

Київський національний торговельно-економічний університет
Кафедра програмної інженерії та інформаційних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка системи обробки інформації в поліграфічному підприємстві»

Студента 4 курсу, 7 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

Коблая Андрія
Сергійовича

підпис студента

Науковий керівник
кандидат педагогічних наук,
старший викладач

Жирова Тетяна
Олександрівна

підпис керівника

Гарант освітньої програми,
кандидат технічних наук,
доцент

Цензура Микола
Олександрович

підпис гаранта

КИЇВ – 2020

ЗМІСТ

ВСТУП.....	4
РОЗДІЛ 1. ОСНОВНІ ТЕХНІЧНІ ВІДОМОСТІ ТА ТЕХНІЧНЕ ЗАВДАННЯ	6
1.1 Загальні відомості.....	6
1.2 Найменування системи	6
1.3 Мета та призначення створення системи.....	6
1.3.1. Призначення системи.....	6
1.3.2. Мета системи	6
1.4 Теоретичні відомості.....	8
1.4.1. Windows Forms	8
1.4.2. База даних	9
1.4.3. Система керування базами даних.....	9
1.4.4. Обґрунтування вибору мови програмування	16
1.5. Вимоги до системи	20
1.6. Висновки до розділу 1.....	21
РОЗДІЛ 2. АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ ТА ХАРАКТЕРИСТИКА ЙОГО РОБОТИ.....	22
2.1. Опис системи «C++ Builder»	22
2.2. Конкурентність програми.....	25
2.3. Висновки до розділу 2.....	29
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ ОБРОБКИ ІНФОРМАЦІЇ НА ПОЛІГРАФІЧНОМУ ПІДПРИЄМСТВІ	30
3.1. Програмне забезпечення «PolyCall».....	30
3.2. Діаграми процесів програми	30
3.3. Інтерфейс програми.....	34
3.4. Висновки до розділу 3	39

					КНТЕУ 121-07-08.БР			
					Розробка системи обробки інформації в поліграфічному підприємстві	Стадія	Аркуш	Арку-
Зм.	Аркуш	№ докум.	Підпис	Дата		Зміст	2	41
Зав. каф.	Криворучко О.В.					Факультет інформацій- них технологій 4 курс, 7 група		
Керівник	Жирова Т. О							
Гарант	Цензура М.О.							
Розроб	Коблай А.С.				Зміст			

ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
ДОДАТКИ.....	42

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		3

ВСТУП

Тема роботи – «Розробка системи обробки інформації на поліграфічному підприємстві».

Об'єкт дослідження – інформаційна система обробки інформації.

Предмет дослідження – програмне забезпечення, що створено задля того, щоб підприємство мало змогу швидко розрахувати необхідну їм кількість матеріалів, що буде використана під час створення поліграфічної продукції (газет та книжок, зокрема).

Мета проекту – створити програмне забезпечення, яке буде обробляти інформацію щодо типу продукції, та кількості цієї продукції, та показувати необхідну кількість матеріалів, що потрібно закупили, показуватиме собівартість одного екземпляру, та всієї продукції в цілому.

Завдання проекту – створення програмного забезпечення, яке буде враховувати кількість необхідних матеріалів для створення поліграфічної продукції, мати можливість зберігати шаблон продукції, буде рахувати ціну за одну одиницю продукції, та ціну всієї партії.

Актуальність – автор вважає, що, хоч ринок поліграфії зараз не знаходиться на підйомі, проте він все ще залишається актуальним, тому що ще багато компаній випускають, і можуть випускати різну продукцію, і. зокрема, хоч на газети зараз зовсім майже немає попиту, зокрема тільки літні люди їх купляють, і то, далеко не всі, є попит на книги, навіть, ті ж самі підручники для школярів, або, для студентів, і завжди, що купляють частіше – це художня література, що дозволяє розслабитись довгими зимовими ночами.

					КНТЕУ 121-07-08.БР			
					Розробка системи обробки інформації в поліграфічному підприємстві	Стадія	Аркуш	Арку-
Зм.	Аркуш	№ докум.	Підпис	Дата		Вступ	4	41
Зав. каф.	Криворучко О.В.							
Керівник	Жирова Т. О							
Гарант	Цензура М.О.							
Розроб	Коблай А.С.				Вступ	Факультет інформаційних технологій 4 курс, 7 група		

І, через те, що програм подібного типу майже немає, а багато підприємств все ще працюють в Excel і, додаючи данні до неймовірно великих таблиць, і, доволі часто можуть помилятись, переплутавши в розпалі робочого процесу строку з даними в Excel, автор вважає – що це доволі актуальний час задля того, щоб вивести дану програму на ринок.

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		5

РОЗДІЛ 1. ОСНОВНІ ТЕХНІЧНІ ВІДОМОСТІ ТА ТЕХНІЧНЕ ЗАВДАННЯ

1.1. Загальні відомості

Найменування системи

Програмне забезпечення розрахунку вартості та створення поліграфічної продукції – «PolyCall».

1.3. Мета та призначення створення системи

1.3.1. Призначення системи

Програмне забезпечення «PolyCall» створено задля того, щоб підприємець міг оцінити, скільки він витратить грошей, на закупівлю ресурсів, та інших матеріалів, скільки буде коштувати партія поліграфічного товару, та його одиниця.

Програма має працювати під операційною системою (ОС) Windows, мати ефективний інтерфейс користувача.

1.3.2. Мета системи

Мета системи – це дати змогу відразу порахувати, наскільки вигідна така поліграфічна продукція, не звертаючись до великих компаній, та використовуючи багато часу та грошей на їх спеціалістів.

Головна ідея системи – ефективність та мінімалізм, велика гнучкість інформації та її корективів.

					КНТЕУ 121-07-08.БР			
					Розробка системи обробки інформації в поліграфічному підприємстві	Стадія	Аркуш	Арку-
Зм.	Аркуш	№ докум.	Підпис	Дата		Р1	6	41
Зав. каф.	Криворучко О.В.							
Керівник	Жирова Т. О							
Гарант	Цензура М.О.				Основні технічні відомості та технічне завдання	Факультет інформаційних технологій 4 курс, 7 група		
Розроб	Коблай А.С.							

Структурно програма повинна складатись з інтерфейсу користувача, реалізованого через форму, інтерфейсу входу, де це потрібно.

Програма повинна виконувати наступні функції:

- проводити введення вхідної інформації;
- вираховувати необхідну інформацію;
- мати можливість зберігати дані за шаблоном їх введення;
- надавати підказки щодо полів задля кращого розуміння.

Автору відомо декілька методів розв'язання поставленого завдання.

Найбільш простим з точки зору розробки є створення даного програмного забезпечення в програмі, яка дозволяє створювати так звані «Windows Forms», типу «C++ Builder».

Наступним варіантом є створення форми вікна, основних їх властивостей і характеристик на асемблері, проте це занадто довгий процес, і при цьому потрібно володіти дуже великою кількістю інформації, щодо правильного і ефективного управління пам'яттю, правильних і надійних збирачів сміття (це данні, які не використовуються програмою).

Найбільш ефективним рішенням можна вважати оренду групи людей, які охарактеризують те, що потрібно зробити, наймуть спеціалістів в індустрії поліграфії, задля того щоб якомога точніше зробити формули для розрахунку, проте такий спосіб не можна вважати самостійною роботою, та такий спосіб потребує дуже великих грошових вкладень.

Враховуючи викладене, основними перевагами використання "C++ Builder" є:

					КНТЕУ 121-07-08.БР	Аркуш
						7
Зм.	Аркуш	№ докум	Підпис	Дата		

1. Відпадає потреба платити будь-кому за їх працю, тому плата – це тільки наш час, який ми витрачаємо на це;
2. Ми набуваємо корисні навички, сутність яких можна застосовувати в створенні інших програм подібного профілю;
3. Нам не потрібно перечитувати і сприймати за короткий час тонну інформації по тому, щоб на досконалому рівні зрозуміти як працює система на нижніх рівнях, під ОС;
4. Як результат попереднього пункту – набагато менше шансів що при тестуванні ми зможемо отримати помилку в якійсь із частей програмного забезпечення.

1.4. Теоретичні відомості

1.4.1. Windows Forms

Windows Forms – інтерфейс програмування додатків (API), відповідальний за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за допомогою створення обгортки для Win32 API в керуваному коді.

Всередині .NET Framework, Windows Forms реалізується в межах простору імен System.Windows.Forms.

Як і Abstract Window Toolkit (AWT) (схожий API для мови Java), бібліотека Windows Forms була розроблена як частина .NET Framework для спрощення розробки компонентів графічного інтерфейсу користувача. Windows Forms побудована на основі застарілого Windows API.

Windows Forms надає можливість розробки кросплатформного графічного користувацького інтерфейсу. Але Windows Forms є лише обгорткою Windows API-компонентів.

					КНТЕУ 121-07-08.БР	Аркуш
						8
Зм.	Аркуш	№ докум	Підпис	Дата		

1.4.2. База даних

База даних (англ. database) – сукупність даних, організованих відповідно до концепції, яка описує характеристики цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування.

