**КУРСОВАЯ РАБОТА**

**по дисциплине: Базы данных**

**на тему: «АИС для ГОУДОД Центра развития творчества детства и юношества»**

**Содержание**

1.Теоретическая часть

1.1 Процессы проектирования

1.2 Инфологическое проектирование

## 1.3 Концептуальное проектирование

## 1.4 Логическое проектирование

## 1.5 Средства создания модели

## 2. Практическая часть

2. 1 Специальная часть

2.1.1 Этапы выполнения курсовой работы

2.2 Технологическая часть

2.2.1 Физическое проектирование

2.3 Инструкция пользователю

Заключение

**1. Теоретическая часть**

##

## Процессы проектирования

Проектирование программных систем складывается из проектирования процессов, данных и событий. В силу специфики курса мы рассматриваем лишь проектирование данных – процесс разработки структуры базы данных в соответствии с требованиями заказчиков. В ходе разработки проекта нужно ответить на следующие вопросы:

* что представляют собой требования заказчиков, и в какой форме они выражены;
* как они преобразуются в структуру базы данных;
* как часто и каким образом структура базы данных должна перестраиваться.

В настоящее время рассматриваются три уровня абстракции для определения структуры данных: концептуальный (точка зрения заказчика), логический (точка зрения разработчика) и физический (точка зрения администратора БД). В соответствии с этим рассматриваются три уровня модели и три шага проектирования. Некоторые источники утверждают, что ни физической модели, ни шага физического проектирования на самом деле нет. Обсуждение этого взгляда проведем, когда уточним содержание всех трех видов абстракции.

Концептуальный уровень – наиболее общее представление об информационном содержании предметной области. Представляется в виде концептуальной модели (КМ), которая часто называется концептуальной схемой или информационной структурой. КМ обладает высокой степенью стабильности, она проблемно-ориентирована и не зависит от конкретной СУБД, операционной системы и аппаратного обеспечения. Ее поведение должно быть полностью предсказуемо.

Концептуальное представление оперирует основными элементарными данными предметной области, называемыми сущностями. Сущности описываются атрибутами. Данные могут находиться в некотором отношении друг с другом: образовывать ассоциации. Эти ассоциации называются связями. КМ должна поддерживать согласованность связей в пределах уровня детализации.

Обычно для концептуального представления используется модель «Сущность-Связь» (ER-модель), которая графически выражается ER-диаграммами. Существуют различные модификации представления (нотации) диаграмм. Ранее уже приводились сведения о ER-модели. Добавим, что, не только сущности, но и связи могут иметь атрибуты, выражающие их свойства. Представление модели внешне напоминает структуру базы данных и служит для отображения на логическую модель.

Логический уровень представления оперирует такими понятиями, как запись, компоненты записи, связи между записями. Соответствующая ему модель называется логической (ЛМ), она представляет собой отображение концептуальной модели в среду конкретной СУБД. Иногда рассматривают не конкретную СУБД, а только ее класс (модель) – иерархическую, сетевую или реляционную. Особенности этих моделей рассматривались ранее.

Физический уровень демонстрирует физическое хранение данных. На этом уровне используются такие понятия, как физические блоки, файлы, хранимые записи, указатели. Взаимосвязи между хранимыми записями, возникающие в процессе их группировки, а также индексные структуры тоже рассматриваются на уровне физической модели (ФМ). С точки зрения чистой базы данных можно абстрагироваться от физической модели представления данных, но знать ее очень полезно для достижения более высокой производительности системы. Если есть возможность влиять на ФМ и есть представление о способах оптимизации в ее рамках, воспользоваться ими может быть полезно, особенно для распределенных данных. Но реально почти никогда это не делается.

Есть и другая классификация уровней представления данных. Согласно стандарту ANSI/SPAC, архитектура БД представлена трехуровневой моделью с внешним, концептуальным и внутренним уровнями. В отличие от предыдущей модели, это не модель проектирования, модель оперирования данными.

Внешний уровень – описание на языке пользователя структуры данных, вида и формы их представления, а также описание операций манипулирования данными. Считается, что для описания предметной области используется несколько внешних моделей. Данный уровень содержит черты как КМ, так и ЛМ, описанных ранее.

