

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему: Дослідження методів та алгоритмів для розробки ІС оптимізації
планування випуску продукції компанії «ЯрмарОК смаколиків»

Виконала: студентка групи 6144 м

_____ *Хасанова К.О.*
(підпис)

Керівник роботи:

доцент кафедри ІУСТ, к.т.н.,
(посада, науковий ступінь вчене звання)

_____ *Приходько Н.В.*
(підпис)

м. Миколаїв – 2021 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 126 "Інформаційні системи та технології"
Освітня програма "Інформаційні системи та технології"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Чернова Лб.С.

" ____ " _____ 2021 р.

*ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ*
на здобуття ступеня вищої освіти "магістр"

Студентці Хасановій Каріні Олексіївні

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження методів та алгоритмів для розробки ІС оптимізації
планування випуску продукції компанії «ЯрмарОК смаколиків»

Керівник роботи к.т.н., доцент кафедри ІУСТ Приходько Н.В.

Затверджені наказом ректора № 1232-уч. від "12" жовтня 2021 року.

2. Термін подання роботи: 21 грудня 2021 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні
джерела, технічна документація на існуючі аналоги систем, матеріали наукового
стажування

4. Перелік питань, що належать до розробки (найменування розділів) Аналіз
методів та алгоритмів оптимізації планування випуску продукції. Розробка
проекту інформаційної системи. Розробка робочої документації. Розділ з
економіки. Розділ з охорони праці. Розділ з охорони навколишнього середовища.
Висновки. Список використаної літератури. Додатки

5. Перелік презентаційних матеріалів Мета, об'єкт та предмет дослідження,
наукова новизна. Задачі. Категорії планування виробництва. Методології
управління виробництва. Схематичне представлення задачі дослідження.
Архітектура програмного продукту. Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	к.т.н., доц. Приходько Н.В.	15.09.20	26.12.20
2	к.т.н., доц. Приходько Н.В.	07.02.20	20.06.21
3	к.т.н., доц. Приходько Н.В.	03.09.21	27.11.21
4		06.09.21	10.12.21
5	ст. викл. Гурець Н.В.	06.09.21	10.12.21
6	ст. викл. Гурець Н.В.	06.09.21	10.12.21

7. Дата видачі завдання 04 вересня 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вибір теми, визначення мети, завдань, об'єкта та предмета дослідження	жовтень 2020 р.	
2	Підбір та опрацювання літератури	Грудень 2020 р.	
3	Аналіз стану проблеми планування роботи підприємства	Лютий 2021 р.	
4	Аналіз методів розв'язання задач теорії розкладів	Травень 2021 р.	
5	Розробка та реалізація проекту	Вересень 2021 р.	
6	Розробка додаткових розділів, оформлення роботи, підготовка презентаційних матеріалів	Жовтень 2021 р.	
7	Представлення до попереднього захисту	1 грудня 2021 р.	
8	Отримання відгуку керівника	13 грудня 2021 р.	
9	Отримання рецензії	15 грудня 2021 р.	
10	Захист кваліфікаційної роботи	21 грудня 2021 р.	

Студент

(підпис)

Хасанова К.О.

(ПІБ)

Керівник роботи

(підпис)

Приходько Н.В.

(ПІБ)

АНОТАЦІЯ

Тема магістерської роботи «Дослідження методів та алгоритмів для розробки ІС оптимізації планування випуску продукції компанії «ЯрмарОК смаколиків»».

На продовольчому ринку України широко представлені різноманітні м'ясні продукти. Підвищеним попитом серед населення користуються вітчизняні ковбасні вироби. Саме їх воліють купувати 98% населення. Тому українські м'ясопереробні підприємства зацікавлені у розширенні асортименту випущених ковбас, підвищенні їхньої конкурентоспроможності та зниженні собівартості.

Метою дослідження є підвищення продуктивності праці та якості виробів за рахунок впровадження інформаційної системи для оптимізації планування випуску м'ясної продукції компанії. Для досягнення мети в роботі вирішено задачі аналізу процесу транспортування товару від постачальника до споживача; аналізу методів випуску продукції; розроблена математична модель; проведено системний аналіз методів оптимізації в якості інструменту для планування; досліджено існуючі системи.

Наукова новизна роботи полягає в адаптації методів системного аналізу оптимізації для планування випуску м'ясної продукції.

На етапі дослідження використовувались методи системного аналізу, дослідження операцій, прийняття рішень та інше. На етапі розробки застосовувалась методологія проектування UML.

Робота виконана українською мовою та викладена на __ сторінках, містить __ рисунків, __ таблиці та _ додатки.

SUMMARY

Theme of the master's thesis "Research of methods and algorithms for development of information system of optimization of planning of release of production of the company "YarmarOK Smakolykiv"".

Various meat products are widely represented on the Ukrainian food market. Domestic sausage products are in high demand among the population. 98% of the population prefer to buy them. That is why Ukrainian meat processing companies are interested in expanding the range of sausages produced, increasing their competitiveness and reducing their cost.

The purpose of the study is to improve productivity and product quality through the introduction of an information system to optimize the planning of the company's meat production. To achieve this goal, the tasks of analyzing the process of transportation of goods from the supplier to the consumer are solved; analysis of production methods; developed mathematical model; systematic analysis of optimization methods as a planning tool; existing systems are investigated.

The scientific novelty of the work is to adapt the methods of system analysis of optimization to the planning of meat production.

During the research phase, methods of system analysis, operations research, decision making, and more were used. The UML design methodology was used during the development phase.

The work is in Ukrainian and is presented on __ pages, contains __ drawings, __ tables and _ appendices.

