

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ**

**Кафедра комп'ютерних наук**

**КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА**

**на тему: «РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ  
ДЛЯ ОПТИМІЗАЦІЇ БУХГАЛТЕРСЬКОГО ОБЛІКУ НА  
ПІДПРИЄМСТВІ»**

Виконав: студент	<u>2</u> курсу, групи <u>8.1228</u>
спеціальності	<u>122 комп'ютерні науки</u>
освітньої програми	<u>комп'ютерні науки</u>
	(шифр і назва спеціальності)
	<u>М.В. Коцарев</u>
	(ініціали та прізвище)
Керівник	<u>доцент кафедри комп'ютерних наук, доцент, к.п.н., Пшенична О.С.</u>
	(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент	<u>завідувач кафедри програмної інженерії, доцент, к.ф-м.н. Лісняк А.О.</u>
	(посада, вчене звання, науковий ступінь, прізвище та ініціали)



## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 29.05.2019**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	10.06.19	
2.	Збір вихідних даних.	05.07.19	
3.	Обробка методичних та теоретичних джерел.	28.08.19	
4.	Розробка першого та другого розділу.	25.09.19	
5.	Розробка третього розділу.	16.10.19	
6.	Оформлення та нормоконтроль кваліфікаційної роботи.	16.12.19	
7.	Захист кваліфікаційної роботи.	14.01.20	

Студент

\_\_\_\_\_

(підпис)

М.В. Коцарєв

(ініціали та прізвище)

Керівник роботи

\_\_\_\_\_

(підпис)

О.С. Пшенична

(ініціали та прізвище)

**Нормоконтроль пройдено**

Нормоконтролер

\_\_\_\_\_

(підпис)

О.Г. Спиця

(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота магістра «Розробка програмного забезпечення для оптимізації бухгалтерського обліку на підприємстві»: 54 с., 14 рис., 1 табл., 7 джерел, 4 додатки.

АЛГОРИТМ, БАЗА ДАНИХ, ЗАПИТ, ЗВІТ, КОНФІГУРАТОР, МОДУЛЬ, ПЛАТФОРМА.

Об'єкт дослідження – бухгалтерський облік та шляхи його оптимізації.

Мета роботи: створення програмного забезпечення засобами 1С для оптимізації роботи бухгалтера.

Метод дослідження – аналітичний, оглядовий.

Для реалізації поставленої мети було вирішено такі завдання:

- визначено сучасні тенденції ведення бухгалтерського обліку та проаналізовано актуальність проблем оптимізації;
- розглянуто існуючі рішення для ведення обліку;
- проведено аналіз і вибір технологій для розробки програмного забезпечення;
- розроблено програмне забезпечення для звірки відповідності сум ПДВ, нарахованих при реалізації товарів і послуг і сум виписаних податкових накладних.

При послідовному рішенні поставлених завдань було розроблено програмне забезпечення для контролю правильності ведення податкового обліку на підприємстві в частині урахування ПДВ.

При виконанні роботи застосовано платформу «1С: Підприємство».

## SUMMARY

Master's Qualification Thesis «Software Development for Optimization of Accounting at the Company»: 54 pages, 14 figures, 1 table, 7 references, 4 supplements.

ALGORITHM, DATABASE, REQUEST, REPORT, CONFIGURATOR, MODULE, PLATFORM.

The object of the study is accounting and ways to optimize it.

The aim of the study is creating software by means of 1C for optimization of work of the accountant.

The methods of research are analytical and survey.

In order to achieve the goal, the following tasks were solved:

- defined current trends in accounting and analyze the relevance of optimization problems;
- considered existing accounting solutions;
- analysis and selection of technologies for development are conducted;
- developed software for verification of compliance of the VAT amounts accrued on the sale of goods and services and the amounts of tax invoices issued;

A coherent solution to the tasks set up, a software was developed to control the correctness of tax accounting at the company in terms of VAT accounting.

In execution of the work, the platform "1C: Enterprise" is used.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Вступ.....	8
1 Інформаційні технології в бухгалтерському обліку .....	9
1.1 Сучасні тенденції ведення обліку .....	9
1.2 Перспективи розвитку програм для ведення обліку .....	11
1.3 Огляд програм для реалізації бухгалтерського обліку .....	13
1.4 Застосування хмарних технологій в бухгалтерському обліку .....	17
1.5 Актуальність проблеми оптимізації.....	19
2 Аналіз і вибір технологій для оптимізації .....	21
2.1 Шляхи оптимізації .....	21
2.2 Технології та засоби використання .....	21
2.2.1 Вбудована мова програмування 1С .....	23
2.2.2 Мова запитів 1С .....	24
2.3 Файлова і серверна бази даних .....	25
2.4 Конфігуратор .....	28
2.5 Модульна система.....	30
2.6 Структура інтерфейсу програмного забезпечення .....	31
3 Принцип роботи та тестування створеного програмного забезпечення...32	
3.1 Принцип роботи .....	32
3.2 Тестування додатка .....	38
3.3 Оцінка користі на підприємстві .....	41
Висновки .....	43
Перелік посилань.....	44
Додаток А Код процедури формування звіту .....	45
Додаток Б Код процедури відкриття.....	48

Додаток В Код процедури заповнення податкових накладних.....	50
Додаток Г Код процедури друку .....	52

## ВСТУП

Питання оптимізації бухгалтерського обліку сьогодні є достатньо актуальним. Бухгалтерський облік забезпечує формування достовірної інформації про діяльність підприємств та їх майновий стан, що дозволяє керівництву відслідковувати і контролювати рівень витрат і доходів. Відповідно, достовірні й своєчасно надані бухгалтерські дані допомагають приймати вірні бізнес-рішення і знижувати витрати компанії.

На даний момент бухгалтерський облік є невід'ємною частиною функціонування будь-якого підприємства. І в сучасних економічних умовах вести бухгалтерський облік неможливо без використання комп'ютера і відповідного програмного забезпечення. В якості базових інформаційних технологій виступають стандартні бухгалтерські інформаційні технології програм типу «1С: Підприємство».

Під час роботи з автоматизованою бухгалтерською системою з'являється можливість швидко підготувати всі квартальні і річні звіти в податкову інспекцію, до різних фондів та органів, розрахувати зарплату, вести облік основних засобів, аналізувати фінансову діяльність.

Однак через комплексність систем обліку програмне забезпечення вимагає оптимізації і доопрацювань для створення повноцінного інструменту бухгалтера, який він зможе застосовувати для коректного відображення фінансового стану підприємства.



# 1 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В БУХГАЛТЕРСЬКОМУ ОБЛІКУ

## 1.1 Сучасні тенденції ведення обліку

Бухгалтерський облік – це процес визначення, вимірювання та надання економічної інформації про організацію або будь-яке підприємство з метою винесення обґрунтованих рішень користувачами цієї інформації.

Отже, ключовими аспектами бухгалтерського обліку є:

- визначення основних фінансових складових частин організації, таких, як активи, зобов'язання, власний капітал, доходи, витрати і рух грошових коштів;

- вимір грошової вартості основних фінансових складових таким чином, який дає достовірне і об'єктивне уявлення інформації у фінансовій звітності;

- надання фінансової інформації у спосіб, найбільш прийнятний для користувачів цієї інформації.

Система бухгалтерського обліку включає такі основні елементи, представлені на рис. 1.1.

Бухгалтерські інформаційні системи пройшли великий історичний шлях становлення й розвитку. Вони змінювалися паралельно зі змінами інформаційних технологій, програмних і апаратних засобів обробки інформації та трансформаціями систем обліку.

Існуючі тенденції розвитку обліку слід розділити на наукову і практичну діяльність. Основна тенденція розвитку бухгалтерського обліку як наукової діяльності полягає в тому, що в бухгалтерських наукових дослідженнях у зв'язку з формуванням постіндустріальної економіки використовується змішана модель, що передбачає цілісність припущень щодо того, «що вже існує», прогнозування на цій основі практики

бухгалтерського обліку, а також встановлення того, «що має бути» в бухгалтерському обліку.

