

Функции

Основными функциями являются предоставление возможности ввода данных, заполнение и ведение различных справочников, а так же получение интересующей результативной информации посредством запросов или отчетов. Дерево функций представлено на рисунке.



Рисунок 1 - Дерево функций

Таблица 1- Команды

Команды			
Код	Название	Город	Главный тренер
1	Металлург	Кандалакша	Иванов С.М.
2	Пламя	Мурманск	Ботов И.Г.
3	Локомотив	Кировск	Степанов А.А.
4	Факел	Мончегорск	Забойнов К.В.

Таблица 2 – График игр

График игр						
Код матча	Дата проведения матча	Место проведения матча	Время проведения матча	Команда хозяев поля	Команда гостей	Главный судья матча
1	05.12.2013	Мурманск	12:00	Пламя	Металлург	Степанов А.Г.
2	10.12.2013	Кировск	15:00	Локомотив	Факел	Игнатов В.П.
3	12.12.2013	Мурманск	11:00	Пламя	Локомотив	Сенченко Н.В.
4	15.12.2013	Кандалакша	12:00	Металлург	Факел	Липатин Л.Г.
5	19.12.2013	Кандалакша	10:00	Металлург	Локомотив	Пашин К.С.
6	20.12.2013	Мончегорск	15:00	Факел	Пламя	Данилов С.С.

Таблица 3 – Забитые голы

Забитые голы					
Код гола	Код матча	Код футболиста, забившего гол	Код команды	Время от начала матча	Общий счёт игры
1	1	200	2	10	3:0
2	1	204	2	14	3:0
3	1	202	2	50	3:0
4	2	302	3	30	1:2
5	2	402	4	64	1:2
6	2	405	4	73	1:2
7	3	202	2	9	2:3
8	3	203	2	26	2:3
9	3	301	3	48	2:3
10	3	301	3	60	2:3
11	3	304	3	84	2:3
12	4	102	1	30	1:1
13	4	401	4	40	1:1
14	5	101	1	15	2:1
15	5	105	1	39	2:1
16	5	303	3	58	2:1
17	6	402	4	47	1:2
18	6	201	2	54	1:2
19	6	201	2	80	1:2

Таблица 4 - Футболисты

Футболисты			
Код	ФИО	Специализация	Код команды
100	Щаев Е.И.	вратарь	1
101	Липунцов В.П.	нападающий	1
102	Малофеев И.В.	нападающий	1
103	Бугаев П.С.	левый защитник	1
104	Романенко А.В.	правый защитник	1
105	Жёлудев Ю.П.	центральной защитник	1
106	Донцов Р.О.	центральной защитник	1
107	Шевченко Л.Ю.	центральной защитник	1
108	Татаринов Ю.В.	левый полузащитник	1
109	Аркадьев А.Е.	правый полузащитник	1
110	Григорьев А.Л.	центральной полузащитник	1
200	Павлов И.А.	вратарь	2
201	Вертянов С.С.	нападающий	2
202	Бушуев К.А.	нападающий	2
203	Болтунов А.И.	левый защитник	2
204	Алексеев А.В.	правый защитник	2
205	Сергеев А.Д.	центральной защитник	2
206	Зотов А.В.	центральной защитник	2
207	Зарученский Д.Л.	центральной защитник	2
208	Михалёв Б.А.	левый полузащитник	2
209	Новиков Г.Р.	правый полузащитник	2
210	Перфилов Г.В.	центральной полузащитник	2
300	Щедров И.Ю.	вратарь	3
301	Уразов В.К.	нападающий	3
302	Шорохов В.Н.	нападающий	3

Футболисты			
Код	ФИО	Специализация	Код команды
303	Волков Н.В.	левый защитник	3
304	Ким В.Н.	правый защитник	3
305	Анищенков В.Д.	центральной защитник	3
306	Шугаев Д.Д.	центральной защитник	3
307	Задарнов Д.А.	центральной защитник	3
308	Смыслов Д.П.	левый полузащитник	3
309	Крымцов Е.И.	правый полузащитник	3
310	Прохоров Е.С.	центральной полузащитник	3
400	Рогов Н.П.	вратарь	4
401	Носков А.Р.	нападающий	4
402	Копченков С.С.	нападающий	4
403	Федоров И.В.	левый защитник	4
404	Сержантов Л.А.	правый защитник	4
405	Бубликов Р.И.	центральной защитник	4
406	Ушаков К.С.	центральной защитник	4
407	Усольцев Л.А.	центральной защитник	4
408	Гусаров Е.Л.	левый полузащитник	4
409	Григорьев Л.А.	правый полузащитник	4
410	Гриб Я.И.	центральной полузащитник	4