ЗМІСТ

Вступ

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ ПЛАНУВАННЯ РОБОТИ ПІДПРИЄМСТВА

1.1 Обумовлення актуальності

1.2 Дослідження процесу виробництва ковбасних виробів

1.3 Аналіз процесу планування випуску продукції

1.4 Сучасні методології управління виробництвом

1.5 Поняття оперативного планування основним виробництвом

1.6 Системи управління виробництвом

2 АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧ ТЕОРІЇ РОЗКЛАДІВ

3 РОЗВ'ЯЗАННЯ ЗАДАЧІ ОПТИМІЗАЦІЇ ТА РОЗРОБКА ПРОЕКТУ ІС

3.1 Математична постановка задачі

3.2 Дослідження властивостей задачі

3.3 Опис програмної реалізації системи

3.4 Обґрунтування вибору технологій розробки

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

4.1 Вступ

4.2 Розрахунок витрат на створення та експлуатацію інформаційної системи

4.3 Економічна ефективність розробки і впровадження системи

5 ОХОРОНА ПРАЦІ

5.1 Вступ

5.2 Аналіз шкідливих та небезпечних факторів

5.3 Розрахунок системи штучного освітлення приміщення відділу інформаційних технологій

5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів при роботі з персональним комп'ютером

5.4.1 Заходи щодо зменшення впливу

5.4.2 Техніка безпеки при роботі на комп'ютері

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ

6.2 Забруднення навколишнього середовища

6.3 Розробка заходів щодо зменшення забруднення навколишнього середовища

Висновки

Список використаної літератури

ДОДАТКИ

ВСТУП

На продовольчому ринку України широко представлені різноманітні м'ясні продукти. Підвищеним попитом серед населення користуються вітчизняні ковбасні вироби. Саме їх воліють купувати 98% населення. Тому українські м'ясопереробні підприємства зацікавлені у розширенні асортименту випущених ковбас, підвищенні їхньої конкурентоспроможності та зниженні собівартості.

Магістерська робота присвячена оптимізації шляхів транспортування товару від постачальника до споживача. Метою даної магістерської роботи є дослідження методів та алгоритмів для оптимізації планування випуску м'ясної продукції.

Метою дослідження є підвищення продуктивності праці та якості виробів за рахунок впровадження інформаційної системи для оптимізації планування випуску м'ясної продукції компанії.

Для досягнення вказаної мети були поставлені наступні задачі дослідження:

1. Проведення аналізу процесу транспортування товару від постачальника до споживача;
2. Проведення аналізу методів випуску продукції;
3. Розробка математичної моделі;
4. Проведення системного аналізу методів оптимізації в якості інструменту для планування;
5. Дослідження існуючих систем

Об'єктом дослідження є оптимізація шляхів транспортування товару від постачальника до споживача.

Предметом дослідження є методи та алгоритми оптимізації в якості інструменту для планування випуску м'ясної продукції.

Наукова новизна роботи полягає в адаптації методів системного аналізу оптимізації для планування випуску м'ясної продукції.

Завданням є системний аналіз та розробка інформаційної системи для оптимізації планування випуску м'ясної продукції компанії «ЯрмарОК смаколиків».

Актуальність залежить від того факту, що останнім часом помітне певне зростання вітчизняної харчової промисловості, який торкнувся і м'ясопереробної галузі.

У першому розділі описується теоретичний огляд.

У другому розділі описується методи прийняття рішень.

У третьому розділі проблеми вибору та інформаційна система.

Четвертий розділ присвячений розрахунку економічної ефективності системи підтримки прийняття рішень.

В п'ятому проводиться аналіз шкідливих та небезпечних факторів, розрахунок системи штучного освітлення та розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів.

В шостому розділі визначається зменшення забруднення навколишнього середовища.

1 АНАЛІЗ СТАНУ ПРОБЛЕМИ ПЛАНУВАННЯ РОБОТИ ПІДПРИЄМСТВА

1.1 Обумовлення актуальності

На сьогоднішній день однією з основних складових нормальної роботи підприємства та її конкурентоспроможності серед невпинно зростаючої кількості підприємств та швидкої зміни технологічних процесів є оптимальне планування випуску продукції. За сучасних умов підприємство повинно швидко реагувати на потреби ринку та бути адаптованим до різких змін у виробництві. Це все обумовлює необхідність впровадження на підприємствах автоматизованих систем управління виробництвом, що в свою чергу сприяє покращенню організації на виробництві, узгодженості роботи усіх його відділів.

У даній роботі розглядається задача оптимізації планування випуску продукції. Ця задача вирішується механізмами розділу дискретної математики – теорії розкладів, що займається проблемами планування.

1.2 Дослідження процесу виробництва ковбасних виробів

Виробництво ковбас складається з таких основних операцій:

- первинної обробки сировини,
- підготовки фаршу,
- формовки,
- теплової обробки.

Первинна обробка сировини включає розрубку туш, обвалу, жирівку.

Основою найбільш раціонального використання сировини є розбирання яловичих і свинячих туш з напрямком кращих частин у реалізацію як упакованого безкісткового м'яса напівфабрикатів, і навіть виробництва делікатесних виробів. У ковбасному виробництві доцільно використовувати

менш цінні частини туш, м'ясну і жирову обрізь, отримувану при розбиранні, і субпродукти.

Підготовка сировини до виробництва м'ясопродуктів

Основна сировина ковбасного виробництва – яловичина і свинина. Для спеціальних сортів ковбас використовують конину, м'ясо птахи, кроликів, харчову кров, і навіть субпродукти.

Обвалу обрубів виробляють після розбирання туш, полу туш чи четвертин на відповідні частини. Обвалка полягає у відділенні м'якуші від кісток. Вона буває ручний та механізованої.

Одним з шляхів раціонального використання сировини є комбінована обвалка яловичини, свинини, баранини із частково-обваленого сировини для супових наборів і рагу. Комбінованій обвалці піддають найбільш трудомісткі частини туші: спинно–реберну, шийну, поперекову і грудну.

Досягнення необхідних технологічних властивостей готового продукту (смаку, пахоців, кольору, консистенції) і запобігання їхню відмінність від мікробіологічної псування здійснює посол м'яса. Посол є обов'язковою і визначальною операцією в технологіях ковбасних і солоних продуктів. При значної спільності технологій кожна з яких має свої особливості й того. Процеси, характерні для посла, можуть продовжувати свій розвиток і по закінченні періоду власне посла. Так, для сирокочених ковбас більшість їх триває в своєрідних умовах при приготуванні фаршу, осаді, копченні, сушінню.