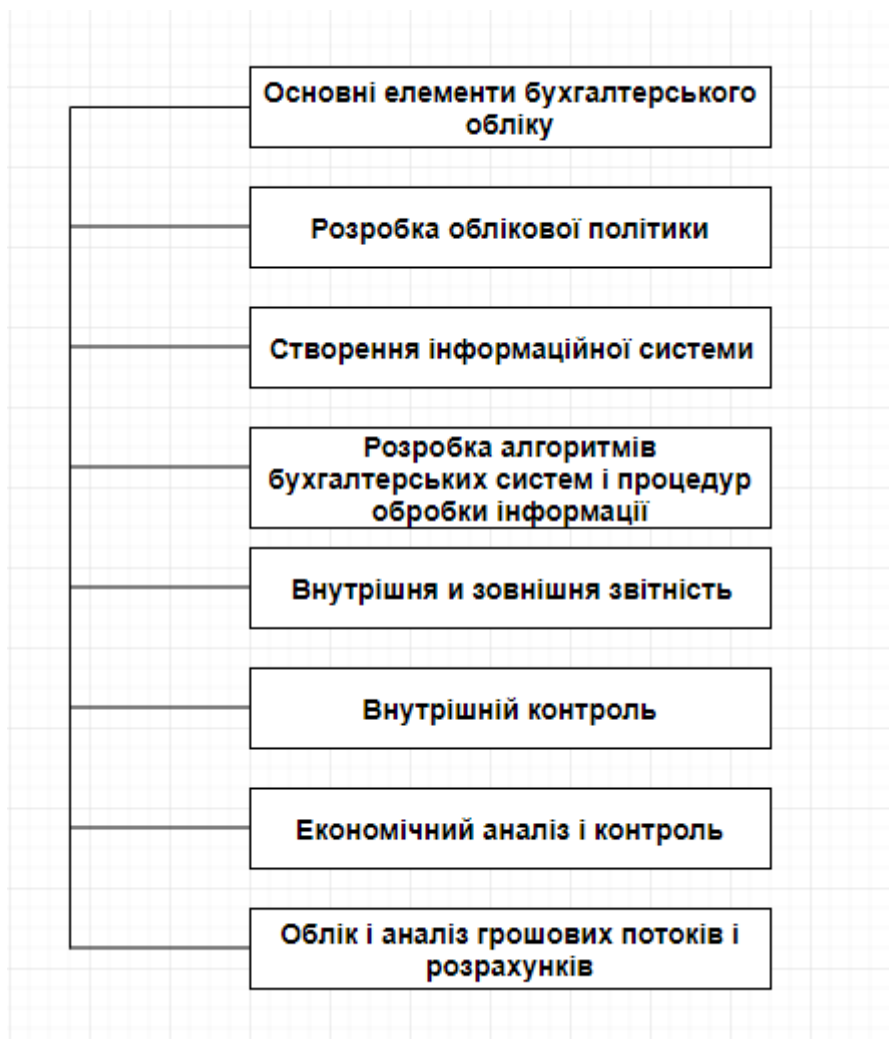


Рисунок 1.1 – Система бухгалтерського обліку

Необхідність здійснення комплексних заходів щодо вдосконалення діючої системи бухгалтерського обліку та звітності викликана насамперед зміною економічної середовища діяльності господарюючих суб'єктів, розширенням взаємовідносин з суб'єктами світового ринку, появою нових об'єктів підприємницької діяльності, змінами правової бази, тобто з його нормативним регулюванням.

Розвиток системи регулювання бухгалтерського обліку та звітності направлено на побудову такої моделі, яка враховувала б інтереси всіх зацікавлених сторін, забезпечувала зниження витрат і підвищення

ефективності управління. В основі цієї моделі лежить розумне поєднання діяльності органів державної влади та професійного співтовариства (професійних громадських об'єднань та зацікавленої громадськості).

Розподіл функцій між органами державної влади та професійним співтовариством має виходити з такого. Забезпечуючи реалізацію конституційного права на інформацію в сфері економіки і підприємницької діяльності, бухгалтерський облік і звітність є однією з гарантій функціонування єдиного ринку і забезпечення єдності економічного простору в країні. Регулювання бухгалтерського обліку і звітності має забезпечувати їх єдність і цілеспрямованість, а також спадкоємність розвитку з урахуванням інтересів широкого кола зацікавлених користувачів.

## **1.2 Перспективи розвитку програм для ведення обліку**

Головними перевагами використання інформаційних систем в бухгалтерському обліку є автоматизація, яка дає змогу швидко та якісно обробляти й передавати інформацію, надійно зберігати її та забезпечувати швидкий зовнішній і внутрішній її обіг.

На сьогоднішній день можна виділити такий перелік напрямків розвитку програмного забезпечення:

а) удосконалення користувальницького інтерфейсу. Налагодження комфортної роботи користувача з програмою і реалізація цього у відповідному програмному середовищі. Інтерфейс має визначати вид, розмір, і розташування основного екрану, функції обробки, доступні через систему меню, панелі інструментів та ін. Програмні продукти повинні гарантувати надійну і безпечну роботу як для комп'ютера, так і для інформаційної системи користувача, забезпечуючи збереження пристроїв комп'ютера, програмного забезпечення і даних;

б) створення умов для автоматизації діяльності фахівців. Гарантування того, що автоматизовані робочі місця будуть повністю підтримувати професійну діяльність кінцевого користувача. АРМи також мають забезпечувати виконання і допоміжних, сервісних функцій, таких як копіювання, відновлення, експорт-імпорт даних;

в) створення інструментальних засобів кінцевого користувача. Забезпечення вдосконалення функцій обробки, створення нових додатків кінцевим користувачем. Для програм бухгалтерського обліку це мають бути такі інструменти:

- генератор екранних форм – дозвіл на створення нових і скасування існуючих екранних форм (розташування на екрані реквізитів, відповідних полях бази даних, використання тексту підказок, кольорове оформлення і ін.);

- мова запитів – забезпечення пошуку і фільтрації записів бази даних, вибірки машинних документів, обчислення над даними бази даних;

- макропрограмування – підключення клавішних і мовних макрокоманд (макросів), призначених для автоматизації рутинних операцій обробки;

- генератор звітів – забезпечення виведення запитової інформації, формування різного рівня підсумків і іншого;

- інтегровані пакети – набори декількох програмних продуктів, функціонально доповнюють один одного і підтримують єдині інформаційні технології.

г) розширення функціональних можливостей програм. Надання можливості розширення сфер дії програм. Розробники програмного забезпечення повинні мати вибір між:

- розширенням можливостей програм за рахунок вдосконалення засобів організації аналітичного обліку і їх налаштування, спробами вийти за межі завдань обліку, розширивши сферу їх застосування. В результаті чого

користувач має отримувати можливість вирішувати завдання оперативного обліку та аналізу;

– розробкою не окремих компонентів автоматизації, а створенням комплексних інформаційних систем масштабу підприємства або корпорації і не обмежуватися тільки рішенням завдань бухгалтерського і оперативного обліку.

### **1.3 Огляд програм для реалізації бухгалтерського обліку**

Існує безліч програм для автоматизованого ведення бухгалтерського обліку, як платних, так і безкоштовних. Розглянемо їх на прикладі наступних програм:

**«1С: Підприємство»** (рис. 1.2) є безперечним лідером в сфері обліку. В основі більш 90% всіх бізнесів України лежить платформа 1С. Вона включає в себе платформу і прикладні рішення, розроблені на її основі, для автоматизації діяльності організацій і приватних осіб.

Практично жодна компанія зараз не може повноцінно функціонувати без 1С. Роль програми в різних бізнес-процесах дуже важлива, адже з її допомогою здійснюється управління підприємством, а для керівництва різних рівнів вона виступає в якості помічника при прийнятті рішень.

Головні переваги програми 1С полягають в її можливості добре організувати і автоматизувати роботу бухгалтерської служби, від чого залежить стабільна і прибуткова робота компанії. А цією частиною ведення бізнесу точно не варто нехтувати.



Рисунок 1.2 – Платформа 1С

Гнучкість платформи дозволяє застосовувати 1С:Підприємство в найрізноманітніших областях:

- автоматизація виробничих і торгових підприємств, бюджетних і фінансових організацій, підприємств сфери обслуговування та ін .;
- підтримка оперативного управління підприємством;
- автоматизація організаційної і господарської діяльності;
- ведення бухгалтерського обліку з декількома планами рахунків і довільними вимірами обліку, регламентована звітність;
- широкі можливості для управлінського обліку і побудови аналітичної звітності, підтримка багатовалютного обліку;
- рішення задач планування, бюджетування і фінансового аналізу;
- розрахунок зарплати і управління персоналом та інші області застосування.

**Bookkeeper** (рис. 1.3) – сучасна онлайн бухгалтерія, що дозволяє вести облік діяльності, складати та подавати звітність підприємств, ФОП всіх систем оподаткування та неприбуткових організацій з будь-якого пристрою.



Рисунок 1.3 – Bookkeeper

У сервісі для ведення бухгалтерії онлайн реалізований функціонал для вирішення актуальних завдань малого і середнього бізнесу, а також некомерційних підприємств. За допомогою віддаленої бухгалтерії кожен користувач може самостійно, без допомоги фахівців з обліку, складати необхідні операції і розрахунки, формувати регламентовані звіти і здавати електронну звітність.

Присутні операції обліку: продаж і придбання, запаси і виробництво, каса і банк, розрахунки по зарплаті, необоротні активи.

Передбачена можливість ведення обліку декількох юридичних осіб в одній базі бухгалтерської програми.

«М.Е.Дос» (рис. 1.4) – сучасне рішення, яке приносить бізнесу багато користі, але при цьому не вимагає великих вкладень грошей і часу.



Рисунок 1.4 – М.Е.Дос

Перелік основних можливостей програми:

- швидке створення звітів;
- реєстрація додатків 2, накладних для податкової;
- миттєвий обмін квитанціями, додатками, звітами і так далі;
- подача документів в органи контролю (ПФУ, Держкомстат та ін.);
- перевірка вхідних і вихідних документів, визначення помилок і розбіжностей;
- оповіщення про отримання документів партнерами та контролюючими органами, підтримка зв'язку з допомогою текстових повідомлень;
- ведення електронного архіву, захищеного надійною системою безпеки;
- накладення електронного цифрового підпису.



Доступ до програми і її функцій можливий в абсолютно будь-якому місці, де є Інтернет. Вся інформація захищена від сторонніх осіб.

#### **1.4 Застосування хмарних технологій в бухгалтерському обліку**

Внаслідок того, що технічний прогрес рухається з величезною швидкістю, з'явилася можливість управляти діяльністю підприємства, перебуваючи в будь-якій точці планети.

Напевно головною відмінністю хмарного сервісу від інших технологій є те, що користувачу не треба встановлювати програмне забезпечення на свій девайс, адже він працює з тим, що встановлено на сервері провайдера. Всі дані зберігаються на сервері в дата-центрі. Клієнт оплачує тільки оренду програмного забезпечення, а технічне обслуговування сервісів здійснює компанія, що надає послуги. Тому підприємство може заощадити на найманні людей, завдання яких полягатиме в створенні локальної мережі та встановленні програмного забезпечення на потужних комп'ютерах, адже всі основні обчислювальні процеси відбуваються на сервері в інтернеті і мобільні пристрої, які мають малу обчислювальну потужність, легко справляються з поставленими завданнями. Це дуже зручно для людей, які працюють віддалено і передають всю інформацію в компанію через інтернет.