Концептуальный уровень – наиболее общее представление об информационном содержании предметной области. Определение совпадает с приведенным ранее.

Внутренний уровень – организованная совокупность структурированных данных, отображение концептуальной модели в конкретную среду хранения. Легко видеть, что это понятие объединяет ранее определенные логическую и физическую модели.

Рассматривая эти два подхода к описанию представления данных, приходим к выводу, что первый более прагматичен. В нем предметная область рассматривается как единое целое, а не как совокупность проектных требований, называемых внешними моделями. Реально проектные требования редко можно назвать полноценной моделью, так как любая модель должна давать на каком-то уровне адекватное представление о предметной области. Полученные же требования зачастую выступают как совокупность представлений о ней разных групп пользователей. Такая ситуация возникает в тех случаях, когда аналитик считает, что пользователи формулируют свои знания как локальные модели, совокупность которых и должна составлять требуемую модель. Неверность этого утверждения хорошо иллюстрируется известной индийской сказкой об исследовании слона пятью слепцами, в ходе которого они предложили свои локальные модели слона (исследователь хобота считал, что слон похож на канат, хвоста – на метелку, бока – на стену, уха – на лист, ноги – на колонну). Реально пользователь часто не в состоянии построить даже локальную информационную модель, а про глобальные связи между ними и говорить не приходится. Другое соображение. Популярное в настоящее время направление проектирования – перепроектирование технологических процессов (реинжиниринг бизнес-процессов – BPR) – отрицает такой подход в силу того, что он консервирует существующую технологию и не дает выделить цель производства. А раз невозможно выделить общую цель производства, результат данного этапа исследования нельзя считать моделью. Но можно, введя понятие типа пользователя (эксперта), рассматривать соответствующую внешнюю модель как точку зрения этого эксперта на предметную область. В этом случае концептуальная модель представляется как единое целое, дополненное совокупностью точек зрения экспертов.

Инфологический подход не предоставляет формальных способов моделирования реальности, но он закладывает основы методологии проектирования баз данных.

Теперь рассмотрим три основных уровня проектирования: инфологическое, концептуальное и логическое – с точки зрения первого подхода.

* 1. **Инфологическое проектирование**

Основными задачами инфологического проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО.

Инфологическая модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависимых от реализации БД. Это описание выражается в терминах не отдельных объектов ПО и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое.

## Концептуальное проектирование

На этапе концептуального проектирования определяются информационные потребности и локальные представления предметной области. Выявляется роль, назначение, взаимосвязь данных, проводится их глобальная спецификация. Результат этапа – описание объектов данных и их взаимосвязи без указания способа их физической организации. Структура данных представляется концептуальной схемой, содержащей набор сущностей, связей и атрибутов. Различаются две важных стадии концептуального проектирования: анализ данных и организация их хранения.

Содержание первой стадии – сбор полной и точной информации о данных предметной области. Заметим, что речь идет о первоначальном сборе информации, в процессе проектирования, как правило, выясняются дополнительные обстоятельства. Нередко для проведения данной работы прибегают к одному из двух методов (или к обоим): анкетированию и работе с экспертами. В первом случае пользователю предлагается анкета, в которой он должен дать свое представление о данных. Во втором случае среди пользователей выбирается группа экспертов, которые все и излагают. Оба этих метода страдают большими недостатками. Анкетирование никогда не гарантирует полноту информации: к этому процессу пользователи относятся как к досадной помехе в основной деятельности, поэтому стараются не дать точные сведения, а поскорее «отбиться». Работа с экспертами несколько лучше, но требует затрат квалифицированного труда. Кроме того, эксперты, будучи специалистами в своей области, не всегда способны достаточно внятно и полно изложить суть проблемы целиком. Тем не менее, не стоит пренебрегать этими методами, но за основу следует взять личное участие разработчиков в исследовании. В частности, следует потребовать набор выходных и входных документов и проследить их движение и модификацию на каждом этапе технологического процесса. Во время такого исследования выявляются недостатки принятого документооборота и делаются попытки, как это предлагает BPR, его оптимизировать с точки зрения машинной обработки.