Приготування фаршу включає у собі вторинне здрібнення м'яса накуттре, складання фаршу з рецептури всіх складових частин фаршу. Через війнукуттеровання значно поліпшується структура і консистенція фаршу, підвищується його в'язкість і липкість.

Готовий ковбасний фарш надходить на шприцевання – наповнення фаршем ковбасних оболонок. До кожного виду ковбас підбирають оболонку відповідно до діючої нормативно-технічної документацією (вид оболонки, діаметр, довжина). Шприцевання ковбасним фаршем на шприцах різних конструкцій. Сутність роботи шприца – витіснення фаршу під тиском в оболонку

з єдиною метою надання йому необхідного споживчого виду. Щільність шприцевання залежить від виду ковбас, зміст вологи в фарші, виду оболонки, та способу термообробки ковбаси. Наприклад, для варених ковбас фарш наповнюють в оболонку не щільно, бо за варінні обсяг їх збільшується.

Наповнені батони надходять на ділянку в'язки шпагатом. Пов'язані ковбасні батони направляють на теплову обробку. І тому їх навішують на палиці за петлі, що роблять при в'язанню.

Кінцевим етапом у виробництві ковбасних виробів є термічна обробка. Воно складається з наступних процесів: опади, об жаркі, варіння, копчення, охолодження, сушіння.

Осадка є першою операцією завершального етапу технологічного процесу – термічної обробки ковбасних виробів, під час яких ковбаси і копченості доводяться до кулінарній готовності. Осадка відбувається у спеціальних камерах, де підтримується певний температурно-зволожувальний режим.

Після опади ковбаси направляють у камері для обсмаженні. Обсмаження – це короткочасна обробка поверхні ковбасних виробів коптільним димом при високих температур перед варкою.

Мета обсмаження – підвищення механічної міцності оболонки, та поверхні шару продукту, зменшення їх гігроскопічності. Процес обсмаження необхідно розглядати, як дво-фаза. Першу фазу – підсушка, друга – обсмаження.

Копчення. Під копченням розуміють поводження продуктів коптільними речовинами, одержуваними як коптільного диму внаслідок неповного згоряння дерева.

Організацію виробничого процесу здійснює одна людина - начальник цеху, у підпорядкуванні якого знаходяться практично всі інші працівники. Він також є технологом та здійснює контроль над технологічним процесом і керівництво всієї бригадою протягом зміни.

Склад бригади наступний:

- слюсар;
- електрик;

- мийник;

- склад робочих здійснюють технологічний процес.

До складу службовців підприємства, ще, входять бухгалтер і лаборант.

Склад виробничого корпусу та його основних підрозділів встановлюється відповідно до видами виконуваних робіт та його трудомісткістю. Склад допоміжних підрозділів приймається відповідно до потужністю підприємства, обраної технологічної схемою і прийнятої структурою управління.

З урахуванням наведених вище чинників ввімкнемо у складі виробничого корпусу такі основні приміщення:

- накопитель-дефростер;

- сировинне відділення;

- приміщення для попереднього подрібнення і посла;

- відділення для витримки м'яса в посолі;

- машинний зал;

- шприцевальна;

- приміщення для охолодження;

- приміщення для зберігання;

- відділення термообробки;

Допоміжні приміщення:

- склад для зберігання спецій;

- приміщення для зберігання інвентарю;

- слюсарна майстерня;

- лабораторія;

- роздягальня персоналу;

- кабінет цеху;

- санвузли.

1.3 Аналіз процесу планування випуску продукції

Успіх розвитку та функціонування кожної компанії залежить від багатьох складових, основною серед яких є ефективне планування її роботи.

Процес планування знаходить своє логічне завершення в плані. Планом називається офіційний документ, у якому відображаються прогнози розвитку підприємства і окремих сторін його діяльності у майбутньому; проміжні і кінцеві задачі та цілі, що стоять перед ним і його окремими підрозділами у відповідній сфері; механізми координації поточної діяльності і розподілу ресурсів; стратегії на випадок надзвичайних обставин.

Можна виділити три основні типи планів:

- плани-цілі;
- плани для повторювальних дій;
- плани для неповторювальних дій.

Відповідно виконання сформованого плану призведе до підвищення продуктивності компанії та зменшення її витрат.

При плануванні організація повинна встановити:

- 1) цілі у сфері якості та вимоги до продукції;
- 2) потреби в розробленні процесів, документів, а також у забезпеченні ресурсами для виробництва конкретної продукції;
- 3) необхідні перевірку, затвердження, моніторинг, контролю;
- 4) протоколи, необхідні для надання доказів того, що процес випуску і кінцева продукція задовольняють вимоги до управління записами.

Основна мета перелічених вимог полягає в плануванні та впровадженні процесів, спрямованих на усунення коливань рівня якості, що є основоположною завданням систем менеджменту якості.

Всі зазначені вище процеси здійснюються в повному обсязі в тому випадку, коли йде підготовка до випуску нової продукції.

У харчовій промисловості в результаті планування якості необхідно визначити типи контрольних параметрів, необхідних для даного процесу і даної продукції. Міжнародно визнаною системою для визначення необхідних

контрольних параметрів для перевірки безпеки харчових продуктів є так звана система НАССР.

Зазвичай процес планування якості фіксується у формі спеціальних "планів за якістю", розроблених для продукції і процесів конкретної організації. Форма таких планів може бути найрізноманітнішою: "плани управління виробничими процесами", "бізнес-плани", "програми безпеки харчових продуктів" і т.д.

Якщо роздрібна торговельна мережа планує власне харчове виробництво (фабрика-кухня, комбінат харчування, кондитерський цех, пекарня і т.п.), то, на думку А. Фоміна, компанія-оператор повинна вжити заходів з ефективного управління харчовими виробництвами (ХВ):

- 1) розробка та впровадження концепції харчового виробництва (насамперед прийняття рішення про те, що виробляти в єдиному центрі, що в магазині, визначення цільової аудиторії мережі);

- 2) визначення конфігурації напрямків виробництва (передусім визначення спеціалізації ХВ);

- 3) розбір проекту ХВ (насамперед вибір обладнання);

- 4) підбір запускає команди;

- 5) вибір каналів збуту;

- 6) точне дотримання графіку реалізації проекту ХВ.