Бухгалтерський облік також можна перенести в хмару. Платформа 1С якраз передбачає хмарні технології, дозволяючи працювати в програмі і використовувати всі її продукти без прив'язки до конкретного місця. Схема структури роботи хмарного сервіса 1С зображена на рис. 1.5.

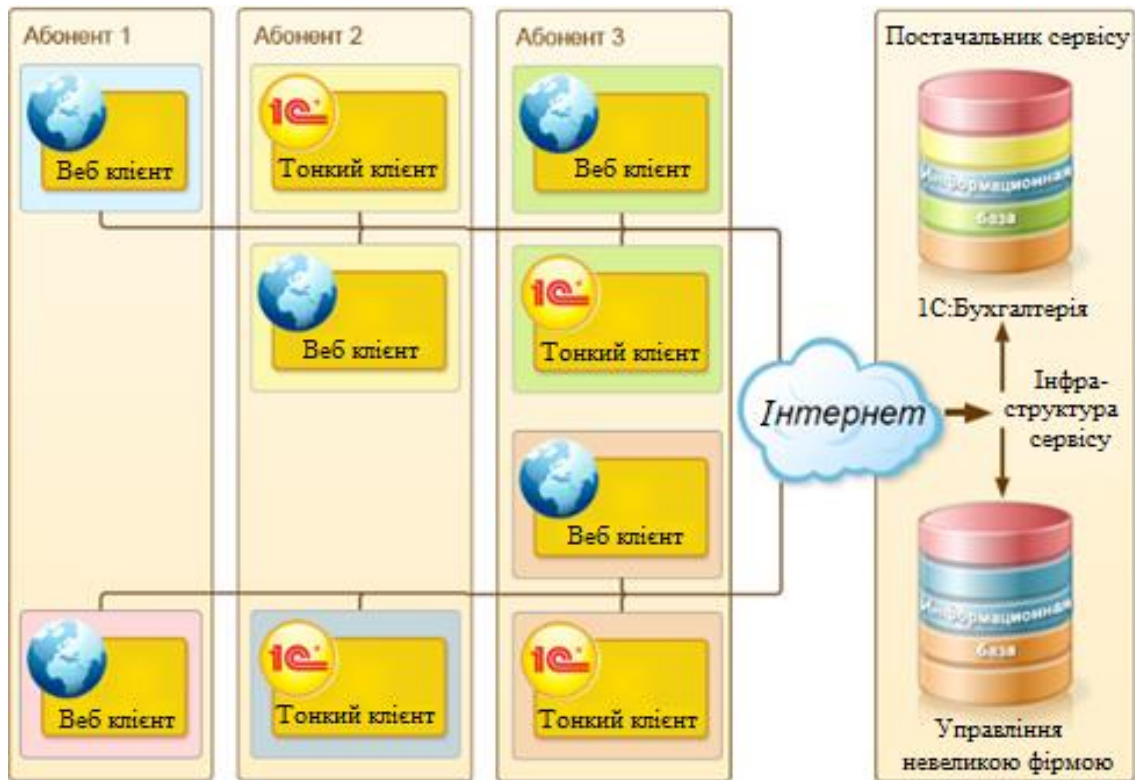


Рисунок 1.5 – Схематичний приклад поділу даних в хмарі 1С

Збереження даних забезпечується за допомогою шифрування каналу передачі інформації. Ніякі локальні дії, наприклад, поломка "залізного" сервера, проблеми з інтернетом або електрикою, не торкнуться інформації. Віддалений доступ до них є тільки у клієнта.

Хмарна онлайн бухгалтерія не вимагає від клієнта покупки віртуального сервера, адміністрування та підписки на оновлення. Для використання необхідно лише підключити інтернет і внести орендну плату.

Розгортка відбувається в хмарному сервері провайдера. Клієнт отримує пароль, за допомогою якого підключається до системи з будь-якого пристрою через інтернет без установки додаткового програмного забезпечення. Це оптимальне рішення для підприємств, коли потрібно забезпечити безвідмовну роботу програми і бази даних та заощадити на співробітниках, які будуть обслуговувати сервера.

## 1.5 Актуальність проблеми оптимізації

Протягом останніх років бухгалтерський облік стає складніше на всіх ділянках. У системах обліку і звітності введені нові об'єкти і поняття, змінилися методи і форми облікової роботи і, як результат, змінився підхід до вибору програм автоматизації бухгалтерського обліку. В даний час вже недостатньо, коли бухгалтерська програма забезпечує тільки ведення бухгалтерського обліку. Це повинні бути програми з великим функціональним наповненням, здатні вирішувати завдання аналізу господарської діяльності, податкового обліку і багато інших.

Крім цього програма повинна бути в першу чергу добре адаптована під кінцевого користувача, а саме бухгалтера, який завдяки простоті і функціональності зможе якісно виконувати свою роботу.

Що для цього потрібно? В першу чергу, програма повинна мати зрозумілий інтерфейс та набір інструментів для роботи в конкретній області, але бухгалтерський облік є комплексною системою, що має на увазі під собою складності в її уніфікації. Вона має безліч модулів для вирішення тих чи інших типових задач, формування звітів по податковим випискам, зарплатним нарахуванням та ін. Тому оптимізувати все одночасно складно і не потрібно, але окремо просто необхідно.

Одним з важливих пунктів є податковий облік без якого будь-яке підприємство не може існувати. Під цим маються на увазі податкові виплати до державного бюджету. Але всюди існує, так званий, "людський фактор", де в одній накладній або навіть в декількох зроблена помилка, яка займе у бухгалтера чимало часу, щоб з'ясувати, де і що саме не так.

Наприклад, менеджер зі збуту неправильно сформував податкові зобов'язання підприємства. Це веде до штрафних санкцій по відношенню до нього. Така помилка також призводить до того, що покупець не матиме можливість отримання податкового кредиту. У зв'язку з чим він може поскаржитися в державну фіскальну службу, що тягне до позапланової

перевірки підприємства. Тут вся відповідальність лягає на бухгалтера, який повинен проконтролювати правильність виписки податкових накладних.

Такі тонкощі роботи і призводять нас до того, що автоматизований бухгалтерський облік все одно вимагає оптимізації і є досить актуальною проблемою в наш час.

## **2 АНАЛІЗ І ВИБІР ТЕХНОЛОГІЙ ДЛЯ ОПТИМІЗАЦІЇ**

### **2.1 Шляхи оптимізації**

Дотримуючись того, що ведення бухгалтерського обліку є досить комплексною системою, яка враховує окремі профілі підприємств, які потребують різної звітності і мають свої тонкощі роботи, можна зробити висновок, що під конкретні запити користувачів потрібні внесення додаткових удосконалень.

Як одну з найбільш актуальних проблем в даній сфері було вирішено проблему контролю виписки податкових накладних, так як ПДВ та податкові накладні – це те без чого не може функціонувати будь-яке підприємство, яке займається торгівлею.

В Україні існує велика кількість установ, що займаються оптовою та роздрібною торгівлею. Отже, цю проблему можна виділити з переліку багатьох.

### **2.2 Технології та засоби використання**

В основі даного програмного забезпечення лежать засоби платформи 1С:Підприємство, які задовольняють в першу чергу користувачів. На сьогоднішній день 1С є лідером по використанню на підприємствах, практично не маючи конкурентів. Актуальна версія 1С на сьогодні це версія 8.3, але в більшості своїй на підприємствах все ще використовується версія 7.7 через свою стабільність і небажання керівництва переходити на більш свіжі версії.

Платформа 1С має наступну структуру(рис. 2.1):



Рисунок 2.1 – Структура платформи 1С

Розглянемо платформу 1С: Підприємство більш докладно.

Платформа 1С: Підприємство – це:

- робота в двох варіантах: файловий і серверний – у кожного свої переваги і недоліки;
- незалежна система обліку, яка легко встановлюється і налаштовується;
- потужна екосистема для програмістів з усіма необхідними об'єктами та інструментами;
- приємний ергономічний інтерфейс;
- порівняно недороге рішення для бізнесу;
- мережа франчайзі, початкове навчання;
- потужна база напрацювань і знань, накопичена роками;

Для організації обліку однієї платформи недостатньо. Платформа – це свого роду проміжна ланка між програмою і базою даних. Програми для цієї ланки називаються конфігураціями. У конфігурації описана структура бази даних у вигляді об'єктів, тексти процедур, що зберігаються в окремих модулях (модулів багато, вони є як у кожного об'єкта, так і у системи в

цілому, а також є загальні модулі). Платформа являє такий рівень абстракції, при якому прямого звернення до бази даних не потрібно, а також дозволяє абстрагуватися від операційної системи і типу бази даних.

Розглянемо платформу з точки зору розробника. По-перше, відсутня безпосередня робота з базами даних. Це просто не потрібно. Для нашої зручності всі дані представлені в вигляді взаємопов'язаних об'єктів (довідники, документи, реєстри відомостей, реєстри накопичення та ін.), а «витягування» цих даних з бази виконується платформою автоматично.

Наприклад, за умови, що в змінній «ЕлементНоменклатуры» міститься посилання на елемент довідника «Номенклатура», у довідника «Номенклатура» є реквізит «ОсновнойПоставщик», типу «Справочник.Контрагенты», а у довідника «Контрагенты» є реквізит «ПолноеНаименование», тоді код:

```
ЭлементНоменклатуры.ОсновнойПоставщик.ПолноеНаименование
```

приведе до того, що платформа 1С, використовуючи внутрішні механізми, знайде запис в таблиці, відповідній довіднику «Контрагенты» за ідентифікатором з поля «ОсновнойПоставщик» елемента номенклатури і поверне значення поля, відповідного полю «ПолноеНаименование» в об'єктній моделі довідника «Контрагенты».