Вторая стадия сводится к разработке графического представления полученной информации в виде схемы, которая включает, в частности, исходные данные с формирующими их процессами и результирующие со ссылкой на использующие процессы. На этом же этапе уточняется степень важности данных, выявляются и фиксируются связи между ними. К данной работе, наряду с проектировщиком, полезно привлекать администратора баз данных и представителей пользователя.

## Логическое проектирование

Роль логическое проектирования – отображение КМ в выбранную модель данных. На этом этапе необходимо определить отношения и атрибуты, выделить ключи. На ряд атрибутов могут быть наложены ограничения, которые выражаются в функциональных зависимостях между ними. Если выбрана реляционная модель данных, в процессе проектирования следует так определять отношения, чтобы атрибуты в каждом из них функционально полно зависели от ключей и не было транзитивной зависимости атрибутов в отношении. В результате должна сформироваться логическая схема БД, находящаяся в 3 нормальной форме. Эта схема, разумеется, не окончательная, в процессе проектирования она может неоднократно корректироваться, в результате чего нормализованность может нарушиться. В этом случае добавляется специальный этап нормализации схемы. Основное назначение этапа нормализации – получение схемы, эквивалентной данной, но не обладающей некоторыми отрицательными свойствами, связанными с функциональными зависимостями.

Семантическая мощность БД возрастает с увеличением числа дополнительных характеристик, таких, как контроль полномочий, контроль достоверности исходных данных, контроль ограничения целостности. При обсуждении реляционной модели говорилось, что в полной мере такими свойствами обладают лишь полностью реляционные СУБД.

Контроль полномочий (разграничение прав доступа) служит защитой от несанкционированного доступа к данным. Для него обычно используется система паролей.

Процедуры контроля исходных данных по заданным ограничениям могут быть как внешними по отношению к базам данных, так и встроенными, если СУБД допускает возможность хранимых процедур.

Ограничения целостности служат для защиты данных от некорректных изменений. Различают статические ограничения, отражающие множество корректных состояний БД, и динамические, определяющие правильные переходы из состояния в состояние. Соответственно, ограничения целостности обеспечиваются вызовом при модификации данных программ статического или динамического арбитража.

## Средства создания модели

Создание ER-модели обычно сопровождается ее графическим представлением. Различные нотации ER-диаграмм поддерживается специальными средствами проектирования программных систем (CASE-средствами). В некоторых случаях подобные средства включаются в инструментальную среду создания баз данных (FoxPro, Access, Oracle и т.п.), иногда они существуют как отдельный продукт. Последний вариант интересен тем, что нет привязки к конкретной СУБД, то есть можно определять концептуальную модель, которая может быть отображена в логическую после того, как будет выбрана СУБД. Рассмотрим один из наиболее популярных средств такого рода – ERWin.

CASE-средство ERWin уже достаточно долго присутствует на рынке инструментальных средств, ориентированных на базы данных. Существует несколько версий этого продукта. ERWin позволяет создавать модели двух уровней: концептуального и логического. Правда называются они довольно своеобразно: концептуальный называется логическим, а логический – физическим. Представление диаграмм соответствует стандарту IDEF1X. Основные элементы модели следующие: сущности, атрибуты сущностей, домены, связи (четыре вида), индексы. Современные версии ERWin не допускают атрибуты связей. Среди атрибутов выделяются первичные ключи. Кроме первичных, можно отметить возможные (альтернативные) ключи, а также ключи поиска. Внешние ключи формируются автоматически при определении связей, причем, в качестве родительского ключа выбирается первичный.

Сильная сторона ERWin заключается в относительной простоте и удобстве работы с продуктом, что немаловажно в случае сжатых сроков разработки (а это случается постоянно). Разумеется, поддерживаются различные уровни документирования модели: описания сущностей, атрибутов, связей, диаграммы, модели в целом и т.п. Есть возможность получить пакет документации о модели с использованием внутреннего генератора отчетов.

