[Название документа]

[Подзаголовок документа]

329195-4

[Год]

**Содержание**

ВВЕДЕНИЕ 4

1 Общая часть 5

1.1 Описание предметной области 5

1.2 Постановка задачи 6

1.3 Функции проектируемого приложения 7

1.4 Анализ рынка существующих решений 8

1.5 Архитектура приложения 10

1.6 Обоснование и выбор методики, технологии и инструментальных средств проектирования и разработки 12

1.6.1 Обоснование выбора используемой СУБД 12

1.6.2 Обоснование выбора языка программирования 13

1.6.3 Обоснование выбора используемой среды разработки 14

2 Специальная часть 15

2.1 Диаграмма прецедентов использования 15

2.2 Инфологическая модель 16

2.3 Даталогическая модель 17

2.4 Разработка пользовательского интерфейса 18

2.5 Исключительные ситуации 18

2.6 Разработанные запросы 18

2.7 Обеспечение целостности базы данных 18

2.8 Разграничение прав доступа. Защита данных 18

2.9 Диаграмма классов 18

ЗАКЛЮЧЕНИЕ 19

**Перечень сокращений и обозначений**

**БД** – база данных;

**ПО** – программное обеспечение;

**ИС** – информационная система;

**АБИС** – Автоматизированные библиотечные информационные системы;

**JSON** – JavaScript Object Notation;

**SQL** – Structured Query Language;

**ПК** – Персональный компьютер.

# ВВЕДЕНИЕ

Активное распространение и применение компьютерных технологий для хранения и передачи информации в библиотеках позволяет вывести на современный уровень организацию библиографического обслуживания, уменьшить количество ошибок, связанных с человеческим фактором, и упростить контроль сотрудников за сроками сдачи печатных изданий.

Данный проект актуален только в обучающих целях, поскольку он не был заказан ни одной библиотекой, то есть не востребован. Несмотря на это, он может понадобиться в будущем для создания более полной по функционалу версии уже по заказу библиотеки.

Цель курсового проекта – разработать информационную систему библиотеки для упрощения учета читателей и сроков сдачи печатных материалов.

Задачи курсового проекта:

* Проанализировать предметную область и источники, по которым будут составлены аналитические сведения, аналоги и технические средства, провести сравнение с существующими аналогами;
* Спроектировать UML-диаграммы, базу данных, макет интерфейса;
* Разработать программу, интерфейс, модули и алгоритмы;
* Протестировать в соответствии с требованиями технического задания, написать Unit-тесты, проверить, написать документацию (руководство оператора).

## Общая часть

### Описание предметной области

Библиотеки – это учреждение, собирающее и осуществляющее хранение произведений печати и письменности для общественного пользования, а также ведущее справочно-библиографическую работу. Библиотекам принадлежит важнейшая роль в глобальной инфраструктуре – роль интеллектуальных посредников, которые отбирают, организуют, хранят информацию и обеспечивают к ней доступ.

В зависимости от целей классификации библиотеки разделяются на роды и виды. Родовидовых классификаций может быть столь много, сколь много целей, ради которых они создаются. Если родовой при­знак — учредитель, то библиотеки можно разделить по видам на личные и общест­венные. Подвидами общественных являются государственные (их разновидности: общефедеральные, субъектов Федерации, отдельных министерств и ведомств, муниципальные), общественных организаций и объединений (разновидности: партийные, профсоюзные, конфессиональные, различных добровольных обществ). Если родовой признак — формы собственности, то в соответствии с ним можно выделить библиотеки частные, коммерческие (платные), бесплатные и т. д.

Библиотеки разделяются и по другим социально существенным признакам, например, по численности персонала, состоянию материально-технической базы (автоматизированная, цифровая).

Нередко библиотеки объединяются в административные (территориальные, отраслевые, ведомственные и иные), централизованные или в корпоративные электронные системы. Последние получают все большее развитие в связи с теми возможностями, которые предоставляют Интернет и подобные системы.

Библиотеки подразделяются также на типы. Характерный признак типа — соче­тание наиболее распространенных (типичных) параметров. Общепринято деление библиотек на два типа:

* массовые;
* научные и специальные.

Массовым библиотекам присущи: ориентация на неопределенно широкий кон­тингент пользователей, на их преимущественно общекультурные, общеобразова­тельные запросы и соответственно — преимущественно универсальный по содержа­нию состав фондов научно-популярной и художественной литературы, относительно быстро сменяемый; общедоступность; территориальный принцип размещения. Мас­совые библиотеки составляют приблизительно одну треть, а их фонды — около 40 процентов совокупного фонда библиотек общественного пользования.