Процеси, пов'язані зі споживачами

Як зазначалося вище, управління якістю починається з вивчення вимог споживачів. У стандарті встановлені три групи питань:

- 1) визначення вимог щодо продукції;

- 2) аналіз вимог щодо продукції;

- 3) вимоги до організації зв'язку зі споживачами.

Для виконання першої групи вимог організація повинна визначити:

- 1) вимоги, установлені замовником, зокрема вимоги до постачання і подальшим обслуговуванням;

- 2) вимоги, не установлені замовником, але необхідні для встановленого чи передбаченого використання, якщо про таке відомо;

- 3) законодавчі та інші обов'язкові вимоги, що відносяться до продукції;

4) будь-які додаткові вимоги, визначені організацією.

Додаткові вимоги можуть бути пов'язані з урахуванням вимог споживачів на зовнішньому і внутрішньому ринках. У тому випадку, коли продукція може стати предметом експорту, слід знати і виконувати законодавчі вимоги конкретних країн.

По ряду товарів, наприклад харчовим продуктам, треба визначати найбільш уразливі групи споживачів і приділяти їм підвищену увагу (до таких груп відносяться, наприклад, немовлята і діти, вагітні жінки, літні люди, особи, які страждають діабетом або іншими тяжкими захворюваннями).

Організація повинна аналізувати вимоги щодо продукції. Цей аналіз повинен проводитися до прийняття організацією зобов'язання щодо постачання продукції замовнику (наприклад, участь у тендерах, прийняття контрактів або замовлень, прийняття змін до контрактів чи замовлень).

Організація повинна визначати і здійснювати ефективні заходи щодо зв'язку із замовниками стосовно:

- 1) інформації про продукцію;
- 2) опрацювання запитів, контрактів чи замовлень та поправки;
- 3) зворотного зв'язку від споживачів, включаючи скарги споживачів.

Для вдосконалення менеджменту діяльності оптових торгових організацій дуже важливо враховувати побажання такої категорії споживачів, як підприємства роздрібної торгівлі.

При побудові розкладу для виробництва на підприємстві виникає проблема планування. Проблемами планування займається розділ дискретної математики – теорія розкладів.

Оскільки, в оптимальному розкладі (плані) на кожному пристрої може бути свій порядок обробки, то з ростом розмірності задачі величина варіантів розташування робіт зростає настільки швидко, що отримати рішення навіть з використанням найбільш потужних електронних обчислювальних машин (ЕОМ) можна лише для задач невеликих розмірів. Ця обставина викликала багато спроб

застосувати будь-який інший метод вирішення задачі, причому природною оцінкою будь-якого методу служить трудомісткість рішення.

Існують різні варіанти задач теорії розкладів, частина з них є NP-повними, частина належить до класу поліноміальних задач, для частини завдань так і не вдалося довести приналежність до певного класу складності. Існує гіпотеза, що задача, яка допускає переривання, не складніша за задачі без переривань. Для більшості задач гіпотеза залишається правдивою, крім однієї, де для варіанту без переривання доведена його приналежність до класу поліноміальних задач, в той час як для аналогічної задачі з перериваннями не існує доказів приналежності до якого-небудь класу складності.

Дана задача складання розкладу виконання робіт з відношенням передування паралельними пристроями з метою мінімізації загального часу виконання робіт є NP-повною задачею без переривань.

Планування виробництва можна розділити на наступні категорії:

- визначення черговості виконання завдань на одному пристрої (single-machine-problem або one-machine sequencing problem);
- визначення черговості виконання завдань на паралельних пристроях (single machine scheduling problem with parallel machines);
- визначення черговості виконання множини завдань для групи розміщених один за одним пристроїв, тобто конвеєр (flow-shop- scheduling);
- визначення черговості виконання множини завдань для групи розміщених один за одним пристроїв, коли кожне завдання, яке повинно на них виконуватися, складається із множини операцій та виконання кожної із цих операцій кожного із завдань можливе тільки на одному строго фіксованому пристрої (job shop scheduling);
- визначення черговості виконання множини завдань, де кожне завдання складається з множини операцій і на цій множині немає умови передування (open shop scheduling).

1.4 Сучасні методології управління виробництвом

Ефективність управління сучасним підприємством та його результат значною мірою залежить від того, які методології (методи) управління виробництвом застосовуються.

За допомогою методології управління керуюча система регулює, об'єднує та координує поведінку та дії всіх підпорядкованих суб'єктів, також забезпечує вибір адекватних методів управління тощо.

Принципи управління – це правила, норми управлінської діяльності, відповідно до яких створюється, функціонує та розвивається система менеджменту організації. Принципи управління є узагальненням практичного позитивного управлінського досвіду і ґрунтуються на певних законах і закономірностях розвитку методології управління виробництвом.

Наведемо огляд існуючих методологій управління виробництвом.

Домінуюча методологія управління виробництвом минулого століття має назву «Фордизм». Фордизм – це модель масового виробництва стандартизованих товарів на складальних конвеєрах з використанням низько кваліфікованих працівників, зайнятих простими операціями. Один з основних постулатів фордизму: «Виробляти великі партії виробів вигідніше, ніж дрібні», міцно укорінився серед управлінців XX століття

З другої половини минулого століття було багато спроб модифікувати фордизькі моделі. Усі рішення та відкриття з часом склалися в нову методологію, що отримала назву ощадне виробництво (Lean Manufacturing (LM)).

У кінці минулого століття з'явилася велика кількість різних методологій та парадигм з управління виробництва, серед яких докладніше розглянемо наступні: Швидкореагуюче виробництво (Quick Response Manufacturing (QRM)) та Активне виробництво (Agile Manufacturing (AM)). Сучасні, більш гнучкі в порівнянні з фордизмом методології, часто об'єднують терміном «постфордизм».