После выбора СУБД появляется возможность привязки к типам данных, принятых в этой СУБД. В терминах ERWin, эта работа производится в физической модели. Можно уточнить типы данных, индексы, а также задать хранимые процедуры, обеспечивающие корректность базы данных. Заметим, что в этой модели несколько меняется терминология: сущности называются таблицами, а атрибуты – столбцами. И, наконец, завершающая работа – автоматическая генерация пустой базы данных по разработанной модели. Базу данных следует проверить, особенно индексы, заполнить тестовыми данными и протестировать. Обычно для создания работоспособной базы данных требуется несколько итераций, в ходе которых изменять следует исключительно модель, а не созданную базу данных.

**2. Практическая часть**

**2. 1 Специальная часть**

база данные проектирование инфологический

**2.1.1 Этапы выполнения курсовой работы**

Концептуальное проектирование.

Программами, реализуемыми в ГОУ ДОД «ЦРТДиЮ им. В.М. Комарова», обеспечивающими целостность образовательно-воспитательного процесса, являются программы дополнительного образования детей, утвержденные Министерством просвещения.

В условиях современного развития системы дополнительного образования детей ГОУ ДОД «ЦРТДиЮ им. В.М. Комарова» стремиться реализовать идею общего интеллектуального развития личности ребенка путем введения развивающих и интегрированных курсов, профильного обучения, повышения мастерства воспитанников.

Одним из основных показателей качества образовательного процесса в ГОУ ДОД «ЦРТДиЮ им. В.М. Комарова» является характеристика его содержания, которая зависит от гармоничного единства теории и практики, классического и современного знания, воспитания и развития. В соответствии с требованиями нормативных документов, ГОУ ДОД «ЦРТДиЮ им. В.М. Комарова» имеет свою образовательную программу, которая является ориентирующей моделью совместной деятельности педагога и ребенка.

Задачи, подлежащие автоматизации:

1. Учет кружков;
2. Учет сотрудников;
3. Учет воспитанников;
4. Учет мероприятий.

Для настоящего курсового проекта входными данными послужили:

1. Структурная схема ГОУДОД ЦРТДиЮ.
2. Списки воспитанников.
3. Списки сотрудников.
4. Перечень существующих кружков и проводимых мероприятий.

Выходными данными являются:

1. Отчеты, сформированные по запросам;
2. Простые запросы: по кружкам, воспитанникам, сотрудникам, мероприятиям.
3. Перекрестные запросы.

Для сотрудников по должностям, для воспитанников по гражданству, для мероприятий по дате.

Требования пользователя заключаются в легкости исправления и внесения изменений в настоящую базу данных.

Концептуальная модель представлена в Приложении Б.

Логическое проектирование.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие так и абстрактные. Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Выделим базовые сущности этой предметной области:

* Сотрудники. Атрибуты сотрудников – Фамилия, Имя, Отчество, Должность, Название отдела, Табельный номер, Номер паспорта, Адрес, Город, Дата рождения. Для сотрудников необходимо хранить сведения о заказах, которые они ведут.
* Кружки. Атрибуты кружков – наименование кружка, наименования отдела, Имя и Фамилия педагога. Для Кружков необходимо ранить сведения об отделах, за которыми закреплены кружки.
* Воспитанники. Атрибуты воспитанников – Фамилия, Имя, Дата рождения, кола, Класс, родной язык, Гражданство, Домашний адрес и наименование кружка.
* Мероприятия. Атрибуты мероприятия – Наименование мероприятия, дата проведения, наименование кружка.

Требования к сущностям:

Сущность Сотрудники – должна содержать полную информацию о каждом работающем сотруднике в организации.

Сущность Кружки - должна содержать информацию о том, как называется кружок, кто ведет кружок (ФИ педагога) и отдел, к которому относится кружок.

Сущность Воспитанники – должна содержать полную информацию об учащихся, записанных в кружках, и указывается наименование кружка, в котором занимается ребенок.

Связь один-ко-многим. Такая связь подразумевает, что каждому объекту одного набора соответствуют ноль или более объектов второго набора, и каждому объекту второго типа соответствует не более одного объекта первого типа.