### **2.2.1 Вбудована мова програмування 1С**

Мова 1С найбільш подібна за своїм синтаксисом мові Visual Basic. Однак відрізняється меншою строгістю і відсутністю деяких конструкцій:

- відсутня необхідність оголошення змінних – форматувати можна прямо в тексті модуля;

- немає жорсткої типізації змінних. В системі присутні типи значень, але відсутні строгі правила по типізації. Змінна, що зберігала посилання, може через пару рядків коду вже зберігати число або рядок;
- можна складати змінні зі значеннями різних типів, при цьому, тип результату буде таким, яким був тип у першій змінній. Наприклад:  
`k = "25" + 1; // k = "251", не 26`
- змінні зі значеннями різних типів можна порівняти на рівність чи нерівність (але не на  $>$   $<$ , це викличе виключення);
- присутні функції і процедури, допускається використання рекурсії (з обмеженням по глибині рекурсії);
- при вказівці функції або процедури без параметрів обов'язково вказувати порожні дужки в кінці;
- модульна система: у кожного довідника і документа декілька модулів. Крім того, модулі є у реєстрів, звітів, обробок і у кожної форми, а також у складі конфігурації є можливість створювати спільні модулі;
- немає класів, успадкування, інкапсуляції, поліморфізму;
- функцію можна передати як посилання, немає анонімних функцій;
- відсутні зворотні цикли (for i: = 5 downto 1 do).

### 2.2.2 Мова запитів 1С

Запити в 1С використовуються тільки для отримання вибірки даних із зручним відбором, сортуванням, угрупованнями. Здебільшого мова запитів 1С дуже нагадує запити SQL, однак, в мові запитів 1С є і функціонал, який відсутній в SQL, а саме:

- робота з об'єктами конфігурації 1С, замість таблиць SQL;
- робота з посиланнями, замість полів-ідентифікаторів SQL;
- підтримка розіменування, аналогічно тому, як це робиться в кодї;
- вибірка по входженню в групу (для довідників);
- ієрархічні підсумки;



- тимчасові таблиці і вкладені запити. Наприклад:  
(SELECT \* FROM (SELECT ...));
- конструктор запитів – зручний, об'єктний, автоматично оптимізує текст запиту для швидкого читання, який вміє розбирати текст запиту (парсити текст запиту і будувати з нього об'єктну модель);
- будівник звіту – надбудова над механізмом запитів, в якій доданий функціонал автоматичного оформлення результатів, а також гнучкого налаштування самого запиту в режимі «Підприємство». В результаті в стандартних платформах 1С з'явився універсальний звіт, в якому користувач може налаштувати які дані і в якому порядку він хоче бачити, і звіт сам згенерує вихідну таблицю;
- система компоновки даних(СКД) – наступна модель, що увібрала в себе функціонал універсального звіту і доповнила його деякими функціями обробки результатів.

### **2.3 Файлова і серверна бази даних**

Платформа складається з декількох частин і може працювати в файловому або серверному варіанті.

У першому варіанті вся база даних міститься в одному файлі (з розширенням «1cd») – платформа встановлюється на кожному комп'ютері і безпосередньо читає або пише в базу даних.

Плюси файлового варіанту: низька ціна і простота розгортання.

Мінуси: структура файлу «1cd» закрита, немає єдиної системи, яка взаємодіє з БД, що в результаті призводить до складнощів у використанні при великій кількості користувачів. Крім того, при великому обсязі бази даних виникають збої при проведенні динамічного оновлення.

Варіант роботи з 1С називається клієнт-серверним тоді, коли для роботи з базою використовують сторонні СУБД. 1С підтримує роботу з: IBM

DB2, MS SQL, Oracle DB, PostgreSQL. Сторонні системи управління базами даних є більш потужними у порівнянні з вбудованими в 1С і забезпечують кращу швидкодію, а також підвищують захищеність баз. SQL вважається найкращим для роботи з 1С. Postgree, наприклад, вимагає частого реіндексування баз. DB2 в порівнянні зі своїми конкурентами, має суттєві обмеження по максимальній довжині числа, а Oracle має відомі проблеми з упорядкуванням даних. Сама платформа використовує триланкову архітектуру (рис. 2.2):

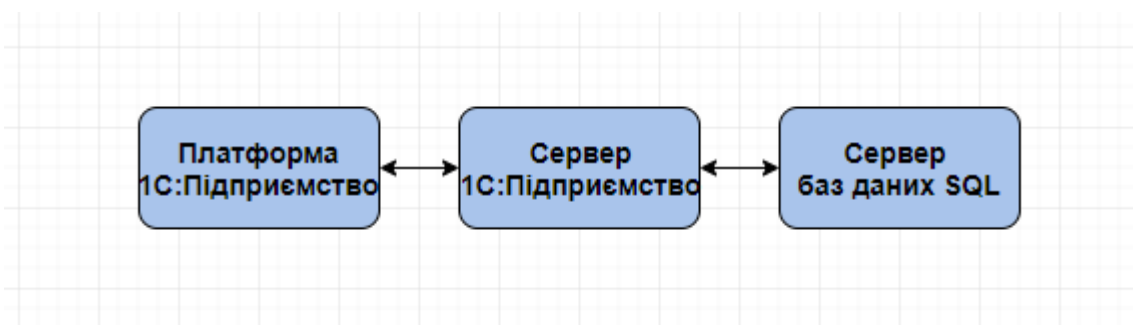


Рисунок 2.2 – Триланкова архітектура

Частина коду виконується на клієнті, частина на сервері. Крім стандартного режиму роботи, який обізвали режимом «товстий клієнт», є два режими роботи: «тонкий клієнт» і «web-клієнт». Також є «керовані форми», зовнішній вигляд яких будується в вигляді абстрактного дерева елементів. Програмувати під керовані форми і триланкову архітектуру складніше, але це дозволяє перенести майже все навантаження на сервер. Користувачі не будуть потребувати потужні комп'ютери, потрібен буде лише потужний сервер.

Плюси платформи: розподіл навантаження, відкритість бази даних, відсутність обмежень.

Мінуси: ціна, адже купувати доведеться окремо клієнтські ліцензії на клієнтські комп'ютери, окремо одну ліцензію на сервер 1С, а також окремо ліцензію на сервер SQL, якщо не використовувати безкоштовний. Також іноді виникають складності в роботі бази даних. Вони вирішуються набагато

швидше за рахунок доступності даних і наявності зовнішніх інструментів роботи з БД. Крім того, і в цьому випадку слід робити резервну копію бази даних.

Коли переносять базу з файлового варіанту в серверний:

- якщо з базою працюють до 5-7 чоловік і база має невеликий розмір буде досить файлового варіанту;

- якщо з базою працюють 8-12 осіб і вага бази наближається до 4 Гб, то має сенс задуматися про перехід на "клієнт-сервер." В даному випадку багато залежить від особистих відчуттів від роботи з базою при великій одночасній кількості користувачів. Справа в тому, що при роботі в базі кожен користувач, звертаючись до реєстрів, блокує доступ до них для інших користувачів і інші користувачі, при спробі звернення до заблокованого реєстру, встають в чергу. А в цей час 1С висить. Тому, якщо подвисання при великій кількості користувачів починають заважати роботі, пора вдосконалювати архітектуру;

- якщо з базою працюють більше 15 чоловік і розмір бази більш 4Гб, то клієнт-сервер необхідний.

Загальним плюсом є універсальність коду конфігурацій: за деякими винятками, конфігурація розроблена для файлової бази даних буде працювати на серверній, і навпаки. Приклад виключення – передача мутабельного значення на сервер або з сервера на клієнт.

Ще один плюс: платформа встановлюється дуже просто, не потрібно тонка настройка і установка будь-яких компонент.

Для серверного варіанту трохи більше налаштувань, але все робиться в діалозі установки і не вимагає прописувати що-небудь в окремих файлах. Деяку складність може представляти установка SQL-сервера, тому що в цьому випадку налаштувань набагато більше. Але це не відноситься до проблем платформи 1С. Більш того, якщо встановити, наприклад сервер MS SQL з усіма параметрами за замовчуванням, то 1С цілком зможе з ним працювати.

## 2.4 Конфігуратор 1С

Будь-яка програма складається з програмного коду, тобто власне з написаних на якій-небудь мові послідовності дій, які повинні бути виконані.

Однак ця сама програма повинна бути десь написана, тобто десь перебувати. У більшості випадків код програми пишеться в звичайних текстових файлах.

В 1С передбачається така річ Конфігурація 1С, яка включає в себе не тільки список налаштувань, шаблонів форм та іншого, а й програмний код 1С. Тому код 1С зберігається в конфігурації.

Конфігуратор є спеціальним режимом запуску 1С і призначений для розробки додатків в 1С:Підприємстві. В цьому режимі створюється або коригується структура бази даних, програмні модулі, виконуються адміністративні роботи. У режимі конфігуратора можна вводити інформацію в базу даних, за винятком завантаження бази даних з архівів.

Інтерфейс конфігуратора складається з панелі інструментів, дерева об'єктів конфігурації і області розробки. Дерево об'єктів – це перше, з чим розробник стикається при запуску конфігуратора(рис. 2.3).

Дерево об'єктів являє собою список зумовлених об'єктів метаданих, які не можна видалити. Дозволено тільки додавання нових видів об'єктів, наприклад, можна додати новий довідник «Адреса доставки», але не можна видалити весь об'єкт «Справочники».