Ощадне виробництво (Lean Manufacturing, LM)

Метод застосовується для оптимізації виробництва і підвищення конкурентоспроможності. Метою методу є побудова виробництва, здатного швидко відповідати на вимоги споживачів і отримувати прибуток при будь-якій зміні ринку, у тому числі при падінні попиту. Створення досконалої виробничої системи, яка б при надходженні замовлення миттєво поставляла необхідну продукцію, і при цьому не відбувалося накопичення проміжних запасів.

Дана методологія спрямована на боротьбу з втратами у всіх їх проявах: зайві складські запаси, час простою, зайві переміщення, враховуючи при цьому зручність і безпеку виконання операцій для персоналу. Сьогодні ця система відома як виробнича система Toyota (Toyota Production System – TPS), принципи та інструменти якої знайшли відображення в її американському варіанті – системі ощадного виробництва (Lean Production). Багато з її складових, були ще й і в радянському варіанті – наукової організації праці (НОП).

Згодом, у рамках концепції ощадного виробництва було виділено безліч напрямків, кожен з яких представляє собою окремий метод: Потік одиничних виробів, Канбан, Всезагальне обслуговування обладнання (Total Productive Management (TPM)), Система 5S, Швидке переналагодження (SMED), Кайдзен, Захист від дурня, Управління персоналом, Точно в строк (Just In Time (JIT)).

Деякі методи самі претендують на статус самостійної виробничої методології або концепції, тому розглянемо деякі популярні з них, а саме концепцію «точно в строк».

Сенс роботи за концепцією «точно в строк» полягає в тому, щоб на всіх фазах виробничого циклу необхідний напівфабрикат до місця подальшої виробничої операції надходив саме тоді, коли це потрібно. Така концепція є в якійсь мірі «витягаючою», тобто такою, коли ділянки, розташовані на наступних етапах виробничого циклу, як би витягають необхідну їм продукцію з попередніх ділянок.

Головна мета виробничої системи «точно в строк» – забезпечити гнучку перебудову виробництва при зміні попиту. Така система забезпечує оперативне регулювання кількості виробленої продукції на кожній стадії виробництва.

Відомі різні модифікації системи «точно в строк», що застосовуються у світовій практиці. В їх основі лежить розробка, вперше застосована японською компанією «Тойота» і отримала широку популярність як система «Канбан».

Швидкореагуюче виробництво (Quick Response Manufacturing, QRM)

Швидке зростання великої кількості продукції, що пропонують виробники своїм замовникам, пов'язано зокрема з: розвитком CAD / CAM (системи автоматизованого проектування і виробництва), що дозволяє компаніям розробляти «під клієнта», а потім виробляти продукцію без несення високих додаткових витрат; розвитком Інтернет, який дозволяє замовнику без особливих

Отже, швидко реагуюче виробництво (QRM) – стратегія, що використовується компаніями для скорочення часу виконання замовлення та яка охоплює все підприємство. Мета QRM – скоротити час виконання замовлення за рахунок усіх операцій компанії, як внутрішніх, так і зовнішніх.

Якщо ощадне виробництво та інші схожі методології, засновані на зниженні витрат, спрямовані на скорочення реального часу роботи, то QRM орієнтований на зниження всього часу виконання замовлення.

Основні концепції QRM полягають у тому, що бізнес побудований на основі виконання роботи «для складу» (коли, щоб швидше виконати замовлення, основна номенклатура продуктів виготовляється заздалегідь та кладеться на склад), що через помилки планування та мінливості попиту призводить до збільшення КШВ, і в результаті це приводить до того, що компанія не може швидко реагувати на потреби клієнтів. Якщо сильно спростити – краще інвестувати у верстати і стандартно швидко реалізацію замовлень, ніж у склади. Тому був започаткований перехід від функціональних цехів до QRM-осередків. У філософії QRM-осередків можна простежити деяку аналогію з Scrum командою. Орієнтація працівників усіх підрозділів на єдину мету – зниження

часових витрат, а звідси і єдині параметри оцінювання роботи для її досягнення, згуртувати команду працівників.

Активне (гнучке) виробництво (Agile Manufacturing, AM)

AM – це стратегія управління компанією, мета якої зробити виробничу компанію більш стійкою до криз, змін попиту та інших непередбачуваних змін.

Для компаній, що працюють по AM, властива здатність швидко реконфігурувати трудові та матеріальні ресурси, щоб не втратити можливості заробити та уникнути неприємностей. Основною перевагою концепції AM є вміння оперативно підлаштовуватися під ситуацію, що змінюється і працювати в умовах невизначеності на ринку. AM підходить для галузей, де високий рівень невизначеності (наприклад IT, споживча електроніка та ін.).

Відбувається постійна готовність до змін і відповідь на них за допомогою сценарних стратегій. Сценарії: якнайбільше інтелектуальних ресурсів і якомога менше матеріальних; постійна група кросфункціональних фахівців; група фахівців на проектах на договірній основі, а також винесення непрофільних робіт на аутсорс; штат працівників не роздувається, а люди становлять ядро компанії та зацікавлені у справі.

Таким чином, особливу увагу, в компаніях типу AM приділяється мінімізації втрат від можливих, несподіваних негативних змін, таких як втрата контрактів або ринку збуту вироблених виробів. Одночасно, дисциплінарна, швидко розширювана команда й розгалужена партнерська мережа створюють передумови для того, щоб швидко реагувати на несподівані зміни.

Проведемо порівняння наведених методологій. Три розглянуті методології відрізняються в першу чергу стратегічною орієнтацією. LM націлене на створення більшого за допомогою мінімальних засобів. Іншими словами, LM постійно виявляє втрати будь-якого плану та їх ліквідує. QRM націлене на єдину мету – зменшення часу циклу виробництва від отримання заявки і до здачі продукту замовнику. У AM головною метою є вдосконалення можливостей для роботи в умовах невизначеності та мінливості ринку.