В загальному випадку БД містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна БД, крім самих даних, містить їх опис та може містити засоби для їх обробки.

В загальному випадку БД можна вважати будь-який впорядкований набір даних. Наприклад, паперову картотеку з формулярами про працівників підприємства у відділі кадрів. На даний час додатки для роботи з БД є одними з найпоширеніших прикладних програм.

1.4.3. Система керування базами даних

Система керування базами даних (СКБД) або Система управління базами даних (СУБД) – комплекс прикладного програмного забезпечення, що надає можливості створення, збереження, оновлення та пошуку інформації в БД з контролем доступу до даних.

Централізований характер управління даними в БД передбачає необхідність існування деякої особи (групи осіб), на яку покладаються функції адміністрування даними, що зберігаються в базі. Зазвичай така роль в ІТ називається адміністратор БД але є випадки, коли в невеликих автоматизованих системах функції адміністратора БД виконує системний адміністратор.

БД можливо класифікувати за низкою критеріїв.

За моделлю організації даних БД поділяються на:

- 1) ієрархічні;
- 2) мережеві;

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		9

- 3) реляційні;
- 4) об'єктноорієнтовані.

За територіальним рознесенням даних БД можуть бути локальними (дані зберігаються на одному комп'ютері) або розподіленими (дані зберігаються на декількох комп'ютерах, пов'язаних між собою мережею зв'язку).

За методом фізичного зберігання даних БД поділяються на:

- 1) БД де дані зберігаються в оперативній пам'яті (RAM);
- 2) БД де дані зберігаються на накопичувачах на жорстких магнітних дисках (HDD).

За вмістом БД можуть поділятися на:

- 1) географічні (збереження графічних об'єктів, фотографій);
- 2) мультимедійні (збереження відеофайлів);
- 3) текстові (збереження документів);
- 4) телеметричні (збереження даних від датчиків телекодової інформації);
- 5) комбіновані тощо.

Ієрархічні бази даних можуть бути представлені у вигляді дерева, що складається з об'єктів різних рівнів. Між об'єктами існують зв'язки типу «предок-нащадок». «Нащадок» може мати лише одного «родича». «Нащадки», що мають спільного «родича», називаються «близнюками».

Мережеві БД подібні до ієрархічних, але в таких БД «нащадки» можуть мати більше одного «родича». Основними поняттями такої БД є вузол (елемент), рівень та зв'язок.

Вузол – це сукупність атрибутів даних, що описують певний об'єкт.

Реляційна модель організовує дані у вигляді двовимірних масивів. Кожна таблиця має такі властивості:

- 1) кожен елемент таблиці – один елемент даних;

					КНТЕУ 121-07-08.БР	Аркуш
						10
Зм.	Аркуш	№ докум	Підпис	Дата		

2) всі осередки в стовпчику таблиці однорідні, тобто всі елементи в стовпчику мають однаковий тип (числовий, символний, грошовий тощо);

3) кожен стовпчик має унікальне ім'я;

4) однакові рядки в таблиці відсутні;

5) порядок проходження рядків і стовпчиків може бути довільним.

Для інтерпретації складних даних підходить об'єктна БД. Вона не потребує змін програми для додавання нових типів даних – вони надходять у зовнішні структури БД.

Об'єктно-орієнтована база даних (ООБД) – БД, у котрій дані оформлені у вигляді моделей об'єктів, вони включають в себе програми, що керуються зовнішніми подіями. Результатом при поєднанні такої БД і об'єктно-орієнтованих мов програмування є об'єктно-орієнтовані системи управління базами даних (ООСУБД). Такі системи управління дозволяють працювати з об'єктами БД як з об'єктами у об'єктно-орієнтованих мовах програмування. Це значно розширює можливості застосування шляхом об'єднання переваг, що притаманні, як СУБД, так і об'єктно-орієнтованим мовам програмування одночасно.

Якщо потрібна продуктивна обробка складних даних, то така БД є рекомендованою.

Централізовані бази даних зберігаються на одному комп'ютері. Якщо комп'ютер являється частиною локальної мережі з іншими комп'ютерами, то можливо використовувати розподілений доступ до БД.

Розподілена БД складається з декількох частин, що зберігаються на різних комп'ютерах мережі. Робота з базою даних у такому випадку здійснюється через систему управління розподіленою базою даних (СУРБД) та каналів зв'язку (локальної обчислювальної мережі).

					КНТЕУ 121-07-08.БР	Аркуш
						11
Зм.	Аркуш	№ докум	Підпис	Дата		

Спосіб доступу до БД може бути локальний або віддалений (мережевий).

Існують такі архітектури централізованих БД з віддаленим доступом:

- 1) файл-сервер;
- 2) клієнт-сервер;

Файл-сервер. Архітектура мережевої БД передбачає виділення машини-сервера, тобто центрального комп'ютера, на якому зберігається БД. Інші машини цієї мережі є робочими станціями, за допомогою яких надається доступ користувачеві до централізованої БД. Файли БД, в залежності від запитів користувача, передаються на робочі станції. Основна обробка проводиться на них. Також користувачі можуть створювати локальні БД на своїх робочих станціях, ці БД будуть використовуватися монопольно цими користувачами.

Клієнт-сервер. У цьому випадку сервер має не тільки зберігати БД, а й виконувати основну роботу з нею, відповідно до запитів користувача.

Історично СУБД орієнтувалися на вирішення завдань, пов'язаних, у першу чергу, з транзакційною обробкою структурованої інформації. Найкращим та перевіреним часом рішенням була і залишається реляційна модель СУБД. Однак в останні роки область застосування БД незмінно розширюється.

З одного боку, потрібно керувати більш широким набором форматів даних, переходячи до вирішення спільних проблем управління корпоративною інформацією. З іншого – саме СУБД беруть на себе основні функції інтеграції даних і додатків корпоративних систем. Саме цим пояснюється активний інтерес до обговорення архітектурних принципів і можливостей реалізації БД різних моделей – постреляційних, об'єктно-реляційних, XML.

За останні роки з'явилися нові важливі області застосування БД, кожна з яких представляє принципово нове середовище, до якого потрібно адапту-

					КНТЕУ 121-07-08.БР	Аркуш
						12
Зм.	Аркуш	№ докум	Підпис	Дата		

вати технології СУБД. Серед них є інтелектуальний аналіз інформації (data mining), сховища та репозиторії інформації.

Інтелектуальний аналіз інформації – це сучасна концепція, що припускає неточності, суперечливості та великі обсяги інформації. Саме тому для їх розуміння потрібні великі інтелектуальні зусилля. Це обробка інформації та виявлення в ній певних моделей, тенденцій, що допомагатимуть користувачам приймати рішення. Необхідність таких БД виникла у наслідок тотального поширення інформаційних технологій, що дозволяють детально протоколювати процеси бізнесу, виробництва тощо. Самим яскравим прикладом таких БД є модуль Google Analytics.

Сховища даних. У них зберігаються дані з однієї або декількох БД.

Репозиторії. Програми, що належать до категорії репозиторіїв призначені не тільки для зберігання і управління даними, а й метаданими, інформацією про структуру даних. Наприклад, система управління документами, чи БД для підтримки комп'ютерного проектування. Характерною рисою є умовно часті зміни метаданих.

Одна і та ж інформація у репозиторії має багато уявлень. Програмний модуль, наприклад, має уявлення у вигляді вихідного та об'єктного кодів, проміжного коду, готової програми тощо. Зв'язки між усіма уявленнями відстежуються так, що зміна в одному з них одразу поширюється на усі інші. Репозиторії також повинні підтримувати поняття версій. Приклад - репозиторій ПЗ, що містить повний перелік ПЗ, їх версії, оновлення (патчі) тощо.

Сховище повинне підтримувати еволюцію структури інформації та її метаданих таким чином, щоб при додаванні нових властивостей даних або нових зв'язків не була потрібна повна перекомпіляція.

					КНТЕУ 121-07-08.БР	Аркуш
						13
Зм.	Аркуш	№ докум	Підпис	Дата		

MySQL – вільна СУБД, власність компанії Oracle. Цю систему з відкритим кодом (open source) було створено як альтернативу комерційним системам. Зараз вона одна із найпоширеніших СУБД. Має чудову підтримку з боку різних мов програмування, її часто використовують для створення динамічних веб-сторінок. Найчастіше її використовують як сервер, яким можуть користуватися локальні та віддалені клієнти, але ця система також має систему внутрішнього сервера. Це дозволяє вбудовувати БД в автономну програму. MySQL надається безкоштовно для некомерційного використання.

Основні переваги СУБД MySQL:

- 1) простота інсталяції та використання;
- 2) підтримка умовно необмеженої кількості користувачів, що одночасно працюють з БД (умовно, бо визначається технічними характеристиками сервера ,у першу чергу оперативною пам'яттю та продуктивністю процесора);
- 3) величезна кількість рядків у таблиці (може досягати 50 мільйонів);
- 4) короткий час виконання команд (швидкодія);
- 5) наявність простої та ефективної системи безпеки.