Научным и специальным библиотекам свойственны: ориентация на ученых и специалистов, а также готовящихся быть таковыми, (библиотеки учебных заведе­ний); фонды преимущественно научной, производственной, учебной литературы; производственный принцип размещения; ограниченная специальными условиями доступность. По содержанию фондов эти библиотеки могут быть узкоспециальны­ми, многоотраслевыми и универсальными.

### Постановка задачи

Программное обеспечение является одной из важных составляющих автоматизированных библиотечных систем. Без программных средств техническое обеспечение будет бесполезным, а базы данных, электронные каталоги невозможно будет создать, а следовательно, и производить поиск нужной информации.

В условиях автоматизации информационно-библиотечного обслуживания решаются следующие задачи:

* происходит систематизация ресурсов на разных носителях, а также создание специальных электронных систем их обработки, учета, поиска.
* в практику обслуживания внедряются современные поисковые системы, обеспечивающие оперативность и удобство при поиске информации.
* информационно-библиотечные процессы автоматизируются, что приводит к экономии ресурсов и повышению эффективности обслуживания.
* спектр услуг, предоставляемых библиотекой, расширяется в соответствии с потребностями пользователей.
* библиотека получает возможность повышать информационную культуру, пользователей.
* применение компьютерных технологий вооружает библиотеку новыми методами и формами продвижения книги и чтения.

Необходимость создания АБИС существует для упрощения работы сотрудников библиотеки.

### Функции проектируемого приложения

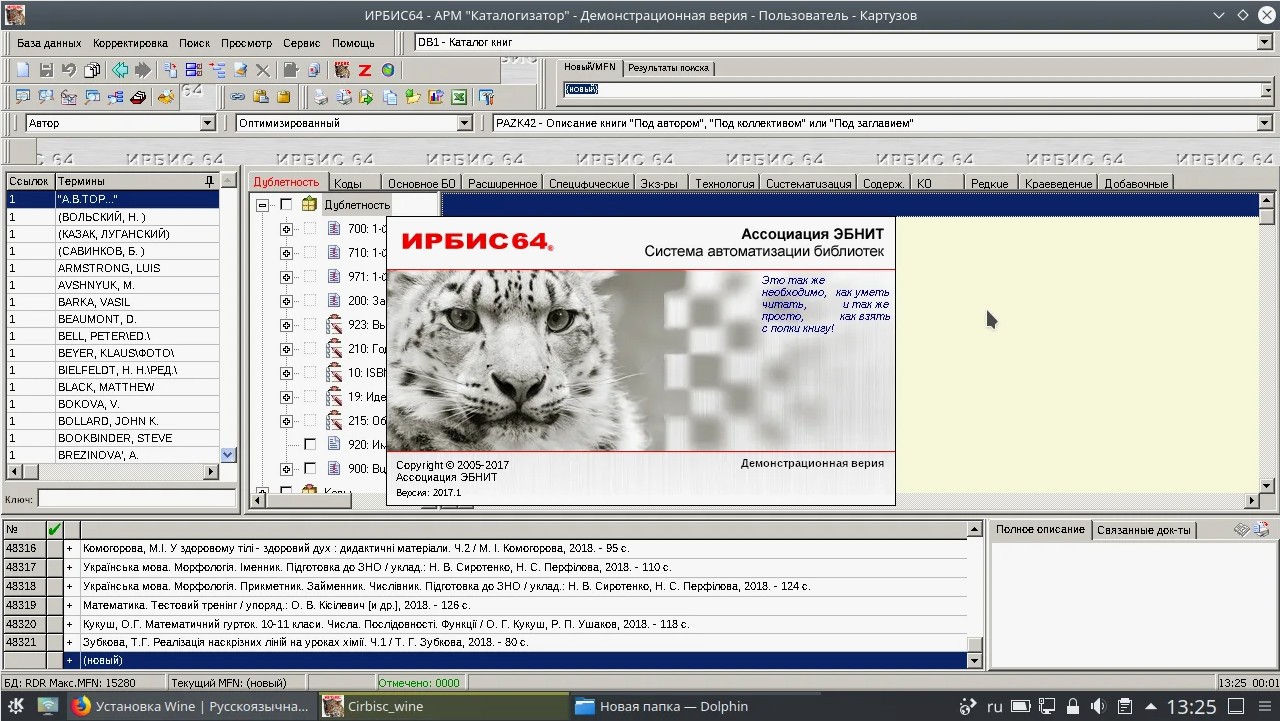
Программное обеспечение АБИС в первую очередь должно реализовать следующие функции:

* обработку, хранение библиографической и фактографической информации, ее поиск по любым элементам записей и их сочетаниям;
* подготовка подборок материалов в виде списков, фактографических и библиографических записей, отсортированным по индексам какой-либо классификации и алфавиту;
* вывод данных о занятости экземпляра издания, осуществление заказа единицы хранения читателем непосредственно при работе с ЭК с автоматическим формированием читательского требования;
* фиксацию выдачи и возврата единиц хранения с параллельным изменением данных об их местонахождении и сведений в читательском формуляре о полученных документах;
* поддержку использования при поиске нормативных записей об индивидуальных авторах и коллективах;
* наличие справочных текстов для ситуаций, которые могут вызвать затруднения у пользователя;
* ввод записей о заказываемых документах и учет поступления их в библиотеку;
* выдачу результатов поиска на экран в принятой форме;
* загрузку данных из текстовых файлов и выгрузку записей из базы данных в текстовые файлы;
* защиту базы данных от несанкционированного доступа;
* восстановление базы данных в случае аварийной ситуации.