1.5 Поняття оперативного планування основним виробництвом

Планування роботи та управління виробництвом (рисунок 1.1) підрозділяється на техніко-економічне та оперативне планування. Оперативне планування підрозділяється на оперативно-виробниче планування та оперативне планування допоміжного виробництва. До складу оперативно-виробничого планування входить оперативно-календарне планування, диспетчеризація.

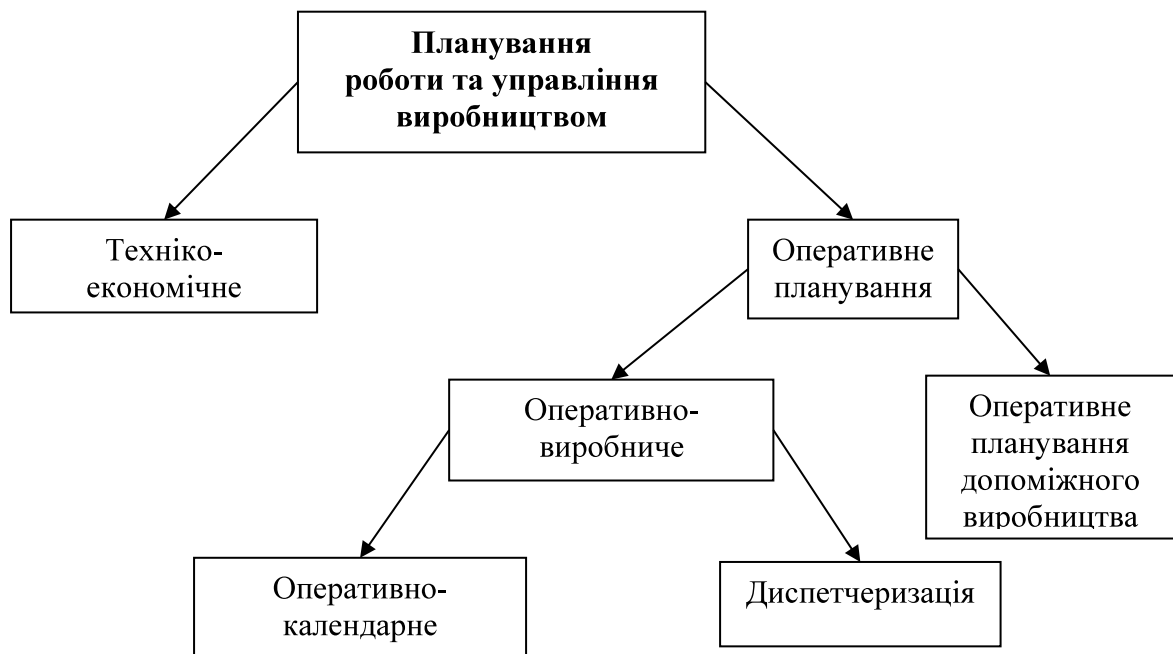


Рисунок 1.1 – Види планування роботи та управління виробництвом

Завданням оперативного планування є організація взаємоузгодженої роботи всіх виробничих підрозділів підприємства для забезпечення своєчасного виконання планового завдання при економному використанні ресурсів та високій якості продукції, тобто досягнення найкращих кінцевих результатів виробництва.

Змістом оперативного управління основним виробництвом є: встановлення місця та часу виготовлення виробів, деталей; облік фактичного ходу виробничого процесу; визначення відхилень від заздалегідь встановленого плану; регулювання ходу виробництва.

Система оперативного управління основним виробництвом (ОУ ОВ) являє собою складну організаційно-планову систему, що включає: функціональну, елементну та організаційну підсистеми.

Наукова обґрунтованість системи ОУ ОВ припускає визначення: вибору елементів системи; вибору та розрахунку календарно-планових нормативів; побудови об'ємних і оперативно-календарних планів; системи контролю та регулювання виробництва; достовірність вихідних даних.

Для оперативного планування виробництва застосовують системи планування і контролю виробництва, що сприяють організації рівномірної, ритмічної взаємоузгодженої роботи всіх виробничих підрозділів підприємства.

Система планування і контролю виробництва складається з п'яти основних рівнів:

- 1) розробка стратегічного бізнес-плану,
- 2) формування плану виробництва,
- 3) створення головного календарного плану виробництва,
- 4) розробка плану потреби в ресурсах,
- 5) формування переліку закупівель, здійснення планів та контроль над виробничою діяльністю.

Метою ОУ ОВ є забезпечення суворого виконання заданого плану випуску продукції за кількістю та номенклатурою і у встановлені строки на основі оптимального використання виробничих ресурсів, а також шляхом виявлення та мобілізації внутрішньовиробничих резервів.

Для реалізації цієї мети необхідна науково-обґрунтована побудована та функціональна система ОУ ОВ, яка має забезпечити вирішення наступних завдань:

- повне, комплектне і рівномірне виконання виробничої програми при дотриманні директивних строків випуску продукції;
- повне та найбільш оптимальне використання засобів виробництва і трудових ресурсів; максимальне прискорення виробництва та забезпечення максимального використання оборотних коштів у стадії виробництва;

- забезпечення умов, що сприяють розвитку передових форм організації праці у виробництві;
- автоматизація виконання основних планово-облікових, облікових робіт та отримання документації.

До функцій системи ОУ ОВ відносяться: управління як процес прийняття рішень; планування як процес визначення лінії поведінки об'єкта управління; облік як процес контролю, аналізу та виявлення відхилень від заданої планом лінії поведінки об'єкта; регулювання як процес локалізації виникаючих відхилень і збереження заданої лінії поведінки керованого об'єкта.

Система ОУ ОВ включає: об'ємне планування; оперативно-календарне планування; оперативний облік; поточний контроль і регулювання.

У процесі об'ємного планування проводиться розподіл річної виробничої програми виробництва в об'ємному і натуральному вираженні між цехами та дільницями відповідно до виділених їм трудових та матеріальних ресурсів. Для вирішення завдань об'ємного планування використовуються методи математичного програмування.