PostgreSQL – об'єктно-реляційна СУБД, інша альтернатива комерційним та відкритим СУБД. На відміну від інших СУБД, її не контролює якась певна компанія – вона розробляється та покращується багатьма людьми та компаніями.

СУБД Oracle. Один із найпотужніших комплексів роботи з БД. За її допомогою можна створювати додатки будь-якої складності. Вона майже не має обмежень на кількість користувачів та інформації (розмір визначається технічними характеристиками серверного обладнання). Також вона не проявляє тенденції зменшення працездатності (швидкодії) при різкому збільшенні кількості користувачів.

					КНТЕУ 121-07-08.БР	Аркуш
						14
Зм.	Аркуш	№ докум	Підпис	Дата		

Остання версія механізму масштабування дозволяє безмежно збільшувати потужність та швидкість сервера, без зупинки працюючих додатків. Також, для цього не потрібно переписувати старі додатки, розроблені для одомашинної архітектури. Oracle працює на більшості популярних операційних систем (ОС), серед яких Sun Solaris, Windows, Linux. На різних ОС стабільність роботи СУБД Oracle підтверджена багаторічним досвідом.

Microsoft SQL Server 2017 – система управління реляційними БД. Розробник – корпорація Microsoft. Основною мовою для запитів є Transact-SQL. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів із розширеннями. Використовується як з малими персональними БД, так і з великими БД масштабу підприємства. При взаємодії з мережею, Microsoft SQL Server використовує протокол передачі табличних даних (Tabular Data Stream, TDS).

Для забезпечення доступу до даних Microsoft SQL Server 2017 підтримує Open Database Connectivity (ODBC) – це інтерфейс, за допомогою якого додатки взаємодіють із СУБД. Також Microsoft SQL Server 2017 надає можливість використання веб-сервісу для підключення, тобто користувач може підключатися навіть із іншої операційної системи за допомогою спеціального додатку.

Також корпорації Microsoft належить Microsoft Office Access, реляційна СУБД з великими можливостями, серед яких є навіть зв'язані запити та зв'язок з іншими таблицями, базами даних. У Microsoft Office Access використовується мова VBA, вона дозволяє писати додатки у самій СУБД.

Основні компоненти MS Access:

- 1) будівник таблиць;
- 2) будівник екранних форм;
- 3) будівник SQL-запитів (мова SQL в MS Access не відповідає стандарту ANSI);

					КНТЕУ 121-07-08.БР	Аркуш
						15
Зм.	Аркуш	№ докум	Підпис	Дата		

4) будівник звітів, що виводяться на екран, до файлу або на принтер.

Вони можуть викликати «скрипти» (мова сценаріїв, англ. scripting language) мовою VBA, тому MS Access дозволяє розробляти програми і БД практично «з нуля» або написати оболонку для зовнішньої БД.

MS Access має архітектуру БД файл-сервер, тому застосовується лише для маленьких додатків.

MS Access дозволяє створювати повноцінні клієнт-серверні додатки на MS SQL Server 2017.

1.4.4. Обґрунтування вибору мови програмування

В даній дипломній роботі в якості БД обрана Microsoft SQL. Причин такого вибору багато, але основними з них є:

- 1) великий функціонал;
- 2) робота під управлінням найбільш розповсюдженої ОС Windows;
- 3) велика кількість інструкцій та іншої методичної літератури;
- 4) можливість горизонтального масштабування (має вбудовані функції);
- 5) наявність простої мови програмування;
- 6) вбудовані механізми захисту даних від втрати (дублювання, резервне копіювання);
- 7) вбудовані механізми інформаційної безпеки (шифрування даних, автентифікація користувачів, розрахунок контрольних сум тощо);
- 8) наявність платформи для обробки даних веб-сайтів тощо.

В даному дипломному проєкті під час практичної реалізації прикладного ПЗ автором використано безкоштовну версію Express, що представляє собою БД початкового рівня, яка ідеально підходить для навчання і створення додатків для обробки даних на настільних комп'ютерах і невеликих серверах (розміром до 10 ГБ).

					КНТЕУ 121-07-08.БР	Аркуш
						16
Зм.	Аркуш	№ докум	Підпис	Дата		

Microsoft SQL Server 2017 у якості мови запитів використовує версію SQL, що отримала назву TRANSACT-SQL (або просто T-SQL), яка є реалізацією SQL-92 (стандарт ISO для SQL) з багатьма розширеннями. T-SQL дозволяє використовувати додатковий синтаксис процедур, що зберігаються і забезпечує підтримку транзакцій (взаємодія БД з керуючим додатком).

Microsoft SQL Server 2017 та Sybase ASE для взаємодії з мережею використовують протокол рівня додатків під назвою Tabular Data Stream (TDS, протокол передачі табличних даних).

Microsoft SQL Server 2017 також підтримує Open Database Connectivity (ODBC) – інтерфейс взаємодії застосунків з СУБД. Починаючи з версії SQL Server 2017 є можливість підключення користувачів через веб-сервіси, що використовують протокол SOAP. Це дозволяє клієнтським програмам, не призначеним для Windows, використовуючи інші операційні системи з'єднуватися з SQL Server 2017.

Microsoft також випустила сертифікований драйвер JDBC, що дозволяє додаткам під керування Java (таким як BEA і IBM Websphere) з'єднуватися з Microsoft SQL Server 2017.

SQL Server 2017 підтримує дзеркалювання та кластеризацію БД. Клас-тер серверу SQL – це сукупність однаково конфігурованих серверів. Така схема допомагає розподілити робоче навантаження між декількома серверами. Усі сервери мають одне віртуальне ім'я, а дані розподіляються за IP-адресами машин кластеру протягом робочого циклу. Також у разі відмови або збою на одному із серверів кластеру доступне автоматичне перенесення навантаження на інший сервер.[3]

Для запобігання втрати даних SQL Server 2017 підтримує надлишкове дублювання даних за такими трьома сценаріями:

					КНТЕУ 121-07-08.БР	Аркуш
						17
Зм.	Аркуш	№ докум	Підпис	Дата		

1) Знімок: Виконується «знімок» БД, який сервер відправляє одержувачам.

2) Історія змін: Всі зміни БД безперервно передаються користувачам.

3) Синхронізація з іншими серверами: БД декількох серверів синхронізуються між собою. Зміни усіх баз даних відбуваються незалежно на кожному сервері, а під час синхронізації відбувається звірка даних. Дублювання такого типу передбачає можливість вирішення протиріч між БД.

SQL Server 2017 має вбудовану підтримку .NET Framework. Завдяки цьому, процедури БД, що зберігаються, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server 2017 не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server 2017.

Слід звернути особливу увагу на основний засіб розробки та адміністрування, яке включене до складу дистрибутива та дозволяє вирішувати практично всі завдання адміністрування MS SQL Server 2017 і є зручним для розробника. Але головна перевага даної СУБД, навіть не в зручному візуальному інструментарії, що входить до складу дистрибутива, а в тісній інтеграції її з іншими програмними продуктами від Microsoft.

Крім того, істотною перевагою цієї СУБД перед іншими є можливість експорту та імпорту в більшість поширених форматів даних, що включає як клієнт-серверні та файл-серверні, так і XML формат.

Враховуючи викладене, перевагу віддано СУБД MS SQL Server 2017, так як вона дозволяє зберігати великий обсяг даних, здійснює тісну інтеграцію з іншими програмними продуктами від Microsoft.[3]

					КНТЕУ 121-07-08.БР	Аркуш
						18
Зм.	Аркуш	№ докум	Підпис	Дата		

Для створення інтерфейсу користувача була обрана платформа C++ Builder, і як можна зрозуміти з назви, мова програмування C++.

C++ Builder – програмний продукт, інструмент швидкої розробки додатків (RAD), інтегроване середовище розробки (IDE), система, яка використовується програмістами для розробки програмного забезпечення на мові програмування C++.

Спочатку розроблявся компанією Borland Software, а потім її підрозділом CodeGear[en], який сьогодні належить компанії Embarcadero Technologies.

C++ Builder об'єднує в собі комплекс об'єктних бібліотек (STL, VCL, CLX, MFC та ін.), компілятор, зневаджувач, редактор коду та багато інших компонентів. Цикл розробки аналогічний Delphi. Більшість компонентів, розроблених в Delphi, можна використовувати і в C++ Builder без модифікації, але зворотнє твердження не вірне.

C++Builder спочатку створювався тільки для платформи Microsoft Windows. Пізні версії, які містили крос-платформову компонентну бібліотеку Borland, підтримують і Windows, і Linux.

В 2003 році Borland випустила C++BuilderX (CBX), написаний за допомогою тієї ж інфраструктури, що і JBuilder[en], який при цьому був мало схожим на C++ Builder або Delphi. Цей продукт призначався для розробки великих програм для великих компаній, але комерційного успіху не мав. В кінці 2004 року Borland оголосила, що продовжить розвиток класичного C++ Builder і об'єднає його з середовищем розробки Delphi, припинивши, таким чином, розробку C++ BuilderX.