### Анализ рынка существующих решений

На российском рынке представлен довольно широкий выбор отечественных разработок программных комплексов. Ведущий разработчик библиотечных систем – ГИВЦ (Главный информационно-вычислительный центр), выполнивший такие разработки АБИС, как «АС-Библиотека», «АБИС-Библиофил», «Библиотека для слепых». Также этим делом занимается БЕН РАН («SOLAR»), ГПНТБ России («ИРБИС» и др.). Комплексы программных средств состоят из модулей ПО (программного обеспечения) автоматизированных рабочих мест (АРМ). То есть создаются АРМы, которые обеспечивают выполнение следующих функций: комплектование, обработка, обслуживание абонемента и др., в зависимости от выбранного комплекса программных средств.

Пример GUI АБИС «ИРБИС64» представлен на рисунке 1.



1. Пример GUI АБИС «ИРБИС64»

В таблице 1 представлено сравнение РитаSQL, ИРБИС64 и РУСЛАН.

Таблица 1 – Таблица сравнения систем РитаSQL, ИРБИС64 и РУСЛАН.

| **Отличия** | **РитаSQL** | **ИРБИС64** | **РУСЛАН** |
| --- | --- | --- | --- |
| Платформа | Windows 10 и выше | Windows 2000/XP и выше | Linux |
| Простая настройка | + | + | - |
| Низкие требования к ресурсам | + | + | - |
| Поддержка облачных технологий | - | - | + |
| Модульная структура | + | + | + |
| Предназначена для крупных библиотек | - | - | + |
| Поддерживает автоматизацию всех основных этапов обработки единиц хранения фондов | - | - | + |
| Возможность перевода всей библиографической информации с других АБИС | - | - | + |

### Архитектура приложения

Систему баз данных можно рассматривать как систему, где осуществлено распределение процесса выполнения по принципу взаимодействия двух программных процессов, один из которых в этой модели называется "клиентом", а другой, обслуживающий клиента, — сервером (машина, хранящая базы данных). Клиентский процесс запрашивает некоторые услуги, а серверный процесс обеспечивает их выполнение. При этом предполагается, что один серверный процесс может обслужить множество клиентских процессов.

Сервер в простейшем случае — это собственно СУБД. Он поддерживает все основные функции СУБД и предоставляет полную поддержку на внешнем, концептуальном и внутреннем уровнях. Клиенты — это различные приложения, которые выполняются над СУБД.

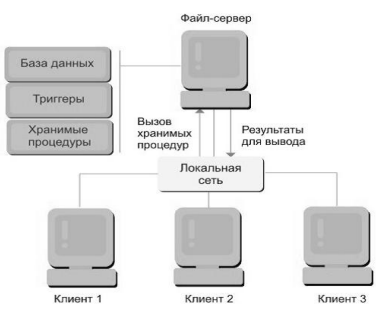
Будет использована модель сервера баз данных.

Технологию "клиент — сервер" поддерживают большинство современных СУБД: Informix, Ingres, Sybase, Oracle, MS SQL Server. В основу данной модели добавлен механизм хранимых процедур и механизм триггеров.

Механизм хранимых процедур позволяет создавать подпрограммы, работающие на сервере и управляющие его процессами. Таким образом, размещение на сервере хранимых процедур означает, что прикладные функции приложения разделены между клиентом и сервером. Трафик обмена информацией между клиентом и сервером резко уменьшается.

Централизованный контроль целостности базы данных в модели сервера баз данных выполняется с использованием механизма триггеров. Триггеры также являются частью БД. Триггер — это особый тип хранимой процедуры, реагирующий на возникновение определенного события в БД. Он активизируется при попытке изменения данных — при операциях добавления, обновления и удаления. Триггеры определяются для конкретных таблиц БД. Внедрение триггеров незначительно влияет на производительность сервера и часто используется для усиления приложений, выполняющих многокаскадные операции в БД.

В данной модели (рисунок 2) сервер является активным, потому что не только клиент, но и сам сервер, используя механизм триггеров, может быть инициатором обработки данных в БД. Поскольку функции клиента облегчены переносом части прикладных функций на сервер, он в этом случае называется "тонким". При всех положительных качествах данной модели у нее все же есть один недостаток — очень большая загрузка сервера.