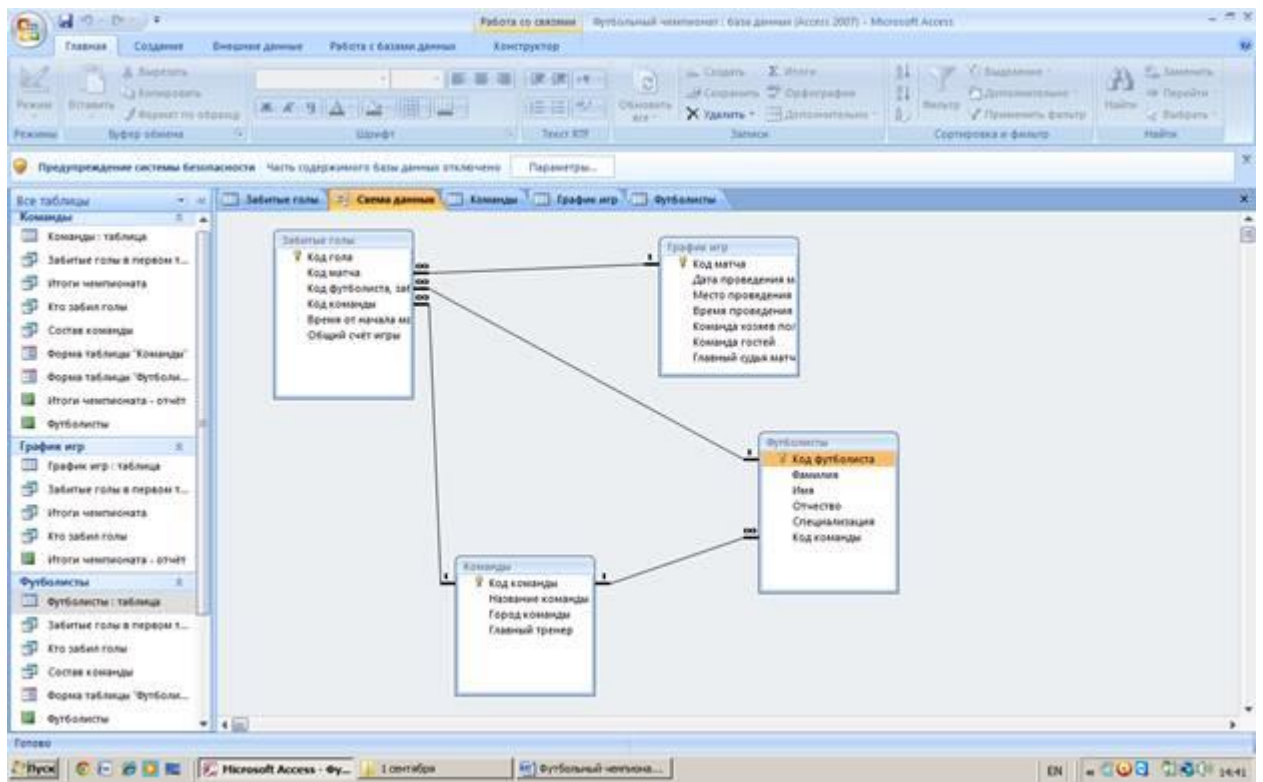


Рисунок 2 -Схема данных

Модель состава системы «Чёрный ящик» - даёт описание входящих в неё элементов, но не рассматривает связи между ними:

Чемпионат по футболу

→ Игроки

→ Команды

→ График игр

забитые голы

UML диаграмма классов



Контрольные вопросы

1. Предварительное проектирование формирует абстракции архитектурного уровня, детальное проектирование уточняет эти абстракции, добавляет подробности алгоритмического уровня. Кроме того, во многих случаях выделяют интерфейсное проектирование, цель которого — сформировать графический интерфейс пользователя (GUI).

2. Системные паттерны (system patterns) описывают системный (верхний) уровень приложения. Их использование может обеспечить построение более эффективных систем. Основные виды системных паттернов: Модель-Вид- Контроллер - разделение приложения или подсистемы на три функциональные части (модель данных, представление (пользовательский интерфейс) и контроллер (управляющую логику)). Сессия, Рабочая нить - в системах распределённой обработки обеспечивает возможность серверам различать клиентов, что позволяет повысить эффективность многопоточных приложений Обратный вызов - организация асинхронного взаимодействия между клиентом и сервером, что снижает нагрузку на сеть, повышает эффективность в использовании процессорного времени как клиента, так и сервера. Постоянное обновление - обеспечение возможности для клиента постоянного автоматического получения обновлений от сервера.