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		19

Приблизно через рік після цього оголошення, Borland випустила Borland Developer Studio 2006, який включав у себе Borland C++Builder 2006, що пропонував покращене керування конфігурацією та зневадженням. Borland Developer Studio 2006 – єдиний повноцінний комплект, який містив Delphi, C++ Builder та C# Builder.

В 2007 році CodeGear випустила C++Builder 2007, в якому реалізувала повну підтримку API Microsoft Windows Vista, збільшила повноту відповідності стандарту ANSI C++, збільшила швидкість компіляції і збірки до 500 %, включила підтримку MSBuild, архітектур баз даних DBX4 і «VCL для Web», які підтримують AJAX. Підтримка API Microsoft Windows Vista включила в себе додатки, з самого початку оформлені в стилі Vista, і природну підтримку VCL для Aero та Vista Desktop. CodeGear RAD Studio 2007 містить C++Builder 2007 і Delphi. Також в 2007 році CodeGear «воскресила» марку «Turbo» і випустила дві «Turbo» версії C++ Builder: Turbo C++ Professional і Turbo C++ Explorer (безкоштовний), які базуються на Borland C++ Builder 2006.

В кінці 2008 року компанія CodeGear випустила нову версію RAD Studio, в яку увійшли Delphi 2009 і C++ Builder 2009. В 2009 році у складі RAD Studio вийшов C++Builder 2010.

Раніше повідомлялося, що наступна версія, CodeGear C++ Builder (кодове ім'я «Commodore»), буде мати підтримку x86-64 і можливість створювати машинний x86-64 код. Однак в 2010 році у складі RAD Studio XE включена версія C++ Builder XE без цієї функціональності.

В 2012 році Embarcadero випустила C++ Builder XE3, сумісний з Windows 8. В 2013 році був випущений C++ Builder XE4. [2]

1.5.Вимоги до системи

Система повинна рахувати точно, враховувати всі можливі данні та коефіцієнти, що будуть враховувати тип продукції, тип паперу, та їх витрати.

					КНТЕУ 121-07-08.БР	Аркуш
						20
Зм.	Аркуш	№ докум	Підпис	Дата		

1.6.Висновки до розділу 1

В даному розділі автор розглянув технічне завдання, можливості його створення, та вимоги, які були створені в процесі розробки, та розписав, які програми він використовував в своїй роботі.

					<i>КНТЕУ 121-07-08.БР</i>	Аркуш
						21
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>		

РОЗДІЛ 2.

АНАЛІЗ ПРЕДМЕТА ДОСЛІДЖЕННЯ ТА ХАРАКТЕРИСТИКА ЙОГО РОБОТИ

2.1. Опис системи «C++ Builder» Система об'єктно-орієнтованого програмування C++ Builder виробництва корпорації Borland призначена для операційних систем Windows 9x–XP. Вона функціонує під керуванням ОС типу Microsoft Windows.

Інтегроване середовище C++ Builder цілком підтримує стандарт мови C++, завдяки чому існує можливість створення за допомогою цієї системи програмування модулів і бібліотек, що застосовуються в інших засобах розробки.

Структура класів, створених мовою C++ системи програмування Borland C++ Builder, побудована в бібліотеці VCL (Visual Component Library) і містить потужний набір засобів та бібліотек, що підвищують продуктивність праці програмістів і скорочують тривалість циклу розроблення.

Багатофункціональне інтегроване середовище C++ Builder містить компілятор, який відповідає вимогам стандарту ANSI/ISO, вбудований дизайнер форм, багатий набір засобів для роботи з компонентами, інструмент Object Inspector, менеджер проектів і налагоджувач. Новий багатоцільовий менеджер проектів з відкритою архітектурою забезпечує повний контроль над вихідними текстами і процесом компоновки, що дозволяє переглядати залежності модулів вихідного коду і налаштовувати параметри компіляції, компоновки і налагодження для кожного з модулів проекту.

					КНТЕУ 121-07-08.БР			
					Розробка системи обробки інформації в поліграфічному підприємстві	Стадія	Аркуш	Аркушів
Зм.	Аркуш	№ докум.	Підпис	Дата		Р2	22	41
Зав. каф.	Криворучко О.В.							
Керівник	Жирова Т. О							
Гарант		Цензура М.О.			Аналіз предмета дослідження та характеристика його роботи	Факультет інформаційних технологій 4 курс, 7 група		
Розроб		Коблай А.С.						

За допомогою цього середовища можна створити додаток будь-якої складності з привабливим графічним інтерфейсом.

Інспектор об'єктів необхідний для завдання властивостей об'єктів і визначення їхньої реакції на різні події. Властивість об'єкта – це одна з його характеристик, наприклад, ширина для кнопки, назва для вікна, колір і стиль для шрифту. Інспектор об'єктів дозволяє швидко і зручно змінювати будь-які влас-тивості поточного, тобто виділеного на формі, об'єкта.

Інспектор об'єктів складається з декількох частин і містить дві вкладки: вкладка властивостей обраного об'єкта (Properties) і вкладка подій (Events), що дозволяє програмі реагувати на будь-які дії користувача так, як це задано програмістом. Кожна з вкладок містить панель із двох стовпчиків. Вкладка Properties має у першому стовпчику назви властивостей, а в другому – поточні значення відповідних властивостей. У першому стовпчи-ку вкладки Events знаходиться назва події, а другому – функ-ція обробки цієї події.

Уся робота в середовищі C++ Builder здійснюється у вікнах, основним з яких є форма, і програмування ґрунтується на взаємодії двох процесів:

- конструювання форми, тобто розміщення на ній необхідних елементів (компонентів) керування розв'язанням задачі;
- написання програмного коду, який забезпечує функціонування цих компонентів.

Згідно з термінологією C++ Builder весь комплекс програм, який створюється при розв'язанні конкретної задачі, є проектом.

Проект - це основна програма, котра цілком створюєть-ся системою і записується у файл з розширенням .dpr. Вона містить інструкції, з яких починається виконання додатка, а та-кож відповідає за проведення процесу компіляції і компонування. В проекті не рекомендується робити будь-які зміни.

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		23

Програма складається з одного чи більше модулів з розширенням .cpp, які містять програмні коди. Крім того, для кожного додатка автоматично створюється ще один файл, що називається заголовним або інтерфейсним і має розширення .h. Для кожного програмного додатка система створює мінімально необхідну програму, тобто пропонує заготовку для написання програмного коду, а користувач повинен записати в цю заготовку свої інструкції, котрі реалізують роботу заданого алгоритму. Враховуючи вище сказане, доцільно зробити висновок, що C++Builder асоціює з кожним додатком три вихідних файли з такими іменами за замовчуванням:

файл реалізації Unit1.cpp – зберігає програмний код додатка. Саме в ньому записуються оброблювачі подій, які відповідають за реакцію програми при впливі користувача на об'єкти компонентів;

заголовний файл Unit1.h – містить список всіх об'єктів і їхніх конструкторів;

файл Project1.cpp – обслуговує всі об'єкти, що укладені в додатку. Будь-яка нова форма, програмний модуль чи модуль даних автоматично включаються в проектний файл. Зміст вихідного тексту проектного файла можна переглянути у вікні редактора коду за допомогою команди головного меню View | Project Source або вибравши одноіменну опцію з контекстного меню адміністратора проекту.

Для створення додатка треба спочатку на формі розмістити компоненти, що забезпечують введення вихідних даних, їхню обробку і виведення отриманого результату, а потім у вікні програмного коду записати оператори мови C++, які забезпечують функціонування цих компонентів.

У заготовці програмного коду коментарі зазначають місця, де користувач за необхідності може розміщати свої інструкції.

					КНТЕУ 121-07-08.БР	Аркуш
						24
Зм.	Аркуш	№ докум	Підпис	Дата		

На цій заготовці ще не відведено місця для операторів про-грами. Воно буде створене при розміщенні на формі компонента, який породжує подію.

Ключове слово `__fastcall` в оголошеннях функцій обробки подій генерується автоматично. Завдяки `__fastcall` передача па-раметрів організується не через стек, а через регістри централь-ного процесора. Виклики оброблювачів подій відбуваються ду-же часто, тому економія часу, який витрачається на вибірку параметрів з пам'яті стека, виявляється дуже відчутною.[2]

2.1. Конкурентність програми

Автор провів аналітику, та виявив, що всі програми, що були створені для розрахунків такого типу – являються локальними на підприємстві, та їх не поширюють в інтернеті.

