

LightSearch PC

[06.09.2018]

Производственную практику от университета я проходил в компании «Баярд» — сеть супермаркетов строительных и отделочных материалов города Хабаровска. Мне в «Баярде» дали задание:

«Написать кросс-платформенный клиент (далее Клиент), позволяющий получать оперативную информацию с базы данных (далее БД).

Параметры подключения к БД должны указываться в ini-файле. Разработать формат ini-файла и создать механизм подключения Клиента по параметрам, указанным в ini-файле.

Исходные данные: БД (Firebird 1.5.4), ОС сервера Linux

В клиенте создать форму диалога с пользователем, в которой нужно будет указать параметр товара для поиска (идентификатор или штрих-код или часть наименования), подразделение (склад или торговый комплекс или по всем). Создать запрос на основании полученных параметров и вывести информацию в форму. На форме в верхней части должны отображаться введенные пользователем параметры и в табличной части результаты запроса (подразделение, идентификатор товара, текущая цена прайса, количество товара в подразделении).»

Это задание я начал с вопроса о запросе. В итоге мы договорились на том, что администратор базы данных напишет мне запросы сам, так как база очень большая (60 гб информации), в ней довольно много таблиц, и я скорее всего на протяжении всей своей практики (которая длилась чуть меньше месяца) изучал бы структуру базы и придумывал бы оптимизированный запрос, необходимый для выполнения моего задания. Поэтому я сразу приступил к разработке интерфейса.

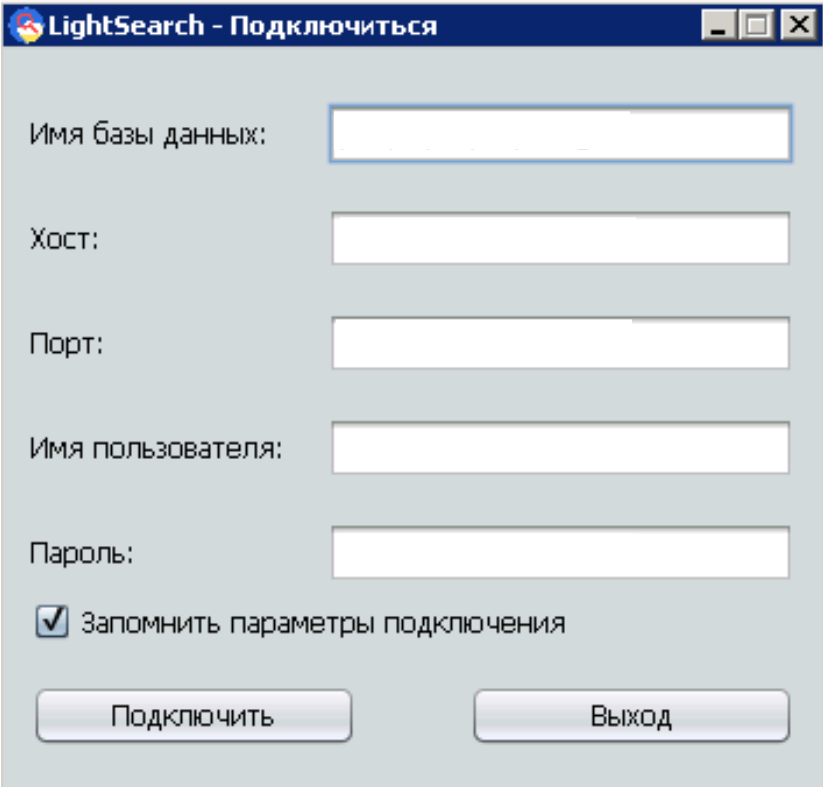
В качестве языка программирования я выбрал Java — ведь основное его преимущество в кросс-платформенности, основное его требование — чтобы на целевой машине была установлена JVM. Ну и к тому же я стал изучать этот язык несколько месяцев назад и мне хотелось написать на нем что-то полезное. Итак, для разработки приложения я установил:

- 1) JDK версии 1.8 — для разработки одной JVM недостаточно;
- 2) NetBeans IDE 8.2 — т. к. в ней я работал ранее, то уже довольно изучил ее и мне в ней комфортно работать;
- 3) Jaybird — 2.2.14 — именно такая версия JDBC драйвера Firebird мне и нужна (далее я напишу, почему именно эту версию я использовал для своего проекта);
- 4) IBExpert — если за меня и написали запрос, то это не значит, что я не должен понимать, как он работает. Мне объяснили, почему именно так написан запрос и какие таблицы он использует. Поэтому мне нужна данная программа для получения схемы таблиц, проверки работы запроса, и в будущем мне

пригодилась для небольшой модификации запроса и проверки его отдельных частей.