Необходимо так же определить первичный ключ. Если таблица никогда не будет использоваться в качестве главной, то ключ для нее определять не нужно. В главных таблицах обычно содержится информация о реальных объектах, причем с каждым объектом ассоциируется только одна запись. Определение ключа таблицы является простейшим способом предотвращения появления в таблице одинаковых записей. В главной таблице связи должен быть определен первичный ключ. Access считает таблицы, у которых такой ключ не определен, подозрительными. При открытии таких таблиц в режиме конструктора появляется диалоговое окно, сообщающее о том, что ключ таблицы не определен. Ключ можно определить и в связанных таблицах, что поможет избежать появления повторяющихся данных. Ключ таблицы можно задать по значению нескольких полей. Access автоматически индексирует таблицу по значению ключа.

В данной курсовой работе для каждой существующей сущности ключом является первое поле т.е. код. В сущности Сотрудники – поле код сотрудника. В сущности Воспитанники – код воспитанника. В сущности Кружки – код кружка. В сущности Мероприятия – код мероприятия.

Концептуальная модель представлена в виде концептуальной схемы в Приложении В.

**2. 2 Технологическая часть**

**2.2.1 Физическое проектирование**

Выбор специального программного обеспечения.

Access — это, прежде всего, система управления базами данных (СУБД). Она предназначена для хранения и поиска данных, представления информации в удобном виде и автоматизации часто повторяющихся операций (таких, как ведение счетов, учет, планирование и т.п.). С помощью Access можно разрабатывать простые и удобные формы ввода данных, а также осуществлять обработку данных и выдачу сложных отчетов.

Система Access — это набор инструментов конечного пользователя для управления базами данных. В ее состав входят конструкторы таблиц, форм, запросов и отчетов. Эту систему можно рассматривать и как среду разработки приложений. Используя макросы или модули для автоматизации решения задач, можно создавать ориентированные на пользователя приложения такими же мощными, как и приложения, написанные непосредственно на языках программирования. При этом они будут включать кнопки, меню и диалоговые окна. Программируя на языке VBA, можно создавать такие мощные программы, как сама система Access. По сути дела, многие инструментальные средства Access (например, мастера и конструкторы) написаны именно на VBA.

Мощность и доступность Access делают эту систему лучшей СУБД из представленных сегодня на рынке. Сначала познакомимся с Access на уровне конечного пользователя. Затем перейдем к более сложным элементам таким как элементы программирования на VBA и взаимодействия с Internet.

Создание таблицы в режиме Конструктора таблиц. Настоящей курсовой работе таблицы были созданы с помощью конструктора. Для этого были проделаны следующие действия:

* Для перехода в окно базы данных нажали клавишу F11.
* Выбираем Таблицы в списке Объекты и нажимаем кнопку Создать на панели инструментов окна базы данных.
* Дважды щелкаем строку Режим конструктора.
* Определяем все нужные поля в таблице.
* До того, как сохранить таблицу, определяем первичный ключ.
* Для сохранения таблицы нажимаем кнопку Сохранить на панели инструментов, а затем введите для нее уникальное имя.

Этот режим позволяет определить требуемую структуру таблицы. В режиме конструктора таблиц можно увидеть названия, типы данных, описания назначения, а также некоторые дополнительные свойства полей таблицы. То, что появляется в верхней части окна таблицы, открытой в режиме конструктора, называется бланком структуры таблицы или просто бланком таблицы. В нижней части окна таблицы, открытой в режиме конструктора, выводятся свойства полей таблицы, а также краткое описание активного свойства таблицы.

- Заданы связи межу таблицами. Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях — обычно между полями разных таблиц. Таблица, содержащая ключевое поле, называется главной, а таблица, содержащая внешний ключ — связанной. Используется вид отношений Один ко многим. Связь с отношением "один ко многим" является наиболее часто используемым типом связи между таблицами. В такой связи каждой записи в таблице "А" могут соответствовать несколько записей в таблице "В", а запись в таблице "В" не может иметь более одной соответствующей ей записи в таблице "А". В таблице "А" ключевое поле должно быть уникальным.