Пару компаній використовують модифікацію до системи «1С», проте це зазвичай вже готові данні, і все, що вводить людина – це лише тираж, себ-то кількість продукції, тому можна вважати, що програма автора – одна з ду-же небагатьох, які були створені з гнучким персоналом

Вариант №1

Tisoft.ru

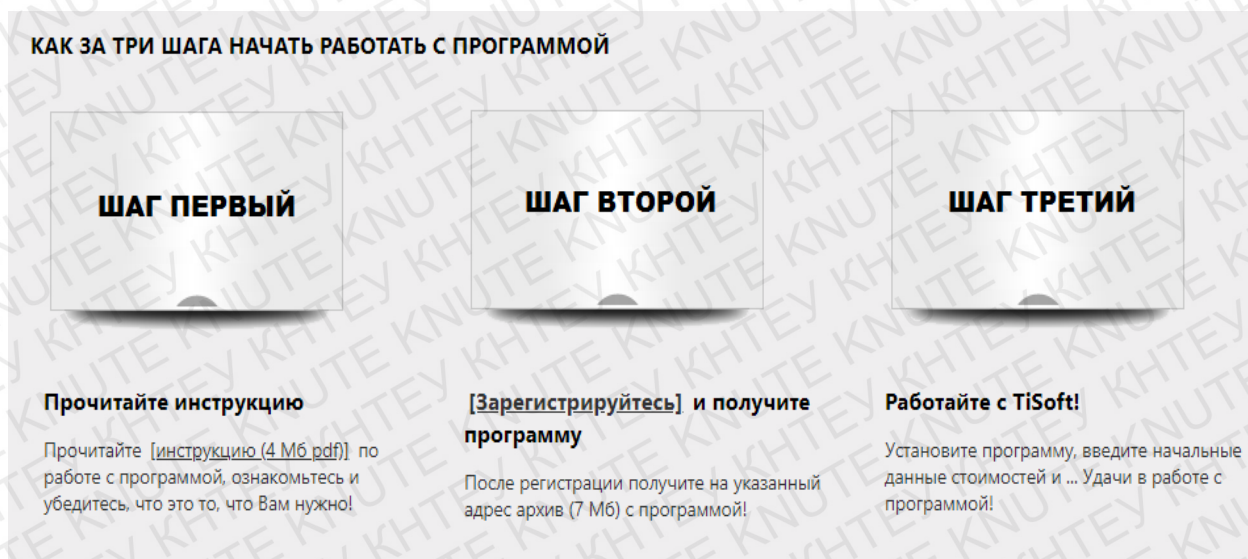


Рис. 2.1. Конкурентна програма №

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		25

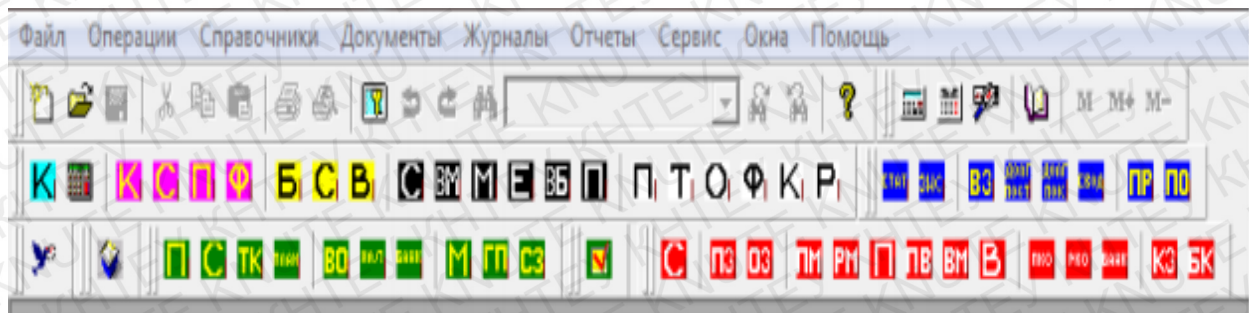


Рис. 2.2. Интерфейс конкурента

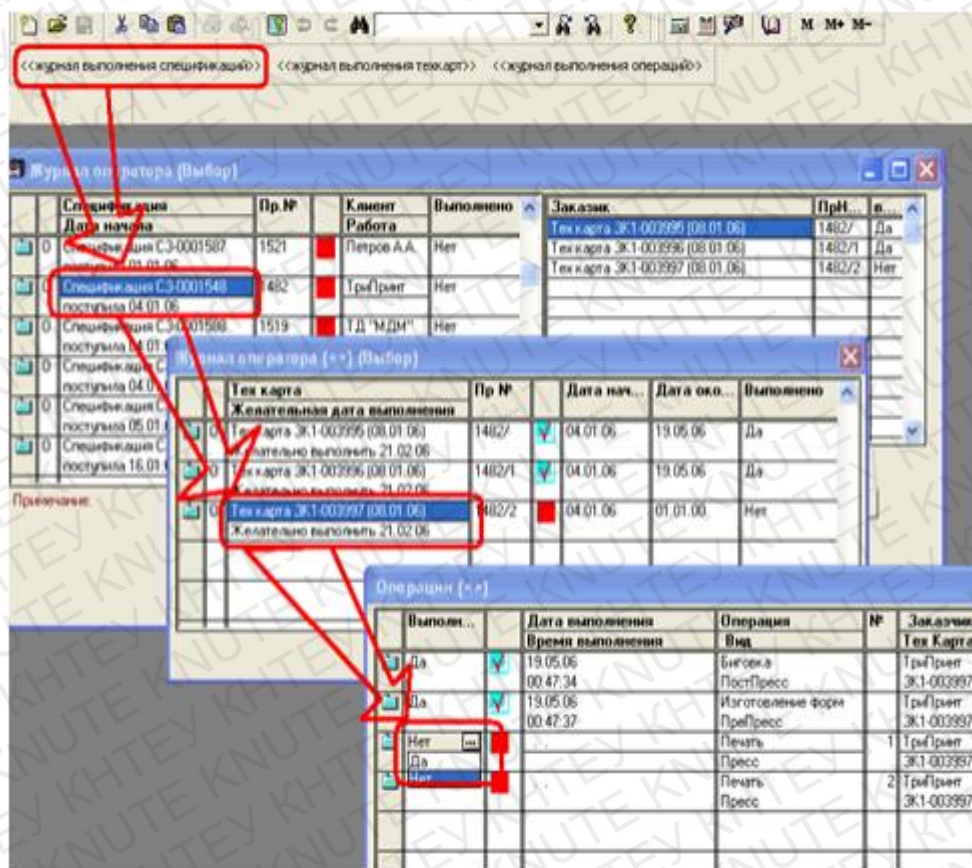


Рис. 2.3. Выполнения работы

Прочитавши інструкцію (рис. 2.1.) можна отримати доступ до програми конкурента та оцінити його інтерфейс (рис. 2.2.).

Автор вважає, що програма конкурента доволі конкурентоздатна, проте, дуже накопичена різними розрахунками (рис. 2.3.), через те, що створена не тільки для розрахунків, а й для того, щоб використовуватись в бухгалтерії,

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		26

маючи доволі багато спеціалізованих для бухгалтерії вікон. Тому ефективність даної програми можна вважати середньою, із-за великого накопичення різної інформації, що являє собою більш спеціалізовану програму, для якої потрібні спеціалізовані навички.

Форма расчета стоимости заказа

СОВЕТ!!

Авторизуйтесь, чтобы задействовать все возможности калькулятора!

Общие параметры

Выберите вид продукции*	Книга в мягкой обложке, брошюра, журнал
Формат в мм. (ширина)*	60
Формат в мм.(высота) *	90
В блоке имеются элементы на вылет?	☐
Обложка имеет элементы на вылет?	☐
Тираж*	50000
	Блок
Способ скрепления*	2 скобы
Количество ч.б. полос (Не более 80 ч.б. и цв.)*	60
Количество цветных полос (Не более 80 ч.б. и цв.)*	60
Бумага на блок*	Мелованная глянцевая 90 гр./кв.м
	Обложка
Красочность обложки*	4+1
Бумага на обложку*	Лен холст желтый 280 гр./кв.м.
Ламинация обложки*	Матовая
Предварительный расчет	

Рис. 2.4. Програма конкурента №2

У даного конкурента можна побачити вже більш легку та зрозумілу версію програми, подібну до програми автора, проте як і в попереднього конкурента – вони фактично заставляють користувачів зареєструватись на їхньому сайті (рис. 2.4.), щоб і покращити свою якість роботи, показуючи всі функції калькулятора (рис. 2.5.).