За деревом об'єктів можна вести пошук, який знаходиться вгорі дерева об'єктів, також працює стандартне поєднання клавіш Ctrl + F. Також можна зробити глобальний пошук по всій базі для пошуку, наприклад, згадок об'єкта в модулях всіх об'єктів.

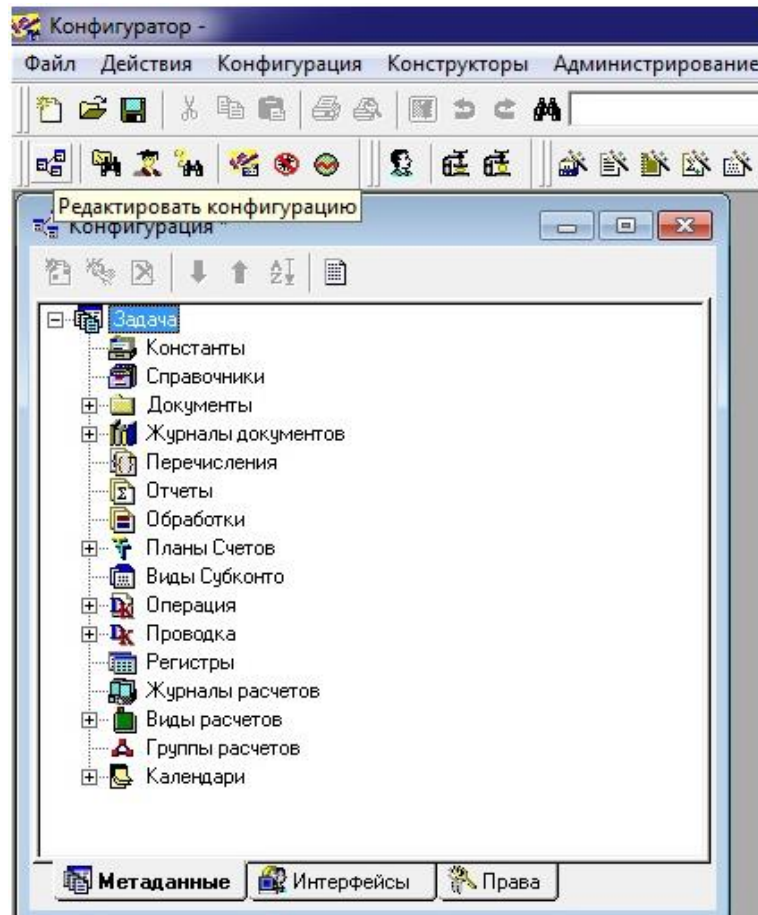


Рисунок 2.3 Дерево об'єктів

У базі знаходиться три конфігурації:

- типова від постачальника. За замовчуванням закрита від змін, якщо знаходиться на підтримці;
- основна, з якою працюють програмісти через конфігуратор. Після внесення в неї змін необхідне оновлення користувальницької бази;
- бази даних, з якою працюють безпосередньо користувачі.

При виникненні позаштатних ситуацій, наприклад, екстреного відключення електроенергії, база перестав відкриватися або працює з помилками. В цьому випадку можна провести тестування бази даних, виявити помилки і виправити їх в меню «Администрирование-Тестирование и исправление».

Цей інструмент допомагає перевірити і відновити логічну цілісність інформаційної бази, проводити реіндексацію і реструктуризацію таблиць, а також перераховувати підсумки.

## 2.5 Модульна система

Кожен об'єкт 1С, включаючи деякі вкладені, має свій модуль, який містить програмний код. Також є незалежні від об'єктів модулі, в яких може бути написаний програмний код, незалежний від конкретного об'єкта. Таким чином в 1С немає «єдиної» програми. Є набір модулів для написання програмного коду для кожного об'єкта конфігурації 1С.

Модуль складається з набору функцій і процедур, тобто обробників подій і методів.

На самому початку модуля можна розташувати змінні, які будуть використовувати обробники і події в цьому модулі.

У самому низу модуля можна розташувати програмний код без оформлення процедури або функції. Це можна назвати автостартом конкретного модуля. При спробі першого звернення до будь-якої процедури або функції цього модуля буде виконано автостарт. Зазвичай його використовують для початкового заповнення змінних розташованих на початку модуля.

Всю програму можна поділити на два види:

- метод об'єкта;
- реакція на події.

Методи. Як вже сказано раніше об'єкт 1С є цілісною структурою, яка включає в себе як дані, так і способи їх обробки. Ці способи – це набір дій (методів), які можна викликати для обробки даних. Приклад такої дії:

СправочникОбъект.Записать () – запис елемента довідника в базу даних.

Методи багатьох об'єктів 1С можуть бути стандартними (тобто запрограмованими в платформі 1С) і написаними програмістом на мові 1С. За допомогою другого можна розширювати функціонал об'єктів 1С за своїм бажанням.

Події. Події є в безлічі інших засобів розробки. Мета програми не тільки щось обчислити при запуску, але і підтримувати роботу користувача.

Подія користувача – натискання кнопки. У відповідь якась частина коду виконається, здійснивши реакцію на дії користувача.

Системні події – запис об'єкта 1С в базу даних(виникла системна подія «Запись обьекта»). Можливо налаштувати реакцію, яка виникне на події, викликані не користувачем, а самою системою. Яскравий приклад такої події – при запуску програми.

## **2.6 Структура інтерфейсу програмного забезпечення**

Оскільки для розробки програмного забезпечення була обрана платформа 1С: Підприємство програма матиме інтерфейс цього середовища, але при цьому всі необхідні вікна будуть налаштовані відповідно до їх очікуваного функціонала. Основою програми є аналіз рахунку 6415, на якому ґрунтується весь принцип роботи, який більш детально розписан в 3-й частині дипломної роботи.

Фактично, 1С володіє інструментами для створення всього необхідного для спрощення роботи бухгалтера за допомогою конфігуратора.

Даний звіт буде мати наступні елементи інтерфейсу:

- вікно для виставлення періоду і контрагента;
- вікно для вибору періоду;
- вікно для вибору контрагента;
- таблицю з аналізом рахунку 6415.

### 3 ПРИНЦИП РОБОТИ ТА ТЕСТУВАННЯ СТВОРЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Принцип роботи

Алгоритм роботи програмного забезпечення побудований на аналізі рахунку бухгалтерського обліку 6415. А саме його кредитового обороту, який виникає при нарахуванні ПДВ при проведенні операцій реалізації. Це нарахована до сплати в бюджет сума ПДВ на яку повинна бути виписана податкова накладна.

Отже, розпишемо принцип роботи поетапно:

– формуємо бухгалтерський запит за конкретний період по конкретному контрагенту(повний текст коду у додатку А, «процедура формування звіту» та додатку Б, «процедура відкриття») по рахунку 6415 з деталізацією до проводки:

Ит = СоздатьОбъект("БухгалтерскиеИтоги");

Ит.ИспользоватьСубконто("НалогиОтчисления");

Ит.ВключатьСубсчета(-1);

Ит.ВыполнитьЗапрос(ВыбНачПериода, ВыбКонПериода, "6415",,,  
2,"Проводка","Ит.ВыбратьПериоды());

– виконуємо вибірку з результатів запиту відбираючи лише результати з кредитовими оборотами;

– у відібраних проводках перевіряємо другу частину (дебетову). Нам необхідно отримати інформацію про замовлення та контрагентів. Якщо це можливо, то отримуємо її безпосередньо:

Ит.ВыполнитьЗапрос(ВыбНачПериода, ВыбКонПериода, "6415",,,  
2,"Проводка","Ит.ВыбратьПериоды());



```

Пока Ит.ПолучитьПериод() = 1 Цикл
  Опер = Ит.Операция;
  Если ит.ДО() <> 0 Тогда //сумма уменьшения НДС Записи книги покупок
    //пока на них не смотрим
    Продолжить;
  Иначе //это КО
    //нужная нам информация находится в дебете
    КорСчет = Опер.Дебет.Счет;
    ЗаказДа = 0; КонтрагентДа = 0;
    Для сч = 1 По КорСчет.КоличествоСубконто() Цикл
      Если КорСчет.ВидСубконто(сч) = ВидыСубконто.Заказы
        ЗаказДа = 1;
      КонецЕсли;
      Если КорСчет.ВидСубконто(сч) = ВидыСубконто.Контрагенты
        КонтрагентДа = 1;
      КонецЕсли;
    КонецЦикла;
    Если ЗаказДа + КонтрагентДа = 2 Тогда //есть аналитика
      Контрагент = Опер.Дебет.Субконто(1);
      Если (Контр.Выбран() = 1) И (Контрагент <> Контр) Тогда
        Продолжить;
      КонецЕсли;
      тз.НоваяСтрока();
      Заказ = Опер.Дебет.Субконто(2);
      ДокОсн = ит.Операция.Документ;
      тз.Контрагент = Контрагент;
      Индекс = "" + Контрагент + Заказ + ДокОсн;
      тз.Заказ = Заказ;
      тз.ДокОсн = ДокОсн;
      тз.ВидНДС = ит.Субконто(1);
      тз.СуммаНДСДт = ит.ДО();
      тз.СуммаНДСКт = ит.КО();
      тз.Индекс = Индекс;

```

– якщо отримати інформацію прямо неможливо (аналітика кореспондуючого рахунку не містить повної інформації), то намагаємося отримати дані непрямим шляхом. Дані, які отримати не вдалося заповнюємо «Порожніми значеннями». Таким чином, вони не будуть втрачені і дозволять бухгалтеру своєчасно побачити помилки:

Иначе // из проводки взять информацию невозможно, смотрим

ДокОсн = ит.Операция.Документ;

Попытка

Контрагент = ДокОсн.Контрагент;

Исключение

Контрагент = ПолучитьПустоеЗначение(Контр);