Автоматическое обеспечение ссылочной целостности данных является важной особенностью Access. Если на связь между таблицами наложены условия ссылочной целостности, то Access не позволяет добавлять в связанную таблицу записи, для которых нет соответствующих записей в главной таблице, и изменять записи в главной таблице таким образом, что после этого в связанной таблице появятся записи, не имеющие главных записей; а также удалять записи в главной таблице, для которых имеются подчиненные записи в связанной таблице. Условия целостности данных определяют систему правил, используемых в Access для поддержания связей между записями в связанных таблицах. Эти правила делают невозможным случайное удаление или изменение связанных данных. После наложения условий целостности данных на операции со связанными таблицами накладываются ограничения. Невозможно ввести в поле внешнего ключа связанной таблицы значение, не содержащееся в ключевом поле главной таблицы.

Так же были созданы перекрестные и с условием отбора и на выборку Запросы. Разработка Запроса производится в режиме Конструктора. Для создания запроса выделите объект Запросы , нажмите кнопку Создать и выберите режим Конструктора. Укажите используемые в запросе таблицу или таблицы (как при работе со схемой данных).

Верхнее окно Запроса в режиме Конструктора отображает схему данных (можно при необходимости изменять связи между таблицами), а нижнее Бланк Запроса По образцу. Каждый столбец бланка относится к одному полю, с которым нужно работать в запросе. Поля могут использоваться для включения их в результат выполнения запроса , для задания сортировки, а также для создания условий отбора записей Выполняя запрос, получаем ответ в виде так называемого динамического набора данных, который не сохраняется при сохранении запроса, однако он может быть использован как источник данных (наравне с таблицами) для построения другого запроса. Динамический набор данных можно сохранить, например в таблице, если выполнить запрос на создание таблицы или запрос на добавление данных, а также использовать в формах и отчетах

Для просмотра результата запроса необходимо воспользоваться кнопкой Представление Таблицы на панели или пунктом меню Вид – Режим Таблицы. Для просмотра запроса в режиме SQL необходимо воспользоваться пунктом меню Вид – Режим SQL

Запросы на выборку отображают данные из одной или нескольких таблиц в виде таблицы в данном случае это запросы: Кружки, Сотрудники, Воспитанники, Мероприятия;

Параметрические запросы. Конкретное значение поля в условии отбора задается пользователем при выполнении Запросы в диалоговом окне. Например используя таблицу СОТРУДНИКИ создан параметрический запрос, позволяющий просматривать только педагогов. Для этого в строку Условие отбора ввести [педагог] для поля Должность. Результат можно просмотреть в режиме таблицы.

В данном примере введено условие, позволяющее пользователю самому вводить должность сотрудника.

Созданы Формы. Порядок разработки простых форм:

1. В окне базы данных выбрать вкладку "Форма" и нажать кнопку "Создать".

2. Выбрать в поле "Источник данных ..." имя таблицы или запроса, на котором будет базироваться форма, а также способ создания формы - мастер форм.

3. В окне "Создание форм" переведите поля, размещаемые на форме из области "доступные поля" в область "выбранные поля" и нажмите "Далее".

4. Выберите тип формы - ленточный и нажмите "Далее".

5. Выберите стиль (фон) формы и нажмите "Далее". Для формы, выводимой на печать, желательно не задавать темный фон.

6. Задайте имя формы (в соответствие с базовой таблицей или запросом) и нажмите "Готово".

7. Откройте созданную форму, просмотрите ее и, при необходимости, перейдите в режим конструктора и вручную измените подписи в области заголовка (если они слишком широки по сравнению с содержимым соответствующих полей) или измените размер и расположение полей в рабочей области.

# Создание главной кнопочной формы:

1. Выполните Сервис/Служебные программы/Диспетчер кнопочных форм.
2. В окне "Диспетчер кнопочных форм" нажмите "Изменить...".
3. В окне "Изменение страницы кнопочной формы" нажмите "Создать" - откроется окно "Изменение элемента кнопочной формы" с тремя полями: в поле "Текст" введите надпись, соответствующую действию кнопки; в поле "Команда" выберите команду "Открытие формы в режиме редактирования" - для кнопок, открывающих форму или команду "Выход из приложения" - для кнопки завершения работы; информация, выбираемая в третьем поле, зависит от второго - для открытия формы надо указать имя открываемой формы.
4. Повторите предыдущий пункт для создания остальных кнопок кнопочной формы.
5. Закройте окно создания главной кнопочной формы, нажав "Закрыть"

- Созданы Отчёты. Конечным продуктом большинства приложений баз данных является отчет. В Access отчет представляет собой специальный тип непрерывных форм, предназначенных для печати. Для создания отчета, который можно распечатать и распределить между потребителями, Access комбинирует данные в таблицах, запросах и даже формах. Распечатанная версия формы может служить отчетом.