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		27

Форма расчета стоимости заказа

Общие параметры

Выберите вид продукции*	Книга в мягкой обложке, брошюра, журнал
Формат в мм. (ширина)*	60
Формат в мм.(высота) *	90
В блоке имеются элементы на вылет?	☐
Обложка имеет элементы на вылет?	☐
Тираж*	50000
Блок	
Способ скрепления*	2 скобы
Количество ч.б. полос (Не более 80 ч.б. и цв.)*	6
Количество цветных полос (Не более 80 ч.б. и цв.)*	60
Бумага на блок*	Мелованная глянцевая 90 гр./кв.м
Обложка	
Красочность обложки*	4+1
Бумага на обложку*	Лен холст желтый 280 гр./кв.м.
Ламинация обложки*	Матовая
Предварительный расчет	

цена экземпляра издания = 32.05
общая стоимость издания = 1602383.10

Рис 2.5. Обрахунок в програмі конкурента

Проте, обрахунок, вже навіть з авторизацією лиш дозволяє зрозуміти, скільки потрібно грошей щоб замовити необхідну поліграфічну продукцію, причому тільки в них на сайті, тому фактично, їх програма дозволяє тільки ціну за 1 екземпляр, і суму за весь тираж, не показуючи таких важливих параметрів як:

- Ціна паперу
- Кількість витрат паперу(початкова бухгалтерія)
- Заготовка і використання шаблону

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		28

2.2. Висновки до розділу 2

Виходячи з всього вищесказаного, можна зробити висновок, що програма автора конкурентно спроможна, тому що відображає необхідну користувачу інформацію, показуючи як основну інформацію, таку як ціна за тираж, за 1 екземпляр, і кількість використаного паперу, як за весь тираж, так і розписано, скільки використовується на технічні необхідності, та інші параметри, які встановив користувач.

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		29

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ ОБРОБКИ ІНФОРМАЦІЇ НА ПОЛІГРАФІЧНОМУ ПІДПРИЄМСТВІ

3.1. Програмне забезпечення «PolyCall»

Програмний засіб «PolyCall» призначений для розрахунків на поліграфічному підприємстві, а саме:

- Суцільну кількість витрат паперу за партію
- З них кількість на саму продукцію
- Кількість на технічні необхідності
- Норму на 1000 м² папіру
- Кількість папіру на 1 номер виробу
- Вартість друку
- Вартість друку без ПДВ
- Загальна вартість всієї продукції
- Загальна вартість всієї продукції без ПДВ
- Ціна за 1 штуку з папером

А також зберігання шаблону розрахунків, для подальшого його легкого використання.

3.2. Діаграми процесів програми

По діаграмі нижче можна побачити, як працює програма (рис. 3.1.)

					КНТЕУ 121-07-08.БР			
					Розробка системи обробки інформації в поліграфічному підприємстві	Стадія	Аркуш	Аркушів
						РЗ	30	41
Зм.	Аркуш	№ докум.	Підпис	Дата	Розробка програмного забезпечення обробки інформації на поліграфічному підприємстві	Факультет інформаційних технологій 4 курс, 7 група		
Зав. каф.		Криворучко О.В.						
Керівник		Жирова Т. О						
Гарант		Цензура М.О.						
Розроб		Коблай А.С.						

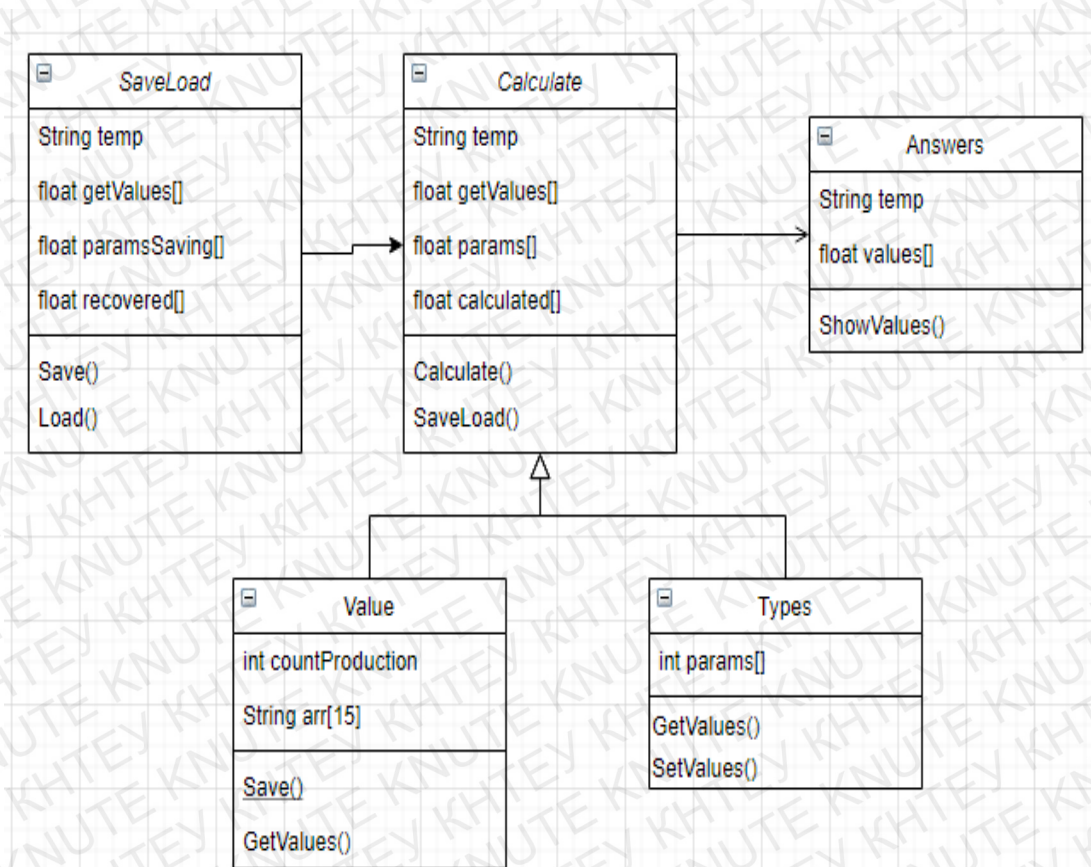


Рис. 3.1. Схема роботи

В наступній діаграмі описана діаграма станів.

Вона описує роботу методики збереження шаблонів в програмі, що дозволяє використовувати створений шаблон при подальшій роботі.

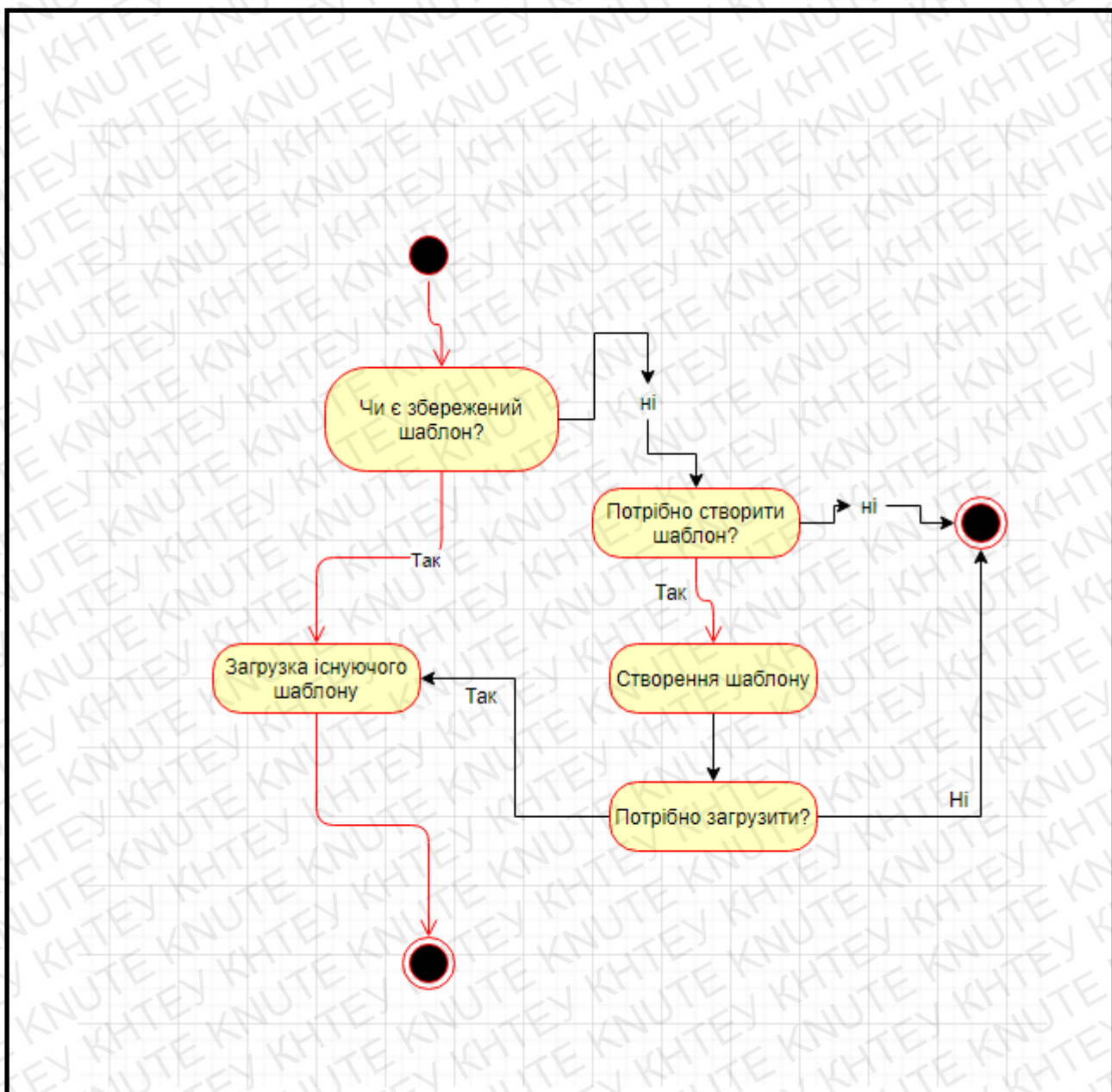


Рис. 3.2. Діаграма станів

На рис. 3.2. описана діаграма станів процесу збереження та загрузки вибраного шаблону, якщо він існує.