Оперативно-календарне планування – продовження та розвиток об'ємного планування. На цьому етапі об'єктом планування є окремі вироби, складальні одиниці, деталі, деталеоперації. Воно ґрунтується на певних нормативах (розміри партій деталей, складальних одиниць та виробів; тривалість виробничих циклів, випередження запуску і випуску партій деталей; нормативи незавершеного виробництва), що дозволяють здійснити зв'язок календарних планів та узгодження роботи взаємопов'язаних робочих місць, дільниць, цехів.

Оперативно-календарне планування здійснюється як на заводському, так і на внутрішньоцеховому рівнях. У завдання заводського оперативно-календарного планування входить визначення кількості та часу передачі по планованих позиціях (деталей, виробів) з цеху в цех по місяцях або кварталах.

До завдань внутрішньоцехового оперативного обліку відносяться: визначення виконання плану випуску деталей і складальних одиниць, а також плану виготовлення деталеоперацій і складальних одиниць дільницями; облік

наявності та надходження матеріалу, незавершеного виробництва на дільниці, цеха; облік браку та використання обладнання на дільницях; визначення виконання змінно-добових завдань дільницями і т.п.

Поточний контроль та регулювання виробництва відносяться до виробничої диспетчеризації, яка є складовою частиною оперативного управління. Вона включає: безперервний облік фактичного ходу робіт з виконання встановленого плану виробництва; прийняття оперативних заходів щодо попередження та усунення відхилень від плану; виявлення та аналіз причин відхилень від встановлених календарних графіків виробництва; координацію поточної роботи; організаційне управління оперативної підготовки всього необхідного для виконання календарних графіків виробництва.

Ефективність виконання завдань з організації та управління виробництвом багато в чому визначається забезпеченням його технічними засобами, які повинні забезпечувати збір та обробку інформації з оперативного обліку, контролю та регулювання ходу виробництва і передачу її за допомогою засобів зв'язку у відповідні підрозділи підприємства.

1.6 Системи управління виробництвом

Вже давно більшість компаній шукають способи спростити управління виробничими процесами. Для цього використовують різні системи управління виробництвом.

Першим кроком в розробці таких систем стало створення концепції MRP (Materials Resource Planning), яка підтримувала планування матеріальних ресурсів для виробництва. Основна мета концепції MRP полягала в мінімізації витрат, пов'язаних зі складськими запасами (в тому числі і на різних дільницях виробництва). В основі цієї концепції лежить поняття BOM (Bill Of Material – специфікація виробів), яка відображає залежність попиту на сировину, напівфабрикати та інші продукти від плану випуску готової продукції. При цьому дуже важливу роль відіграє час, для обліку якого необхідно мати чітке

уявлення про технологічний ланцюжок випуску продукції, тобто знати, яка послідовність і тривалість операцій. На підставі плану випуску продукції, ВОР і технологічного ланцюжка здійснюється розрахунок потреби в матеріалах до конкретних строків.

Однак у концепції MRP є серйозний недолік, а саме те, що при розрахунку в рамках цієї концепції не враховуються потреби в матеріалах, ні наявні виробничі потужності, ні їх завантаження, ні вартість робочої сили. Цей недолік був виправлений в концепції MRP II (Manufacturing Resource Planning – планування виробничих ресурсів). Поняття «MRP II» використовується для позначення відмінності «плану виробничих ресурсів» (MRP II) від «плану потреби в ресурсах» (MRP). MRP II дозволяла враховувати і планувати всі виробничі ресурси підприємства – сировину, матеріали, обладнання, персонал і т.д. MRP II забезпечує координацію маркетингу і виробництва.

Маркетинговий, фінансовий і виробничий відділ погоджують загальний, придатний для роботи план, виражений у виробничому плані.

Виробничий план встановлює загальні рівні виробництва і матеріально-виробничих запасів на відповідний горизонт планування строків. Першочергова мета полягає в тому, щоб визначити норми виробництва, які дозволять виконати поставлені в стратегічному бізнес-плані завдання.

Відділи маркетингу і виробництва повинні щотижня і щодня взаємодіяти з метою коригування плану з урахуванням змін (зміна розміру замовлення, скасування замовлення або затвердження відповідну дату поставки), що відбуваються. Зміни такого роду здійснюються в рамках генерального календарного плану виробництва.

Завдання, які підлягають виконанню при календарному плануванні виробництва, полягають у наступному: вибір ресурсів для виконання усіх поставлених завдань, агрегування та організація упорядкування різних видів ресурсів, детальне планування використання ресурсів (пристроїв) у часі, вирішення проблем упорядкування виконання завдань на кожній із одиниць

обладнання, побудова календарних планів роботи обладнання та виконання завдань, календарне планування роботи виробничого транспорту.

Оперативно-календарне планування складається з чотирьох функціональних етапів:

- 1) об'ємно-календарне планування (Master Production Schedule, MPS);
- 2) баланс виробничих потужностей (Capacity Planning Problem, CPP);
- 3) розрахунок виробничого розкладу (Production Scheduling, PS);
- 4) групування деталей і складальних одиниць та обладнання (Group Technology, GT).

Менеджери відділів маркетингу та виробництва можуть вносити зміни в генеральні календарні плани виробництва з урахуванням змін прогнозованого попиту. Управління підприємства може змінювати виробничий план відповідно до загальних змін попиту або положення з ресурсами. Проте, всі співробітники працюють в рамках системи MRP II. Вона служить механізмом координації роботи маркетингового, фінансового, виробничого та інших підрозділів компанії. MRP II являє собою метод ефективного планування всіх ресурсів виробничого підприємства.

По мірі розвитку концепції MRP II до неї поступово додавалися можливості обліку інших витрат підприємства. Так з'явилася концепція ERP (Enterprise Resource Planning – планування ресурсів підприємства). В основі ERP лежить принцип створення єдиного сховища даних, що містить всю ділову інформацію, накопичену організацією в процесі ведення бізнесу, зокрема фінансову інформацію, дані, пов'язані з виробництвом, управлінням персоналом, і будь-які інші дані, які стають одночасно доступними для всіх працівників, що володіють відповідними повноваженнями.