1. Модель сервера БД

### Обоснование и выбор методики, технологии и инструментальных средств проектирования и разработки

#### Обоснование выбора используемой СУБД

Для реализации базы данных была выбрана система управления базами данных Microsoft (MS) SQL Server.

В таблице 2 представлено сравнение MS SQL Server, PostgreSQL, MongoDB, Oracle и MySQL.

Таблица 2 – Таблица сравнения СУБД MS SQL Server, PostgreSQL, MongoDB, Oracle и MySQL.

| **Отличия** | **MS SQL Server** | **PostgreSQL** | **MongoDB** | **Oracle** | **MySQL** |
| --- | --- | --- | --- | --- | --- |
| Высокая надежность | + | + | + | + | - |
| Взаимодействие с продуктами MS | + | - | - | - | - |
| Низкие затраты ресурсов | ± | - | + | - | + |
| Хорошая документация | + | - | + | + | + |
| Поддерживает набор пользовательских интерфейсов | + | + | - |  | + |
| Может работать с другими базами данных | + | - | - | + | + |
| Скорость и простота в использовании | + | + | + | - | + |

В отличие от всех остальных наиболее известных СУБД, MS SQL Server имеет ряд преимуществ. Главным из них является то, что он обеспечивает создание информационных систем с архитектурой «клиент-сервер», в которой он играет роль сервера баз данных. Одним из важных свойств Microsoft SQL Server является возможность управления целостностью данных.

Также можно сказать, что он удовлетворяет требованиям, предъявляемым к системам распределенной обработки информации. Эта СУБД поддерживает: тиражирование данных, параллельную обработку, создание и обработку больших баз данных на недорогих аппаратных платформах, отличается простотой управления и использования.

К достоинствам данной СУБД можно отнести:

* продукт очень прост в использовании;
* текущая версия работает быстро и стабильно;
* движок предоставляет возможность регулировать и отслеживать уровни производительности, которые помогают снизить использование ресурсов;
* доступ к визуализации на мобильных устройствах;
* очень хорошо взаимодействует с другими продуктами Microsoft.

#### Обоснование выбора языка программирования

Был выбран язык программирования C# (си-шарп).

В таблице 2 представлено сравнение языков программирования С#, С++, Java, Python и R.

Таблица 2 – Таблица сравнения языков программирования С#, С++, Java, Python и R.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Отличия** | **С#** | **С++** | **Java** | **Python** | **R** |
| Встроенные библиотеки для работы с данными | - | - | - | + | + |
| Высокая производительность | + | + | + | - | - |
| Бесплатная лицензия | + | + | + | + | + |
| Простота в изучении | + | - | - | + | + |
| Универсальность | + | + | + | + | - |
| Строгая типизация | + | + | + | - | - |
| Эффективность для работы с данными | + | - | - | + | + |

Причины выбора языка C#:

* C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности - как FCL, так и CLR;
* C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;
* C# является мощным объектным языком с возможностями наследования и универсализации;
* C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от С++ к C#;
* Сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства С++ как указатели, адресация, разыменование, адресная арифметика;
* Мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;
* Реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

#### Обоснование выбора используемой среды разработки

Для разработки пользовательского приложения было выбрано инструментальное ПО «Microsoft Visual Studio».

В таблице 3 представлено сравнение сред разработки Microsoft Visual Studio, KDevelopи Gambas.

Таблица 3 – Таблица сравнения сред разработки Microsoft Visual Studio, KDevelopи Gambas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Отличия** | **MS Visual Studio** | **KDevelop** | **Gambas** |
| Совместимость с Windows | + | + | - |
| Совместимость с Linux | + | + | + |
| Совместимость с macOS | - | + | - |
| Язык, на котором написана среда разработки | C ++/C# | C/C ++ | BASIC |
| Поддерживает C# | + | + | - |
| Возможность простой разработки Windows Forms | + | - | - |

Причины выбора Microsoft Visual Studio:

* Разработка будет производиться на ПК с операционной системой Windows, поэтому важно, чтобы ПО было с ней совместимо;
* Клиентское приложение ИС будет написано на C#, поэтому было необходимо выбрать среду разработки с поддержкой этого языка программирования;
* Графический интерфейс клиентского приложения ИС будет формой Windows Forms, поэтому было необходимо выбрать среду разработки, где создать форму легко.

## Специальная часть

### Диаграмма прецедентов использования

### Инфологическая модель

### Даталогическая модель

### Разработка пользовательского интерфейса

### Исключительные ситуации

### Разработанные запросы

### Обеспечение целостности базы данных

### Разграничение прав доступа. Защита данных

### Диаграмма классов

# ЗАКЛЮЧЕНИЕ