КонецПопытки;

Если (Контр.Выбран() = 1) И (Контрагент <> Контр) Тогда

Продолжить;

КонецЕсли;

тз.НоваяСтрока();

тз.Контрагент = Контрагент;

Попытка

Заказ = ДокОсн.Заказ;

тз.Заказ = Заказ;

Исключение КонецПопытки;

тз.ДокОсн = ДокОсн;

тз.ВидНДС = Опер.Кредит.Субконто(1);

тз.СуммаНДСДт = ит.ДО();

тз.СуммаНДСКт = ит.КО();

тз.Индекс = "" + Контрагент + Заказ + ДокОсн ;

КонецЕсли;

КонецЕсли;

КонецЦикла;

– всі дані, які пройшли відбір заносимо в попередньо створену тимчасову «таблицю значень», яка має структуру:

тз.НоваяКолонка("Контрагент","Справочник.Контрагенты");  
 тз.НоваяКолонка("Заказ","Документ");  
 тз.НоваяКолонка("ДокОсн","Документ");  
 тз.НоваяКолонка("ВидНДС","Справочник.НалогиОтчисления");  
 тз.НоваяКолонка("СуммаНДСДт","Число",12,2);  
 тз.НоваяКолонка("СуммаНДСКт","Число",12,2);  
 тз.НоваяКолонка("НН","Документ");  
 тз.НоваяКолонка("ВидНДСНН","Справочник.НалогиОтчисления");  
 тз.НоваяКолонка("СуммаНДСНН","Число",12,2);  
 тз.НоваяКолонка("СуммаБазыНН","Число",12,2);  
 тз.НоваяКолонка("Индекс","Строка",200);

Маємо дві частини: перша заповнюється даними з запиту по рахунку 6415 (поля «Контрагент, Замовлення, ДокОсн, ВідНДС, СуммаНДСДт, СуммаНДСКт, Індекс»). Решта поля заповнюються при обробці результатів наступного запиту. Поле «Індекс» є службовим і призначене для пошуку записів в ТЗ в процесі обробки результатів другого запиту.

– далі виконується другий запит за той же період, що і перший, але вже за документами «Податкова накладна» (повний текст коду у додатку В, «процедура заповнення податкових накладних»):

```

"/>{{ЗАПРОС(Сформировать)
|Период с ВыбНачПериода по ВыбКонПериода;
|Обрабатывать НеПомеченныеНаУдаление;
|Без итогов;
|ДокНН = Документ.НалоговаяНакладная.ТекущийДокумент;
|Контрагент = Документ.НалоговаяНакладная.Контрагент;
|Заказ = Документ.НалоговаяНакладная.Заказ;
|ВидНДС = Документ.НалоговаяНакладная.ВидНДС;
|РодДок = Документ.НалоговаяНакладная.РодительскийДокумент;
|БазаНДС = Документ.НалоговаяНакладная.БазаНДС;
|СуммаБезНДС = Документ.НалоговаяНакладная.СуммаБезНДС;

```

```

|НДС = Документ.НалоговаяНакладная.НДС;
|Функция НДССумма = Сумма(НДС);
|Функция СуммаБезНДССумма = Сумма(СуммаБезНДС);
|Группировка Контрагент без групп;
|Группировка ДокНН;
|Группировка Заказ;
|"/}}ЗАПРОС

```

– потім виконуємо дозаповнення ТЗ даними з цього запиту. Для пошуку рядка ТЗ використовуємо поле «Індекс» ТЗ, створений раніше. У разі, якщо запис в ТЗ з обчисленим за результатами другого запиту індексом не знайдено, то в ТЗ створюємо новий запис для того щоб інформація не була втрачена. Вона буде як і в разі першого запиту сигналізувати бухгалтеру про ймовірну помилку.

Пока Запрос.Группировка() = 1 Цикл

Стр = 0;

Контрагент = Запрос.Контрагент;

Заказ = Запрос.Заказ;

ДокОсн = Запрос.РодДок;

Индекс = Формат(""+Контрагент + Заказ + ДокОсн,"С200");//строка

Если тз.НайтиЗначение(Индекс,Стр,"Индекс") = 1 тогда //Нашли

//проверяем заполнены ли поля налоговой накладной в таблице

Если ПустоеЗначение(тз.ПолучитьЗначение(Стр,"НН")) = 1

тз.УстановитьЗначение(стр,"НН",Запрос.ДокНН);

тз.УстановитьЗначение(стр,"ВидНДСНН",Запрос.ВидНДС);

тз.УстановитьЗначение(стр,"СуммаНДСНН",Запрос.НДССумма);

тз.УстановитьЗначение(стр,"СуммаБазыНН",Запрос.СуммаБезНДССумма);

Иначе //налоговая накладная уже есть, дописываем найденную

ДобавитьСтрТЗ(Запрос);

КонецЕсли;

Иначе

```

ДобавитьСтрТЗ(Запрос);
КонецЕсли;
КонецЦикла;

```

– після заповнення «таблиці значень» даними з двох запитів звертаємо ТЗ і сортуємо записи в ній для коректного виведення в таблицю для відображення результату користувачеві.

```

ЗаполнитьНН();
тз.Свернуть("Контрагент,Заказ,ДокОсн,НН","СуммаНДСДт,СуммаНДСКт");
тз.Сортировать("Контрагент,Заказ,ДокОсн");

```

– якщо потрібно виводимо на друк(повний текст коду знаходиться у додатку Г, процедура друку):

```

Процедура Печать()
Перем Путь,ИмяФ;
РасположениеФайла(Путь,ИмяФ);
Расшифровка = СоздатьОбъект("СписокЗначений");

Расшифровка.Установить("Отчет", Путь+ИмяФ);
Расшифровка.Установить("ВыбНачПериода", ВыбНачПериода);
Расшифровка.Установить("ВыбКонПериода", ВыбКонПериода);
Расшифровка.Установить("Контр", Контр);
Таб.ВывестиСекцию("Шапка");
тз.ВыбратьСтроки();
стКонтр = "";
стЗаказ = "";
стДокОсн = "";

```

### 3.2 Тестування додатку

Для демонстрації роботи даного програмного забезпечення виконаємо перевірку правильності виписаних податкових накладних.

Спочатку для обраної бази потрібно обрати контрагента та період(рис. 3.1 та 3.2):

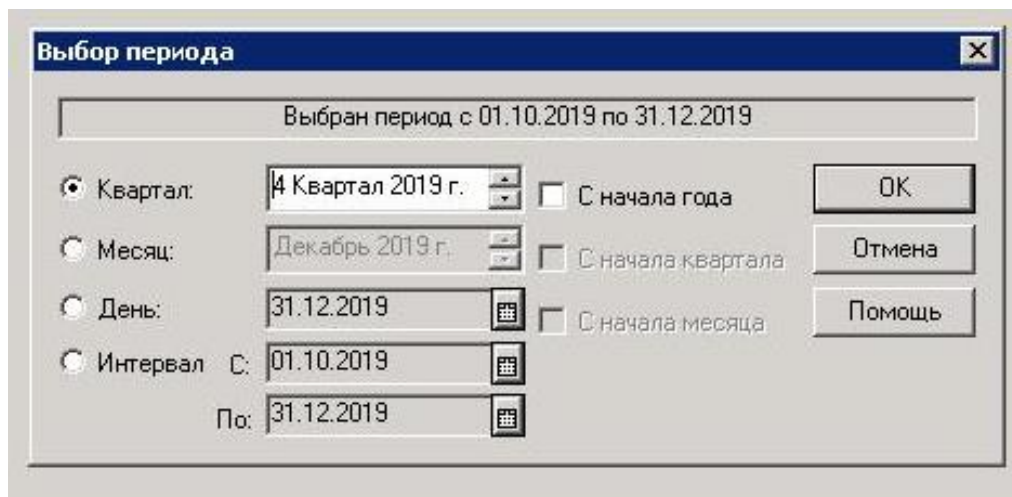


Рисунок 3.1 – Виставлення періода

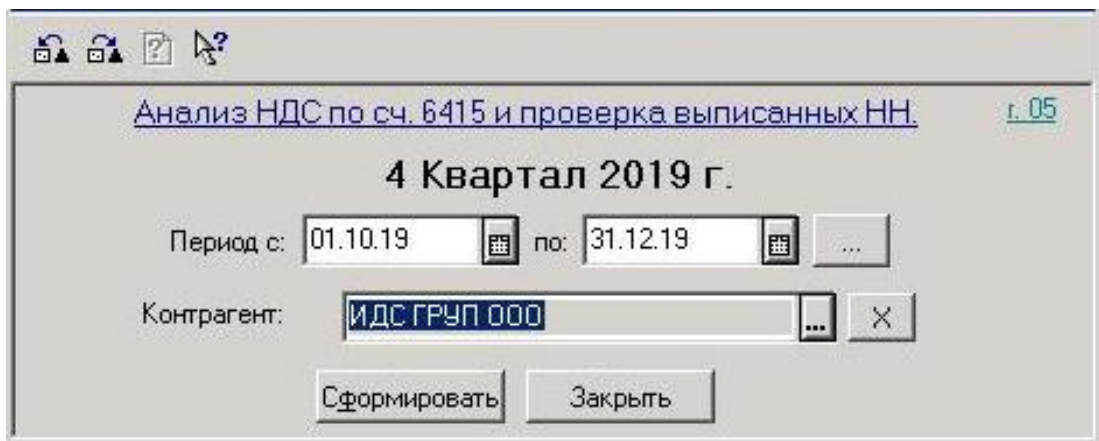


Рисунок 3.2 – Вибір контрагента

В результаті роботи програми отримуємо проаналізовані дані у вигляді таблиці, де є інформація по імені контрагента, його замовленням, (рис. 3.3) виписаним на підставі замовлень видатковим накладним, а також податковим накладним і сумам по ним.