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		32

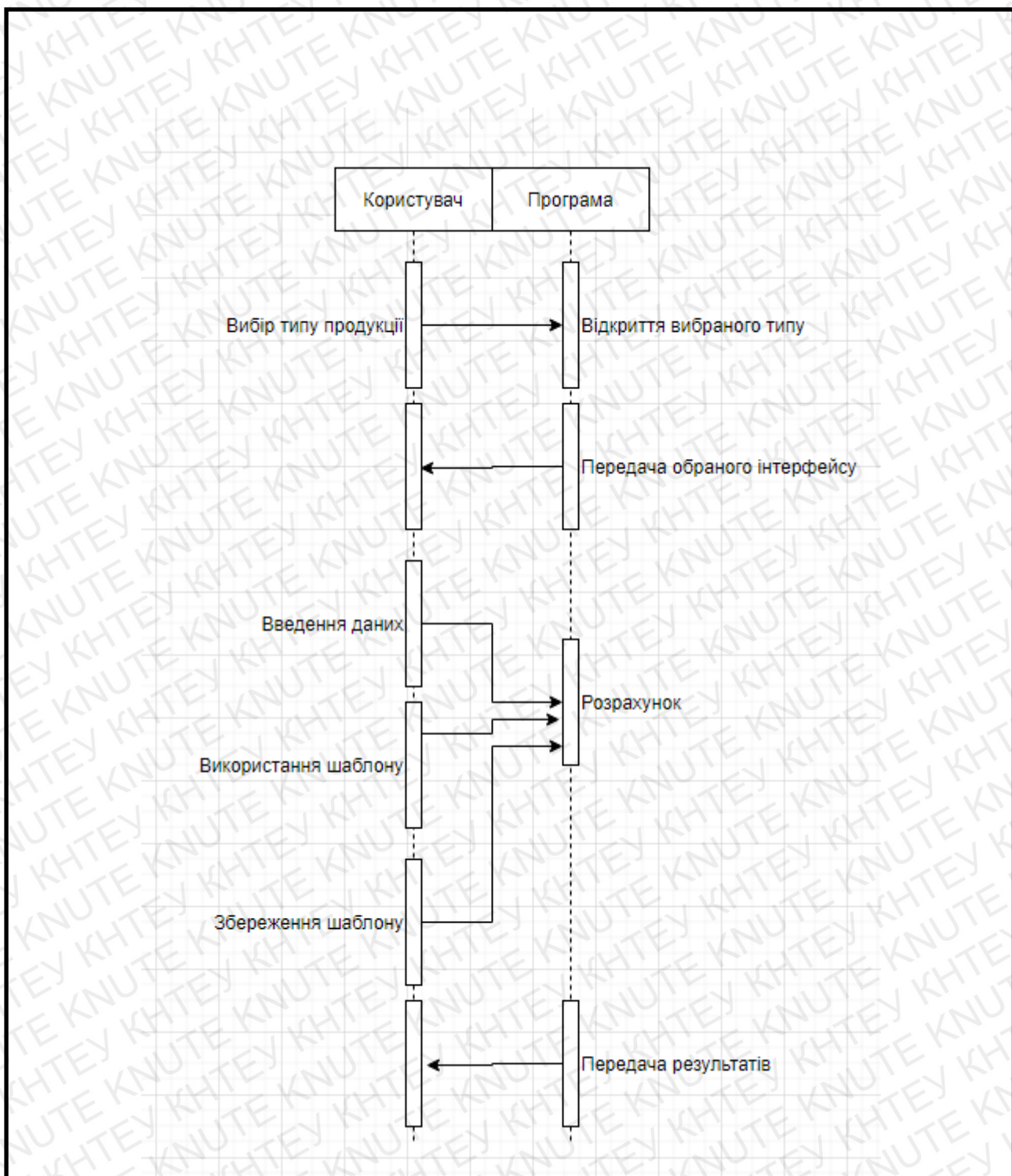


Рис. 3.3. Діаграма послідовностей

На діаграмі послідовностей (рис. 3.3.) зображено, яким чином працює програма з користувачем.

3.3. Інтерфейс програми



Рис. 3.4. Меню вибору типу продукції

На рис. 3.4. зображено меню, яке зустрічає користувача коли він відкриває програму.

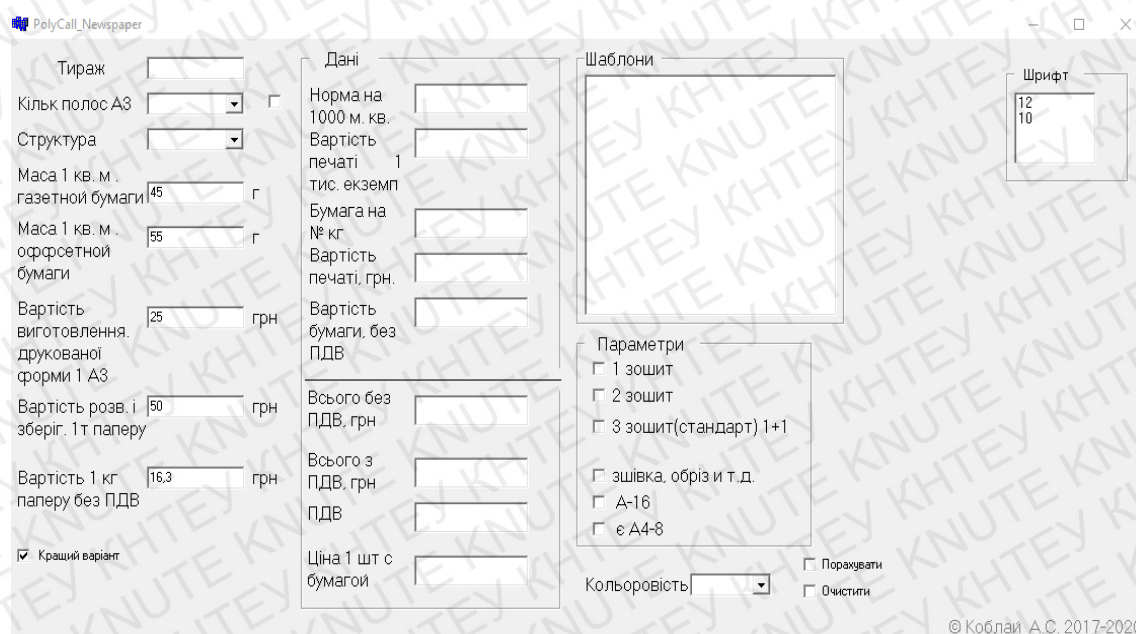


Рис. 3.5. Інтерфейс вікна «Газети»

На рис. 3.5. користувач бачить декілька груп даних, а саме:

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		34

- Ліва група даних, це данні, які вводяться, напряму
- Чек-бокс «Кращий варіант», який створений для скорочення в більшу сторону, щоб у того, хто буде створювати продукцію даного типу не було помилок с людським фактором, і програма рахувала трошки більше, а не в 0%, не враховуючи людську похибку.
- Група даних «Дані», що відповідає за данні розрахунків, які користувач отримує при вводі даних(рис 3.6.)
- Група «Вже використані», яка має в собі шаблони компанії, та шаблон, з ім'ям, яке було йому дано(в даний момент – «Shablon»)
- Та група «Параметри», в якій користувач задає необхідні йому параметри, і задає колір бумаги, який використовується
- Та пару кнопок «Готово», для розрахунків, та «Очистити», для того щоб стерти вхідні та вихідні дані

The screenshot shows the 'PolyCall_Newspaper' application window. It is divided into several sections:

- Тираж (Run):** Includes fields for 'Кільк полос АЗ' (44), 'Структура' (4(2+1)40(1+1)), 'Маса 1 кв. м. газетной бумаги' (45 г), 'Маса 1 кв. м. офсетной бумаги' (55 г), 'Вартість виготовлення друкованої форми 1 АЗ' (25 грн), 'Вартість розв. і зберіг. 1 т паперу' (50 грн), and 'Вартість 1 кг паперу без ПДВ' (163 грн). There is a checked checkbox for 'Кращий варіант'.
- Дані (Data):** Includes 'Норма на 1000 м. кв.' (1035,8 кг), 'Вартість печаті 1 тис. екземп' (48,6 грн), 'Бумага на № кг' (2331), 'Вартість печаті, грн.' (2946,55), 'Вартість бумаги, без ПДВ' (37995,3), 'Всього без ПДВ, грн' (40941,85), 'Всього з ПДВ, грн' (67130,22), 'ПДВ' (11188,37), and 'Цена 1 шт с бумагой' (1,343).
- Шаблоны (Templates):** A large empty box for selecting a template.
- Параметры (Parameters):** Includes checkboxes for '1 зошит', '2 зошит', and '3 зошит(стандарт) 1+1' (checked). There are also checkboxes for 'зшивка, обріз и т.д.' (checked), 'A-16', and 'с А4-8'.
- Шрифт (Font):** A small window showing font selection options (12, 10).
- Кольоровість (Color):** A dropdown menu set to '(2+0)'.
- Buttons:** 'Порахувати' (Calculate), 'Очистити' (Clear), and 'Зберігти' (Save).
- Footer:** Copyright notice '© Коблай А.С. 2017-2020'.