3. Паттерны управления разделены на паттерны централизованного управления (то есть паттерны, в которых одна из подсистем полностью отвечает за управление, запускает и завершает работу остальных подсистем) и паттерны управления, подразумевающие децентрализованное реагирование на события, (согласно этим паттернам на внешние события отвечает соответствующая подсистема.). Вызов — возврат - вызов программных процедур осуществляется "сверху - вниз", то есть управление начинается на вершине иерархии процедур и через вызовы передается на нижние уровни иерархии. Диспетчер - один системный компонент назначается диспетчером и управляет запуском и завершением других процессов системы и координирует эти процессы. Процессы могут протекать параллельно. Передача сообщений - В рамках данного паттерна событие представляет собой передачу сообщения всем подсистемам. Любая подсистема, которая обрабатывает данное событие, отвечает на него. Управляемый прерываниями - при использовании данного паттерна внешние прерывания регистрируются обработчиком прерываний, а обрабатываются другим системным компонентом.

4. Связанность модуля является мерой взаимозависимости модулей. При создании систем необходимо стремиться к максимальной независимости

модулей, т.е. связанность модулей должна быть минимальной. Модули связаны по данным, если они взаимодействуют через передачу параметров и при этом каждый параметр является элементарным информационным объектом. Это наиболее предпочтительный тип связанности (сцепления). Модули связаны по образцу если один модуль посылает другому составной информационный объект (например, объект – библиографическая запись, которая содержит имя автора, название книги и т.д.). Модули связаны по управлению, если один посылает другому информационный объект – флаг, предназначенный для управления его внутренней логикой. Модули связаны по общей области в том случае, если они ссылаются на одну и ту же область глобальных данных. Связанность (сцепление) по общей области является нежелательным, так как, во-первых, ошибка в модуле, использующем глобальную область, может неожиданно проявиться в любом другом модуле; во-вторых, такие программы трудны для понимания, так как программисту трудно определить какие именно данные используются конкретным модулем. Модули связаны по содержимому в том случае, если один из них ссылается внутрь другого. Это недопустимый тип сцепления, ибо полностью противоречит принципу модульности, т.е. представления модуля в виде черного ящика.

5. Сцепление - мера взаимозависимости модулей по данным [3]. Сцепление - внешняя характеристика модуля, которую желательно уменьшать. Независимые модули могут быть модифицированы без переделки каких-либо других модулей. Количественно сцепление измеряется степенью сцепления (СЦ). Выделяют 7 типов сцепления. 1. Полностью независимые модули (СЦ = 0). Модули, не вызывающие друг друга и не использующие общих данных, не сцеплены и являются полностью независимыми. Чем больше информации о других модулях используется в них, тем менее они независимы и тем сильнее сцеплены. 2. Сцепление по данным. Модуль А вызывает модуль В. Все входные и выходные параметры вызываемого модуля - простые элементы данных. 3. Сцепление по образцу. В этом случае модули ссылаются на одну и ту же глобальную структуру данных. Недостатком такого сцепления является то, что оба модуля должны знать о внутренней структуре данных. 4. Сцепление по общей области. Модули разделяют одну и ту же глобальную структуру данных. В этом случае возможностей для появления ошибок при модификации структуры данных в одном модуле много больше. 5. Сцепление по управлению. Модуль А явно управляет функционированием модуля В с помощью передачи флагов, переключателей или кодов, посылая ему управляющие данные. Возвращение флага состояния как неявной переменной не означает

сцепления по управлению. Если модуль передает информацию о самом себе или об обработанных данных, то это не всегда служит проявлением сцепления по управлению. Передача флага конца файла позволяет решить вопрос о возможности обработки этого файла. 6. Сцепление по внешним ссылкам. Модуль имеет сцепление по внешним ссылкам, если у него есть доступ к данным в другом модуле через внешнюю точку входа. 7. Сцепление по кодам. Один модуль прямо ссылается на содержание другого модуля (не через его точку входа). Например, коды их команд перемежаются друг с другом.

6. Делегирование — основной паттерн проектирования, в котором объект внешне выражает некоторое поведение, но в реальности передаёт ответственность за выполнение этого поведения связанному объекту. Шаблон делегирования является фундаментальной абстракцией, на основе которой реализованы другие шаблоны - композиция (также называемая агрегацией), примеси и аспекты. Неизменяемый объект — в объектно-ориентированном программировании объект, который не может быть изменён после своего создания. Интерфейс — основной паттерн проектирования, являющийся общим методом для структурирования компьютерных программ для того, чтобы их было проще понять. В общем, интерфейс — это класс, который обеспечивает программисту простой или более программно-специфический способ доступа к другим классам. MVC расшифровывается как модель-представление-контроллер. Это способ организации кода, который предполагает выделение блоков, отвечающих за решение разных задач. Один блок отвечает за данные приложения, другой отвечает за внешний вид, а третий контролирует работу приложения.