среднее количество передач за матч'+ #13 + #10 +

'SPG = Среднее количество перехватов за матч'+ #13 + #10 +

'BPG = Среднее количество блокшотов за игру'+ #13 + #10 +

'TOV = Среднее количество потерь за игру'+ #13 + #10 +

'PF = Среднее количество фолов'+ #13 + #10 +

'ORB = Среднее количество подборов в атаке за матч'+ #13 + #10 +

'DRB = Среднее количество подборов в защите за матч'+ #13 + #10 +

'FGM = Среднее количество бросков за матч'+ #13 + #10 +

'FGA = Среднее количество попыток броска за матч'+ #13 + #10 +

'FG% = Средний процент попадания с игры'+ #13 + #10 +

'3PM = Среднее количество бросков из-за дуги за матч'+ #13 + #10 +

'3PA = Среднее количество попыток броска из-за дуги за матч '+ #13 + #10 +

'3P% = Средний процент попадания из-за дуги за матч'+ #13 + #10 +

'FTM = Среднее количество попаданий со штрафной линии за матч'+ #13 + #10 +

'FTA = Среднее количество попыток со штрафной за матч'+ #13 + #10 +

'FT% = Средний процент попадания со штрафной линии'+ #13 + #10 );

**end**;

**procedure** TForm1.N3Click(Sender: TObject);

**begin**

ShowMessage('Данная база данных игроков NBA с их подробной статистикой.'+ #13 + #10 +

'Благодаря ей можно проанализировать большинство аспектов'+ #13 + #10 +

'игры какого-либо игрока. В базе отображены такие важные показатели,'+ #13 + #10 +

'как очки, подборы, асисты, стилы, блокшоты и другое.'+ #13 + #10 +

'Таким образом, при помощи базы данных можно проанализировать' + #13 + #10 +

'практически все стороны игры того или иного игрока.'+ #13 + #10 +

'Цифры игроков говорят о многом и дают представление об их слабых и сильных сторонах.');

**end**;

**procedure** TForm1.N4Click(Sender: TObject);

**begin**

ShowMessage('Проект разработал студент ИВТб-2301-01-00 Кудяшев Ярослав');

**end**;

**procedure** TForm1.btn1Click(Sender: TObject);

**begin**

Form2.brsrsSeries1.Clear;

Form2.brsrsSeries2.Clear;

**for** i:=0 **to** 500 **do** //length(team) do

**begin**

**if** team[i] = name **then**

Form2.brsrsSeries1.Add(FG[i+1],Player[i],GetDefaultColor(1));

**end**;

**for** i:=0 **to** 500 **do** //length(team) do

**begin**

**if** team[i] = name **then**

Form2.brsrsSeries2.Add(P[i+1],Player[i],GetDefaultColor(1));

**end**;

Form2.Show;

**end**;

**end**.