Анализ кредита счета 6415 за период с 01.12.19 по 31.12.19						
Контрагент	Заказ	Основание	Сумма НДС	НН	Сумма НДС	Сумма базы
Комбинат Звезда	Счет-фактура В-00011284 (22.11.19)	Расх. наклад. В-00018212 (03.12.19)	387.96			
<b>ИТОГО:</b>			<b>387.96</b>		<b>0.00</b>	<b>0.00</b>

Рисунок 3.3 – Сформована таблиця аналізу

Червоний колір сигналізує бухгалтеру про потребу втручання. При перегляді інформації через рахунок-фактуру(рис. 3.4) по даній накладній бачимо, що податкова накладна була виписана, проте товар не був відвантажений і оплачений.

Документ	Номер	Дата	Время	Информация	Сумма	Автор	ИДОбе
<input checked="" type="checkbox"/> Налог, наклад.	Дк-0000054	03.12.19	13:04:47	ИДС ГРУП ООО	1,017.048		1426528

Рисунок 3.4 – Рахунок-фактура

Розглянемо також інший можливий варіант. Також обираємо контрагента, виставляємо період та перевіряємо виписку накладних на відповідності(рис. 3.5).

Анализ кредита счета 6415 за период с 01.10.19 по 31.12.19						
Контрагент	Заказ	Основание	Сумма НДС	НН	Сумма НДС	Сумма базы
Комбинат Звезда	Счет-фактура В-00010869 (07.11.19)	Расх. наклад. В-00017896 (18.11.19)	1.20	Налог наклад. Но-0000380 (18.11.19)	1281.85	6409.25
		Расх. наклад. В-00017896 (18.11.19)	1280.65			
	Счет-фактура В-00010872 (07.11.19)	Расх. наклад. В-00017895 (18.11.19)	14.60	Налог наклад. Но-0000379 (18.11.19)	339.18	1695.90
		Расх. наклад. В-00017895 (18.11.19)	324.58			
	Счет-фактура В-00011284 (22.11.19)	Расх. наклад. В-00018212 (03.12.19)	387.96			
<b>ИТОГО:</b>			<b>2008.99</b>		<b>1621.03</b>	<b>8105.15</b>

Рисунок 3.5 – Сформована таблиця аналізу

Знову бачимо помилку, допущену при виписці даної податкової накладної. Знову дивимося рахунок-фактуру (рис. 3.6), щоб зрозуміти, що не так.

Документ	Номер	Дата	Время	Информация	Сумма	Автор	ИД Олс	ИД Документ
Расх наклад	В-00018212	03.12.19	08:49:34	Комбинат Звезда	2,327.748		1426101	

Рисунок 3.6 – Рахунок-фактура

Цього разу маємо видаткову накладну, але відсутня податкова. Отже, відвантаження товару було здійснено, але він не був оплачений.



### 3.3 Оцінка користі на підприємстві

Відгук про використання програмного забезпечення головним бухгалтером ТОВ "Велес опт":

Дане програмне забезпечення в умовах напруженої роботи бухгалтерії, з постійними внесеннями різних змін до податкового кодексу України дуже актуальне для роботи з великою кількістю співробітників, як для офісної, так і для віддаленої роботи.

Грунтуючись на тому, що основним фундаментом поповнення бюджету країни є податки на додану вартість (ПДВ), а при неправильному формуванні податкових зобов'язань підприємства з'являється загроза як штрафних санкцій, так і податкових перевірок, а також блокування податкових накладних в системі електронного адміністрування податку на додану вартість.

Додаток дозволяє контролювати виписку податкових накладних (які в день можуть досягати 500 штук), що економить гроші підприємству та час бухгалтеру.

Інтуїтивно зрозуміло, коли некоректно виписана накладна. Рядок просто підсвічується червоним кольором. Відразу можна створити звіт по конкретному контрагенту і знайти з якої причини помилився відповідальний за виписку податкової накладної. Причин може бути безліч. Наприклад, недотримання першої події, де клієнт оплатив товар і необхідно виписати податкову накладну по банківській виписці або відвантаження товару буде протягом декількох днів. Також якщо виписати накладну неправильно, то можна завищити свої зобов'язання через виписку податкової накладної і на оплату товару і на відвантаження тільки в іншій даті. Це призведе до сплати до бюджету і наступні незручності як для свого покупця так і для себе. Тому треба враховувати всі нюанси податкового законодавства.

Для коректної оцінки програмного забезпечення було обрано такі показники якості продукту з оцінкою від 1 до 5 (табл. 1):

Таблиця 3.1 – Оцінка якості

№	Показники якості	Оцінка
1	Користувальницький інтерфейс	4
2	Відповідність профілю діяльності замовника	5
3	Ресурсна ефективність	5
4	Гнучкість налаштування	4
5	Навчання персонала	5
6	Співвідношення ціна можливості	5

Узагальнюючи, можна сказати, що в першу чергу це дуже корисний інструмент для новачків в професії. Проте необхідний і для досвідчених співробітників, так як значно оптимізує пошук і усунення помилок, а також зменшує ризики для підприємства.

Даний програмний продукт добре зарекомендував себе на нашому підприємстві і мені, як бухгалтеру, він дуже спрощує пошук і усунення помилок за виписаними накладними. Можу сміливо рекомендувати його до установки на інших підприємствах.

## ВИСНОВКИ

У висновку варто сказати, що тема правильної організації облікової політики на підприємстві в наш час є досить актуальною, так як від правильного вибору технології для роботи залежить успіх будь-якої організації. У сучасних умовах для підвищення ефективності управління потрібне вдосконалення облікової політики і, в першу чергу, роботи з документами, оскільки кожне управлінське рішення завжди базується на інформації.

В даній дипломній роботі було розглянуто бухгалтерській облік в цілому, його тенденції ведення, сучасні існуючі інструменти та актуальність проблем, які існують на цей час. Після чого було обрано напрямок оптимізації та програмне забезпечення, інструментами якого керувався автор. Як результат роботи розроблено програмне забезпечення, що відповідає сучасним стандартам, використовується на підприємстві та вирішує поставлені задачі.

Плюси від впровадження даної технології зрозумілі всім. Адже це економія часу для бухгалтера, мінімізація витрат для підприємства на можливі штрафні санкції. Даний додаток також володіє простотою в управлінні, зрозумілим і звичним оку інтерфейсом.

У майбутньому планується розвиток додатку та розширення функціоналу.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Корольов О. Л., Похилько Е. Н. Інформаційні системи підприємств. Сімферополь : ТНУ, 2011. 194 с.
2. Остап'юк М., Лучко М., Даньків Й. Історія бухгалтерського обліку. Київ : Знання, 2009. 278 с.
3. Лень В., Гливенко В. Бухгалтерський облік в Україні. Основи та практика. Навчальний посібник. Київ : Центр навчальної літератури, 2018. 608 с.
4. Ясенев В. Н. Информационные системы и технологии в экономике. Москва : ЮНИТИ, 2008. 560 с.
5. Рязанцева Н., Рязанцев Д. 1С: Предприятие. Секреты программирования. Санкт-Петербург : БХВ-Петербург, 2012, 368 с.
6. Аналоги 1С:Підприємство: безкоштовні і "умовно безкоштовні". TQM systems. URL: <http://tqm.com.ua/ua/likbez/ua-articles/bezkoshtovni-analohy-1s> (дата звернення 12.08.19)
7. 1С:Предприятие. Википедия. URL: <https://ru.wikipedia.org/wiki/1С:Предприятие> (дата звернення 15.07.19)

## ДОДАТОК А

### Код процедури формування звіту

Процедура Сформировать()

тз.Очистить();

тз.НоваяКолонка("Контрагент", "Справочник.Контрагенты");

тз.НоваяКолонка("Заказ", "Документ");

тз.НоваяКолонка("ДокОсн", "Документ");

тз.НоваяКолонка("ВидНДС", "Справочник.НалогиОтчисления");

тз.НоваяКолонка("СуммаНДСДт", "Число", 12,2);

тз.НоваяКолонка("СуммаНДСКт", "Число", 12,2);

тз.НоваяКолонка("НН", "Документ");

тз.НоваяКолонка("ВидНДСНН", "Справочник.НалогиОтчисления");

тз.НоваяКолонка("СуммаНДСНН", "Число", 12,2);

тз.НоваяКолонка("СуммаБазыНН", "Число", 12,2);

тз.НоваяКолонка("Индекс", "Строка", 200); //индекс по контрагенту

Ит = СоздатьОбъект("БухгалтерскиеИтоги");

Ит.ИспользоватьСубконто("НалогиОтчисления");

Ит.ВключатьСубсчета(-1);

Ит.ВыполнитьЗапрос(ВыбНачПериода, ВыбКонПериода, "6415",,,

2,"Проводка", "Ит.ВыбратьПериоды());

Пока Ит.ПолучитьПериод() = 1 Цикл

Опер = Ит.Операция;

Если ит.ДО() <> 0 Тогда //сумма уменьшения НДС Записи книги покупок

//пока на них не смотрим

Продолжить;

Иначе //это КО

//нужная нам информация находится в дебете

КорСчет = Опер.Дебет.Счет;

ЗаказДа = 0; КонтрагентДа = 0;