Рис. 3.6. Введені дані та результат

Також, на обох вікнах програми, як «Газети», так і «Книги», є підказка, за що відповідає те чи інше поле, і для того щоб її показати, потрібно натиснути праву кнопку миші(ПКМ).

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		35

На рис. 3.7. відображено вікно «Книги»

PolyCall_Books

Дані_введення

Тираж

Кільк стр

Тип

Бумага

Бумага грн/кг

Розмір листу

Тип машини

Група складності

☐ 1-а

☐ 2-а (основна)

Обкладинка

Тип

Бумага гр/м2

Бумага грн/кг

Доп послуги

☐ Шиття чи скоба

☐ Фальцювання

☐ Різка

☐ Приклейка форзац

☐ Картон

Обкладинка

Кольоровість

Тип

☐ Порахувати

☐ Очистити

Дані_Обраховані

Друк тексту

Бумаги на тираж

На технічні потреби

на приладження

на печать

на палітурку

Разом

Обкладинка

Всього бумаги

листів на тираж

приладжені

печать

Разом тех потрб

лакування/плівка

Картон

Всього на тираж

на обкладинку

листів

Кг

відходи

Плівка

потрібно

Ціна

ціна 1 шт

© Коблай А.С. 2017-2020

Рис. 3.7. Інтерфейс вікна «Книги»

На даному інтерфейсі видно 2 групи даних:

- Дані для введення(ліва сторона)

					КНТЕУ 121-07-08.БР	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		36

- І дані, які бачить користувач, після вводу даних і натискання кнопки «Готово» (рис. 3.8.).

Задля покращення розрахунку була додана більшість факторів, які допоможуть якомога точніше провести розрахунок в програмі, так як ця програма створена не для конкретної фірми, де вже зрозуміло більшість з додаткових параметрів, наприклад, таких як:

- Тип обкладинки
- Колірність обкладинки
- Кількість полос в обкладинці
- Ціна на кількість грам бумаги на метр квадратний
- Тип та розмір бумаги
- Група складності поліграфічних виробів
- Розмір листу та тип машини.

					КНТЕУ 121-07-08.БР	Аркуш
						37
Зм.	Аркуш	№ докум	Підпис	Дата		

Рис. 3.8. Вікно «Книги» з введеними даними

					<i>КНТЕУ 121-07-08.БР</i>	Аркуш
						38
Зм.	Аркуш	№ докум	Підпис	Дата		

В цілях, щоб користувач не сказав, що програма рахує невірно, порахувавши, і потім змінивши цифри, які він вписав спочатку, програма блокує зміну даних введення, допоки користувач не натисне кнопку «Очистити».

І так само, як і в інтерфейсі «Газети», в інтерфейсі «Книги» є підказки до полів, тому користувач може використовувати підказку до полів.

Можна побачити, цифри що вводяться трішки більші по шрифту, ніж основний текст. Це було зроблено тому, що користувачі скаржились, що цифри погано видно, тому було прийнято рішення зробити дані що вводяться більше за розміром.

3.4. Висновки до розділу 3

В даному підрозділі було розглянуто програмне забезпечення, його інтерфейс, можливості та особливості будови його вікон та побудови його на рівні програмного забезпечення, показуючи його через UML діаграми.

					КНТЕУ 121-07-08.БР	Аркуш
						39
Зм.	Аркуш	№ докум	Підпис	Дата		

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В даній роботі були розглянуті різні рішення, щодо виконання поставленої автору задачі, було відображено інтерфейс створеної програми.

Була розглянута конкурентно спроможність створеної програми, її переваги та недоліки, в порівнянні з програмами – конкурентами.

Автор рекомендує всім, що хоче створювати чи буде створювати програмне забезпечення для поліграфії зрозуміти, хто саме буде кінцевий користувач, і наскільки він розуміє те, що відбувається на самому виробництві.

Тому що в різних користувачів – і цілі різні

Бухгалтеру – потрібно багато даних, самих різних типів, від відсотка паперу, що буде йти як «залишки» на виробництві, до всієї суми виробництва, що буде розписана поетапно.

В той час як звичайному користувачеві не потрібна настільки велика інформація, і він може обмежитись лиш сумою використаного матеріалу при виробництві, його ціною за ціною за весь процес виробництва, щоб прикинути, якщо це спонсор наприклад, наскільки це буде вигідно або невигідно відповідно.

					КНТЕУ 121-07-08.БР			
					Розробка системи обробки інформації в поліграфічному підприємстві	Стадія	Аркуш	Арку-
						ВП	40	41
Зм.	Аркуш	№ докум.	Підпис	Дата				
Зав. каф.		Криворучко О.В.						
Керівник		Жирова Т. О						
Гарант		Цензура М.О.			Висновки та пропозиції	Факультет інформацій-них технологій 4 курс, 7 група		
Розроб		Коблай А.С.						

СПИСОК ВИКОРИСТАННИХ ДЖЕРЕЛ

1. Проект «C++ Builder - Wikipedia» [Електронний ресурс]. – Режим доступу : https://ru.wikipedia.org/wiki/C%2B%2B_Builder
2. Проект «Особливості роботи в середовищі C++ Builder» [Електронний ресурс]. – Режим доступу : <http://cpp.dp.ua/osoblyvosti-roboty-v-seredovyshhi-c-builder/>
3. Проект «Microsoft SQL Server - Wikipedia» [Електронний ресурс]. – Режим доступу : https://ru.wikipedia.org/wiki/Microsoft_SQL_Server
4. Проект «Diagrams.net» [Електронний ресурс]. – Режим доступу : <https://app.diagrams.net>
5. Проект «Проектування програмного забезпечення – habr.com» [Електронний ресурс]. – Режим доступу : <https://habr.com/ru/post/74330/>
6. Проект «Типографія – «Контент пресс»» [Електронний ресурс]. – Режим доступу : <http://c-press.ru/>
7. Проект «Печатнік.com – онлайн форум» [Електронний ресурс]. – Режим доступу : <https://pechatnick.com/forum/viewtopic.php?t=11664>
8. Проект «OSzone.net – інформаційний портал, «Поліграфія 2.6»» [Електронний ресурс]. – Режим доступу : <http://soft.oszone.net/program/733/Poligrafiu/>
9. Проект «Архів поліграфічного форуму» [Електронний ресурс]. – Режим доступу : <https://forum.print-forum.ru/archive/index.php/t-524527.html>

					КНТЕУ 121-07-08.БР			
					Розробка системи обробки інформації в поліграфічному підприємстві	Стадія	Аркуш	Арку-
Зм.	Аркуш	№ докум.	Підпис	Дата		ВП	41	41
Зав. каф.	Криворучко О.В.							
Керівник	Жирова Т. О							
Гарант	Цензура М.О.							
Розроб	Коблай А.С.				Висновки та пропозиції	Факультет інформаційних технологій 4 курс, 7 група		

ДОДАТКИ

Додаток А

Розрахунок додаткових компонентів

```
a=t1*t2*r;  
ves1list=int(a*1000+0.5)/1000.0;  
str=StrToFloat(Edit7->Text);  
spusk=str/z; double test=spusk;  
if (float(spusk)==spusk)  
{v=spusk*t*ves1list;}  
else  
{  
int(spusk);  
v=(spusk*t*ves1list)+((t/2)*ves1list);  
}  
if(ComboBox4->ItemIndex==0) procent=procent - 0.3;  
lists=n*spusk;  
Edit8->Text=lists;  
Edit8->Text=Edit8->Text + " листів"; //листы на тех  
  
pechat = (2*procent)+3.8;  
Edit9->Text=pechat;  
Edit9->Text=Edit9->Text+ " %";  
dopProcent=0;  
if(CheckListBox1->Checked[0]==true) {dopProcent=dopProcent+1;}  
if(CheckListBox1->Checked[1]==true) {dopProcent=dopProcent+0.5;}  
if(CheckListBox1->Checked[2]==true) {dopProcent=dopProcent+0.2;}  
if(CheckListBox1->Checked[3]==true) {dopProcent=dopProcent+0.2;}  
if(CheckListBox1->Checked[4]==true) {dopProcent=dopProcent+0.5;}  
P=pechat+dopProcent;
```