Все отчеты для данной работы созданы в режиме конструктора. Чтобы перейти в режим Конструктора, выберите команду Вид, Конструктор или нажмите на стрелку, расположенную справа от кнопки Вид панели инструментов Конструктор отчетов и в открывшемся списке выберите элемент Конструктор.

* 1. **Инструкция пользователю**

При запуске программы «АИС ГОУДОД ЦРТДиЮ», появляется главное меню, которое содержит 3 кнопки: «Формы», «Отчеты», «Выход».

При нажатии кнопки «Формы» открывается новая кнопочная форма, в которой имеются четырнадцать копок:

* «Кружки» - «Запрос»
* «Сотрудники» - «Запрос» - «По должности»;
* «Мероприятия» - «Запрос» - «по дате»;
* «Воспитанники» - «Запрос» - «по кружку» - «по гражданству»;
* «на главную»;
* «выход».

При нажатии кнопки «Кружки» открывается форма «Кружки». Данная форма предназначена для хранения, добавления, изменения либо удаления данных о кружках. Рядом расположена кнопка запроса для кружков.

При нажатии кнопки «Сотрудники» открывается форма “Сотрудники”, на которой имеется информация о сотрудниках организации. В данной форме можно вносить изменения в записи.

* «Найти запись» предназначена для поиска по записям.
* «Выход» предназначена для закрытия формы.

К кнопке «Сотрудники» прикреплена кнопка «Запрос», с помощью которой выводится запрос по всем сотрудникам.

* «Найти запись» предназначена для поиска по записям.
* «Выход» предназначена для закрытия формы.

Кнопка «по должностям» позволяет вывести отчет сотрудников по должностям.

При нажатии кнопки «Мероприятия» открывается форма Мероприятия. Также ядом находится кнопка запроса и кнопка «по дате» предназначена для вывода запроса мероприятий по дате.

* «Выход» предназначена для закрытия формы.

При нажатии кнопки «Воспитанники» открывается форма “Воспитанники”, на которой имеется информация об учащихся в кружках. В данной форме можно вносить изменения в записи. К кнопке «Сотрудники» прикреплена кнопка «Запрос», с помощью которой выводится запрос по всем воспитанникам.

* «по гражданству» предназначена для вывода учащихся по гражданству;
* «по кружку» предназначена для вывода запроса по кружку;
* «Найти запись» предназначена для поиска по записям.
* «Выход» предназначена для закрытия формы.

При нажатии на главной форме кнопки «Отчеты» открывается новая кнопочная форма «Отчеты», в которой перечислены отчеты, подготовленные на основе информации содержащийся в разработанной АИС.

Данная форма содержит двенадцать кнопок:

- «Отчет кружки» - кнопка “Распечатать данный отчет”

- «Отчет сотрудники» - кнопка “Распечатать данный отчет”

- «Отчет воспитанники» - кнопка “Распечатать данный отчет”

- «Отчет мероприятия» - кнопка “Распечатать данный отчет”

- «Отчет педагоги» - кнопка “Распечатать данный отчет”

- «На главную»

Кнопка «Выход» - осуществляет выход из программы.

Интерфейс пользователю представлен в Приложении Г.

**Заключение**

Настоящая курсовая работа посвящена проектированию АИС для ГОУДОД ЦРТДиЮ.

В первой части пояснительной записки курсовой работы рассмотрена структура и назначение образовательного учреждения ЦРТДиЮ, описаны задачи и функции.

При создании программного обеспечения АИС для ГОУДОД ЦРТДиЮ произведен выбор программных средств и приведено обоснование данного выбора. Описан процесс проектирования базы данных, обеспечивающей работу АИС для ГОУДОД ЦРТДиЮ, в результате которого построена концептуальная модель БД.