Для сч = 1 По КорСчет.КоличествоСубконто() Цикл

Если КорСчет.ВидСубконто(сч) = ВидыСубконто.Заказы

```

        ЗаказДа= 1;
    КонечЕсли;
    Если КорСчет.ВидСубконто(сч) = ВидыСубконто.Контрагенты
        КонтрагентДа= 1;
    КонечЕсли;
КонецЦикла;
        Если ЗаказДа + КонтрагентДа = 2 Тогда //есть аналитика
            Контрагент = Опер.Дебет.Субконто(1);
            Если (Контр.Выбран() = 1) И (Контрагент <> Контр) Тогда
                Продолжить;
            КонечЕсли;
            тз.НоваяСтрока();
            Заказ = Опер.Дебет.Субконто(2);
            ДокОсн = ит.Операция.Документ;
            тз.Контрагент = Контрагент;
            Индекс = "" + Контрагент + Заказ + ДокОсн;
            тз.Заказ = Заказ;
            тз.ДокОсн = ДокОсн;
            тз.ВидНДС = ит.Субконто(1);
            тз.СуммаНДСДт = ит.ДО();
            тз.СуммаНДСКт = ит.КО();
            тз.Индекс = Индекс;
                Иначе // из проводки взять информацию невозможно, смотрим
            ДокОсн = ит.Операция.Документ;
            Попытка
                Контрагент = ДокОсн.Контрагент;
            Исключение
                Контрагент = ПолучитьПустоеЗначение(Контр);
            КонечПопытки;
            Если (Контр.Выбран() = 1) И (Контрагент <> Контр) Тогда
                Продолжить;
            КонечЕсли;
            тз.НоваяСтрока();
            тз.Контрагент = Контрагент;
            Попытка

```

Заказ = ДокОсн.Заказ;  
тз.Заказ = Заказ;  
Исключение КонецПопытки;  
тз.ДокОсн = ДокОсн;  
тз.ВидНДС = Опер.Кредит.Субконто(1);  
тз.СуммаНДСДт = ит.ДО();  
тз.СуммаНДСКт = ит.КО();  
тз.Индекс = "" + Контрагент + Заказ + ДокОсн ;  
КонецЕсли;  
КонецЕсли;  
КонецЦикла;  
ЗаполнитьНН();  
тз.Свернуть("Контрагент,Заказ,ДокОсн,НН", "СуммаНДСДт,СуммаНДСКт");  
тз.Сортировать("Контрагент,Заказ,ДокОсн");  
Печать();  
КонецПроцедуры

## ДОДАТОК Б

### Код процедури відкриття

Процедура ПриОткрытии()

Контр.ВыборГруппы(0);

ВыбНачПериода = НачалоПериодаБИ();

ВыбКонПериода = КонецПериодаБИ();

Если глФлагРасшифровки = 1 Тогда

    Закрыть = 1;

    Обновить = глОбновить;

    Если Обновить <> 0 Тогда

        ВыбНачПериода = глРасшифровка.Получить("ВыбНачПериода");

        ВыбКонПериода = глРасшифровка.Получить("ВыбКонПериода");

        Контр = глРасшифровка.Получить("Контр");

        Таб = глТаблица;

        Таб.Очистить();

    КонецЕсли;

Если Обновить <> 2 Тогда

    Сформировать();

    глФлагРасшифровки = 0;

    глОбновить = 0;

    глРасшифровка = 0;

    СтатусВозрата(0);

    Возрат;

КонецЕсли;

Иначе

    Закрыть = 0;

    Обновить = 0;

    ВыбНачПериода = НачалоПериодаБИ();

    ВыбКонПериода = КонецПериодаБИ();

    Таб = СоздатьОбъект("Таблица");

    Таб.ИсходнаяТаблица("Таблица");

КонецЕсли;



Форма.КнопкаПоУмолчанию("Сформировать");  
КонецПроцедуры

## ДОДАТОК В

### Код процедуры заполнения податковых накладных

Процедура ЗаполнитьНН()

Запрос = СоздатьОбъект("Запрос");

ТекстЗапроса =

"//{{ЗАПРОС(Сформировать)

|Период с ВыбНачПериода по ВыбКонПериода;

|Обрабатывать НеПомеченныеНаУдаление;

|Без итогов;

|ДокНН = Документ.НалоговаяНакладная.ТекущийДокумент;

|Контрагент = Документ.НалоговаяНакладная.Контрагент;

|Заказ = Документ.НалоговаяНакладная.Заказ;

|ВидНДС = Документ.НалоговаяНакладная.ВидНДС;

|РодДок = Документ.НалоговаяНакладная.РодительскийДокумент;

|БазаНДС = Документ.НалоговаяНакладная.БазаНДС;

|СуммаБезНДС = Документ.НалоговаяНакладная.СуммаБезНДС;

|НДС = Документ.НалоговаяНакладная.НДС;

|Функция НДССумма = Сумма(НДС);

|Функция СуммаБезНДССумма = Сумма(СуммаБезНДС);

|Группировка Контрагент без групп;

|Группировка ДокНН;

|Группировка Заказ;

|"//}}ЗАПРОС

;

Если (Контр.Выбран() = 1) Тогда

ТекстЗапроса = ТекстЗапроса + "Условие(Контрагент = Контр);";

КонецЕсли;

// Если ошибка в запросе, то выход из процедуры

Если Запрос.Выполнить(ТекстЗапроса) = 0 Тогда

Возврат;

КонецЕсли;

Пока Запрос.Группировка() = 1 Цикл

Стр = 0;

Контрагент = Запрос.Контрагент;

Заказ = Запрос.Заказ;

ДокОсн = Запрос.РодДок;

Индекс = Формат(""+Контрагент + Заказ + ДокОсн,"С200");//строка

Если тз.НайтиЗначение(Индекс,Стр,"Индекс") = 1 тогда //Нашли

//проверяем заполнены ли поля налоговой накладной в таблице

Если ПустоеЗначение(тз.ПолучитьЗначение(Стр,"НН")) = 1

тз.УстановитьЗначение(стр,"НН",Запрос.ДокНН);

тз.УстановитьЗначение(стр,"ВидНДСНН",Запрос.ВидНДС);

тз.УстановитьЗначение(стр,"СуммаНДСНН",Запрос.НДССумма);

тз.УстановитьЗначение(стр,"СуммаБазыНН",Запрос.СуммаБезНДССумма);

Иначе //налоговая накладная уже есть, дописываем найденную

ДобавитьСтрТЗ(Запрос);

КонецЕсли;

Иначе

ДобавитьСтрТЗ(Запрос);

КонецЕсли;

КонецЦикла;

КонецПроцедуры

## ДОДАТОК Г

### Код процедуры друку

Процедура Печать()

Перем Путь,ИмяФ;

РасположениеФайла(Путь,ИмяФ);

Расшифровка = СоздатьОбъект("СписокЗначений");

Расшифровка.Установить("Отчет", Путь+ИмяФ);

Расшифровка.Установить("ВыбНачПериода", ВыбНачПериода);

Расшифровка.Установить("ВыбКонПериода", ВыбКонПериода);

Расшифровка.Установить("Контр", Контр);

Таб.ВывестиСекцию("Шапка");

тз.ВыбратьСтроки();

стКонтр = "";

стЗаказ = "";

стДокОсн = "";

Расшифровка.УдалитьВсе();

ФиксСтрок = Таб.ВысотаТаблицы();

СуммаДок = 0; СуммаНН = 0; СуммаБазы = 0;

СуммаДокВсего = 0; СуммаННВсего = 0; СуммаБазыВсего = 0;

Пока тз.ПолучитьСтроку() = 1 Цикл

Если тз.Контрагент <> стКонтр Тогда

    Если стКонтр <> "" Тогда

        Таб.ВывестиСекцию("ИтогоКонтр");

    КонецЕсли;

    стКонтр = тз.Контрагент;

Если СуммаДок <> СуммаНН Тогда

    Область = Таб.Область(Таб.ВысотаТаблицы(),Таб.ВысотаТаблицы(),);

    Область.ЦветТекста(255,255,255);

    Область.ЦветФона(255,0,0);

КонецЕсли;

```

    Таб.ВывестиСекцию("Стр|Контр");
    СуммаДок = 0; СуммаНН = 0; СуммаБазы = 0;
Иначе
    Таб.ВывестиСекцию("Пусто|Контр");
КонецЕсли;

Если тз.Заказ <> стЗаказ Тогда
    стЗаказ = тз.Заказ;
    Таб.ПрисоединитьСекцию("Стр|Заказ");
Иначе
    Таб.ПрисоединитьСекцию("Пусто|Заказ");
КонецЕсли;

стДокОсн = тз.ДокОсн;
Таб.ПрисоединитьСекцию("Стр|ДокОсн");
СуммаДок = СуммаДок + тз.СуммаНДСКт;
СуммаДокВсего = СуммаДокВсего + тз.СуммаНДСКт;

Если ПустоеЗначение(тз.НН) = 0 Тогда
    Таб.ПрисоединитьСекцию("Стр|НН");
    СуммаНН = СуммаНН + тз.СуммаНДСНН;
    СуммаБазы = СуммаБазы + тз.СуммаБазыНН;
    СуммаННВсего = СуммаННВсего + тз.СуммаНДСНН;
    СуммаБазыВсего = СуммаБазыВсего + тз.СуммаБазыНН;
Иначе
    Таб.ПрисоединитьСекцию("Пусто|НН");
КонецЕсли;
КонецЦикла;
Таб.ВывестиСекцию("Итог");
Если СуммаДокВсего <> СуммаННВсего Тогда
    Область = Таб.Область(Таб.ВысотаТаблицы(),,Таб.ВысотаТаблицы(),);
    Область.ЦветТекста(255,255,255);
    Область.ЦветФона(255,0,0);
КонецЕсли;

```

Таб.ТолькоПросмотр(1);

Таб.Опции(0,0,ФиксСтрок,1,"Анализ6415","Окно6415");

Таб.Показать("Анализ счета 6415","");

КонецПроцедуры