Такой набор достаточен для того, чтобы начать разработку. После установки и настройки всех этих программ я приступил к первому пункту — написание интерфейса программы.

Для написания интерфейса я использовал библиотеку Swing. В совокупности с NetBeans это делается приятно и легко. После нескольких часов получились две формы: первая — форма авторизации, которая потом в дальнейшем не модифицировалась:



The image shows a Java Swing window titled "LightSearch - Подключиться". The window has a light gray background and a blue title bar. It contains the following elements:

- Five text input fields arranged vertically, each preceded by a label: "Имя базы данных:", "Хост:", "Порт:", "Имя пользователя:", and "Пароль:". The first field is currently selected with a blue border.
- A checkbox labeled "Запомнить параметры подключения" (Remember connection parameters) which is checked.
- Two buttons at the bottom: "Подключить" (Connect) on the left and "Выход" (Exit) on the right.

И вторая форма, но это уже конечный ее вид, первоначальный я не сохранил:

The screenshot shows the 'LightSearch' application window. The title bar includes the application name and standard window controls. The main interface is divided into several sections:

- Search Section:** Includes a 'Параметр товара:' (Product parameter) text label, a search input field, and two checkboxes: 'Режим сканнера' (Scanner mode) and 'Автоматический поиск' (Automatic search).
- Filter Section:** Contains 'Подразделение:' (Department) with radio buttons for 'Склад' (Warehouse), 'Торговый комплекс' (Shopping complex), and 'Все' (All); 'Выберите склад:' (Select warehouse) and 'Выберите ТК:' (Select shopping complex) dropdown menus; and a 'Поиск' (Search) button.
- Table Section:** A table titled 'Таблица товаров' (Goods table) with the following columns: 'Подразделение' (Department), 'Идентификатор' (Identifier), 'Наименование' (Name), 'Текущая цена' (Current price), and 'Кол-во в подразделении' (Quantity in department).
- Footer:** A message 'Добро пожаловать!' (Welcome!).

И вот, когда я проработал логику подключения к бд, реализовал ее и подключил драйвер Jaybird, то я столкнулся с первыми «граблями» — Jaybird успешно подключился, но выскакивали постоянно неизвестные ошибки. Потом я зашел в один из форумов и прочитал, что оказывается версия Jaybird 3.0.4 не работает с FireBird 1.5.4, а работает с версией, начиная с 2.2. Поэтому мне и пришлось скачать драйвер меньшей версии — 2.2.14. Он заработал как часы, но есть один неприятный баг — почему-то к данным в конце добавляются пробелы, когда идет их выгрузка с бд, это были вторые «грабли» — потому что теперь нельзя было ставить ширину столбцов по размеру содержимого, это выглядело очень жутко. Поэтому пришлось написать метод, который рассчитывает оптимальную ширину столбца.

Потом я повозился вот с этой частью:

This is a partial screenshot of the 'LightSearch' application window, showing the top portion of the interface. It includes the 'Параметр товара:' (Product parameter) section with a search input field and checkboxes for 'Режим сканнера' (Scanner mode) and 'Автоматический поиск' (Automatic search). Below this is the 'Подразделение:' (Department) section with radio buttons for 'Склад' (Warehouse), 'Торговый комплекс' (Shopping complex), and 'Все' (All), followed by 'Выберите склад:' (Select warehouse) and 'Выберите ТК:' (Select shopping complex) dropdown menus, and a 'Поиск' (Search) button.

Надо было реализовать следующую функцию: сделать поиск только по Складам, только по Торговым комплексам, или по всем подразделениям одновременно. Я ее сделал следующим образом: подразделение выбирается через радио-кнопку:

Подразделение:

☒ Склад ☐ Торговый комплекс ☐ Все

При выборе определенного подразделения активируется соответствующий выпадающий список:

Выберите склад:

Выберите ТК:

<Все>

<Все>

Но если кликнуть на радио-кнопку «Все», то блокируются все выпадающие списки, ведь не надо выбирать, какой необходим склад или ТК. Этот блок мне дался довольно гладко, без происшествий.

Теперь про строку поиска:

Параметр товара:

Вот тут я столкнулся с проблемой. Дело в том, что запрос, написанный администратором базы данных, работал корректно, но почему-то Jaybird данной версии не понимает trim и не понимает upper. Поэтому эти функции пришлось делать в самой программе, благо что у Java уже есть такие функции по умолчанию.

Запрос был написан так, что строка поиска принимает все что угодно — часть наименования, полное наименование, или штрих-код товара. Конечно же, перед отправкой строки в запрос, она проходит проверку в виде последовательно расположенных функций trim — это необходимо для вырезания лишних символов у таких строчек, как «___~~~@@!!гвоздь##&&», и « ». В первом случае строка преобразуется в «гвоздь», а во втором случае программа просто не даст нажать кнопку «Поиск» — она будет заблокирована, ведь кнопка блокируется, если в строке введено меньше трех символов.

Программа работала, все показывалось, и мой руководитель практики сказал, что нужно еще реализовать следующую функцию — считывать штрих-код с ценника при помощи сканера:



Он подключается через USB. Я подключил его к ноутбуку, взял лист с ценниками, нажал кнопку — он пиликнул. В первый раз я не поставил курсор в поле «Поиск», и поэтому не получил результата. Но зато курсор был поставлен внутри программы на какой-то строчке, и когда я увидел какие-то цифры, то до меня дошло, как он работает: он просто перехватывает нажатие клавиатуры и имитирует нажатие клавиш. Тогда я стал думать, как мне теперь сделать работу со сканером удобной: ведь нажимать каждый раз на Enter или на кнопку «Поиск» не так уж и удобно. Поэтому я добавил галку «Автоматический поиск», при активации которой при вводе более трех символов через приблизительно секунду поиск начинается сам:

☐ Режим сканнера ☐ Автоматический поиск

Выберите склад: <Все> ▼ Выберите ТК: <Все> ▼ Поиск

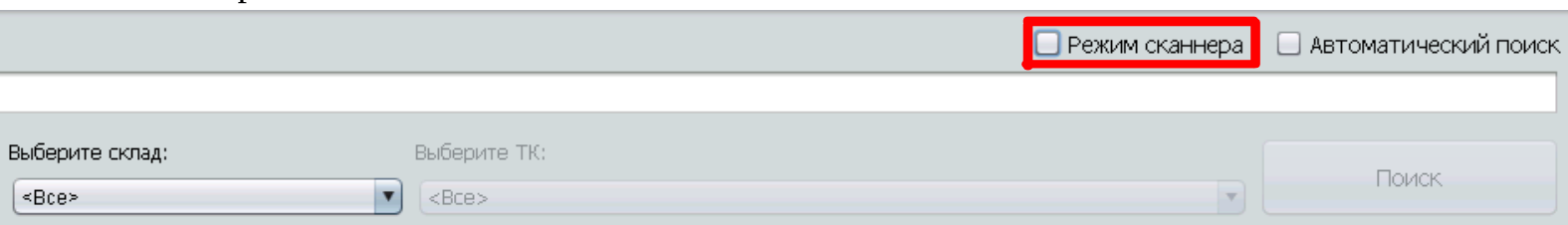
Работает она так: при выборе данной функции активируется таймер, который каждую чуть менее секунды времени проверяет: не менее ли трех символов написано в строке поиска. Если да, то он отправляет уже заранее отфильтрованное содержимое строки поиска в запрос.

Теперь нажимать на Enter и на кнопку поиска совсем не обязательно.

После этого я снова попробовал попользоваться сканером, и... сработало! Но после второго считывания в строке получил примерно такое:

13434768747767823. У меня не очистился результат после первого поиска в строке, и тогда я решил добавить очищение строки поиска перед отправкой строки на запрос в «Автоматический поиск», но понял, что идея неправильна, ведь эта функция рассчитана не только на сканер, но и на человека. Я просто представил, что если человек включит эту функцию и будет вводить что-то в

строке поиска, то у него будет меньше чем за секунду стираться то, что он написал, а это нехорошо. Поэтому я решил создать новую функцию — Режим сканера:

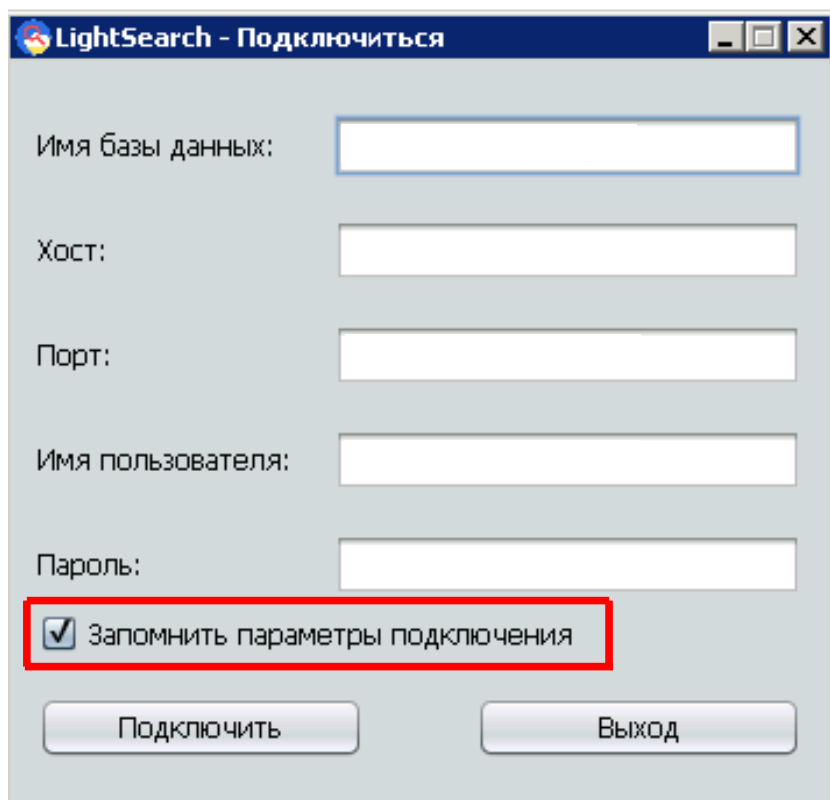


Она работает следующим образом: при ее выборе при нажатии на кнопку поиска или на Enter будет очищаться строка поиска.

Для того, чтобы «Режим сканера» корректно работал с «Автоматическим поиском», необходимо было доработать таймер: теперь еще и таймер проверяет, включена ли функция «Режим сканера»: если включена и если введено не менее трех символов, то идет отправка отфильтрованного содержимого строки поиска в запрос, и затем очищается строка поиска.

И теперь можно пользоваться отдельно автоматическим поиском и режимом сканера. Конечно же пользоваться сканером намного приятнее с функцией «Автоматический поиск». После тестов со сканером я немного откалибровал время таймера. Теперь функции работают корректно.

Теперь мне осталось сделать ini-файл для функции запоминания параметров подключения:



Я не стал особо думать и просто сделал запись параметров по порядку, разделяя их точкой с запятой. Теперь необходимо было реализовать шифрование. Я не стал придумывать «велосипед» и использовал уже готовое решение в виде класса, который реализует метод шифрования DES. ini-файл я сохраняю в папке

с проектом, а ключ храню в system32, ведь по сути здесь не нужна особо сильная защита. Но конечно я исправлю ситуацию с system32, ведь теперь клиент не становится полноценно кросс-платформенным. Но для начала сойдет и такой вариант.

Я показал программу своему руководителю практики и он мне сказал, что было бы неплохо добавить возможность при нажатии на сочетание клавиш выравнивать размер столбцов по размеру таблицы. Я добавил такую функцию: теперь при нажатии на ctrl+R столбцы выравниваются по таблице.
До выравнивания:

Таблица товаров

Подразделение	Идентификатор	Наименование
Склад Комплекс 3 (Переулок Трубны...	888457	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 200 золото
Склад Комплекс 4 (ВыбОР - ул.Мороз...	888457	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 200 золото
Склад Комплекс 6 (Проспект 60-лет ...	888457	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 200 золото
Склад Комплекс 3 (Переулок Трубны...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 200 хром
Склад Комплекс 4 (ВыбОР - ул.Мороз...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 200 хром
Склад Комплекс 5 (Волочаевская 87) ...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 200 хром
Склад Комплекс 6 (Проспект 60-лет ...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 200 хром
Склад Комплекс 3 (Переулок Трубны...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP хром
Склад Комплекс 4 (ВыбОР - ул.Мороз...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP хром
Склад Комплекс 5 (Волочаевская 87) ...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP хром
Склад Комплекс 6 (Проспект 60-лет ...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP хром
Склад Комплекс 3 (Переулок Трубны...	1011205	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика CP хром
Склад Комплекс 4 (ВыбОР - ул.Мороз...	1011205	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика CP хром
Склад Комплекс 6 (Проспект 60-лет ...	1011205	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика CP хром
Склад Комплекс 3 (Переулок Трубны...	1011212	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика GP золото
Склад Комплекс 4 (ВыбОР - ул.Мороз...	1011212	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика GP золото
Склад Комплекс 6 (Проспект 60-лет ...	1011212	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика GP золото
Склад Комплекс 6 (Проспект 60-лет ...	1018747	... Глазок дверной лат. Н-М №12/22/200/40-60 стекл.оптика золото
Склад Комплекс 3 (Переулок Трубны...	1018761	... Глазок дверной Н-М №12/22/200/40-60 пластик. оптика черный
Склад Комплекс 4 (ВыбОР - ул.Мороз...	1018761	... Глазок дверной Н-М №12/22/200/40-60 пластик. оптика черный

Количество записей: 31

После выравнивания:

Таблица товаров

Подразделение	Идентификатор	Наименование	Текущая цена, руб	Кол-во в подразделении
Склад Комплекс 3 ...	888457	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 2...	204	5
Склад Комплекс 4 ...	888457	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 2...	204	4
Склад Комплекс 6 ...	888457	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 2...	204	3
Склад Комплекс 3 ...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 2...	204	5
Склад Комплекс 4 ...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 2...	204	7
Склад Комплекс 5 ...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 2...	204	2
Склад Комплекс 6 ...	888464	... Глазок дверной АЛЛЮР ГДШ-4 БШт 60-100 угол обзора 2...	204	5
Склад Комплекс 3 ...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP ...	194	4
Склад Комплекс 4 ...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP ...	194	2
Склад Комплекс 5 ...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP ...	194	2
Склад Комплекс 6 ...	1011199	... Глазок дверной DVZ3 16/200/50*90 пластик. оптика CP ...	194	4
Склад Комплекс 3 ...	1011205	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика CP ...	163	3
Склад Комплекс 4 ...	1011205	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика CP ...	163	7
Склад Комплекс 6 ...	1011205	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика CP ...	163	7
Склад Комплекс 3 ...	1011212	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика GP ...	163	4
Склад Комплекс 4 ...	1011212	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика GP ...	163	4
Склад Комплекс 6 ...	1011212	... Глазок дверной DVZ1 16/200/35*60 пластик. оптика GP ...	163	1
Склад Комплекс 6 ...	1018747	... Глазок дверной лат. Н-М №12/22/200/40-60 стекл.опти...	224	8
Склад Комплекс 3 ...	1018761	... Глазок дверной Н-М №12/22/200/40-60 пластик. оптика...	184	3
Склад Комплекс 4 ...	1018761	... Глазок дверной Н-М №12/22/200/40-60 пластик. оптика...	184	1
Склад Комплекс 6 ...	1018761	... Глазок дверной Н-М №12/22/200/40-60 пластик. оптика...	184	4

Количество записей: 31

Потом я сделал инсталлятор для своей программы. Сначала я сделал его только при помощи NetBeans — и результат мне не понравился: нельзя было сменить иконку, ставилась программа только в Appdata Roaming. Поэтому почитав в интернете об инсталлерах, я установил себе launch4j и Inno Setup Compiler. При помощи первой программы я сделал.WRAPPER, а при помощи второй непосредственно инсталлер.

Именно в момент создания инсталлятора я и придумал название своей программы — LightSearch. И придумал логотип для нее:



Теперь пользователь может при помощи инсталлятора выбрать любую удобную для него папку, в которую будет установлена программа, создать ярлык, и пользоваться программой, как и всеми другими — запускать файл с расширением .exe.

Пока что код я выкладывать не буду, т. к. в дальнейшем планируется переделывать данное приложение. Необходимо вытащить из него всю бизнес-логику для обеспечения универсальности и гибкости программы. И в это же время будет рассмотрен более серьезно вопрос об шифровании и учета операционной системы.

Сделал я этот проект быстрее, чем ожидал — примерно через 2 недели работы 5/2 по 8-9 часов в день. Поэтому я захотел сделать еще и клиента на Android, но об этом уже в другом документе :)

Сюда будет добавляться новая информация по мере развития проекта.