У дев'ятому виданні «Словника APICS» Американської Асоціації Контролю над Виробництвом та матеріально-виробничих запасів (APICS) дається таке визначення ERP: призначена для ведення звітності інформаційна система ідентифікації та планування підприємства – глобальних ресурсів,

необхідних для виробництва, транспортування і складання звітів за замовленнями клієнтів.

У роботі ERP-система має наступне визначення – це інтегрована система на базі інформаційних технологій для управління внутрішніми і зовнішніми ресурсами підприємства.

Концепція ERP знайшла широке застосування, оскільки планування ресурсів дозволяло скоротити час випуску продукції, знизити рівень товарно-матеріальних запасів, а також поліпшити зворотний зв'язок зі споживачем при одночасному скороченні адміністративного апарату. Стандарт ERP дозволив об'єднати всі ресурси підприємства і підвищити ефективність управління ними.

Для повноцінної експлуатації повинні бути передбачені програми для планування, календарного планування, калькуляції собівартості та інше на всіх рівнях організації, у робочих центрах, відділах, підрозділах та для всіх їх разом. Важливо відзначити, що ERP охоплює компанію цілком, а MRP II відноситься до виробництва. Проектування та застосування ERP-систем широко описані у роботах.

На теперішній час практично всі сучасні західні системи управління виробництвом базуються на концепції ERP і відповідають її рекомендаціям. Ці рекомендації виробляються американською громадською організацією APICS, що об'єднує виробників, консультантів у галузі управління виробництвом, а також розробників програмного забезпечення (ПЗ).

Одним із стандартів систем управління підприємствами є CSRP (Customer Synchronized Resource Planning), крім усього іншого, він охоплює і взаємодію з клієнтами: оформлення нарядів / замовлень і технічних завдань, підтримка замовника на місцях і т.п. Таким чином, якщо стандарти MRP, MRP II та ERP орієнтовані на внутрішню організацію підприємства, то стандарт CSRP включає в себе повний цикл – від проектування майбутнього виробу, з урахуванням вимог замовника, до гарантійного і сервісного обслуговування після продажу. Суть концепції CSRP головним чином полягає в тому, щоб інтегрувати замовника в систему управління підприємством. Відповідно до даної концепції не відділ

збуту, а безпосередньо сам покупець розміщує замовлення на виготовлення продукції, сам відповідає за правильність його виконання і при необхідності відстежує дотримання строків виробництва і поставки. При цьому, саме підприємство може дуже чітко відстежувати тенденції попиту на його продукцію.

На сьогодні існує безліч програмних продуктів, які дозволяють вирішувати ті чи інші завдання календарного планування. Наприклад, функціональні етапи календарного планування MPS та CPP виконуються, як правило, системами класу ERP (Enterprise Resource Planning), етапи PS та GT реалізуються засобами MES (Manufacturing Execution Systems). Хоча для CPP і PS етапів календарного планування виробництва також успішно використовують APS (Advanced Planning Systems).

На передових як вітчизняних так і зарубіжних підприємствах широко впроваджуються:

- системи управління ресурсами підприємства (ERP), що формують в автоматизованому режимі номенклатурні плани виробництва;
- програми для вдосконаленого планування (Advanced Planning & Scheduling, APS), що представляє собою концепцію оптимізованого (або синхронного) виробничого планування, головною особливістю якої є можливість швидкого складання планів з урахуванням наявних ресурсів і виробничих обмежень (переналагодження обладнання, доступність оснащення, зв'язок між пристроями та ін.) і швидкого перепланування за заздалегідь складеним сценарієм оптимізації, концепція APS поєднала в собі основні елементи концепцій MRP, MRP II, Finite Capacity Scheduling (FCS);
- виконавчі виробничі системи (Manufacturing Execution System, MES), що пов'язують воедино всі бізнес-процеси підприємства з виробничими процесами, які підвищують ефективність використання верстатної системи;
- системи управління трудовими ресурсами (Human Resource Management, HRM), їх ділять на «розрахункові», «облікові» та системи управління трудовими

ресурсами за рівнями автоматизації управління персоналом, які відповідають етапам розвитку прикладних програмних рішень для кадрових служб;

- системи для ефективного управління складом (Warehouse Management System, WMS), що підтримують усі складські бізнес-процеси, якісне планування операцій, розробку раціональної топології складу, управління ресурсами, номенклатурою, впровадження передових методів для обробки і зберігання запасів;

- системи управління ланцюгами поставок (Supply Chain Management, SCM), які призначені для автоматизації та управління всіма етапами постачання ресурсами підприємства і для контролю за рухом товару на підприємстві;

- корпоративна інформаційна система управління відносинами з клієнтами (Customer Relationship Management, CRM), що дає можливість не просто автоматизувати взаємодію з клієнтами і процес продажів, а вибудовувати їх роботу таким чином, щоб отримувати максимальний результат;

- система управління життєвим циклом продукції (Product Lifecycle Management, PLM), являє собою методологію застосування сучасних інформаційних технологій для підвищення конкурентоспроможності промислових підприємств.

Основними компонентами PLM-системи є:

- PDM-система (Product Data Management) – система управління даними про виріб, є основою PLM, призначена для зберігання і управління даними;

- CAD-система (Computer Aided Design) – проектування виробів;

- CAE-система (Computer Aided Engineering) – інженерні розрахунки;

- CAPP-система (Computer Aided Production Planning) – розробка прогресивних технологічних процесів;

- CAM-система (Computer Aided Manufacturing) – розробка керуючих програм для верстатів;

- MPM-система (Manufacturing Process Management) – моделювання та аналіз виробництва виробів;

– система скорочення витрат (Enterprise Asset Management, EAM) [33] орієнтована на скорочення витрат, пов'язаних із обслуговуванням устаткування і підвищенням продуктивності.

Інтеграція різних систем привела до розмивання меж між ними. Наслідком цього процесу стало нечітке розуміння, навіть у середовищі IT-фахівців, функціональної спеціалізації різних систем планування та управління виробництвом. Виробники ERP-систем стали заявляти, що їхні продукти тепер «легко вирішують» всі MES-завдання, а виробники MES-систем, у свою чергу, почали переконувати представників виробництва, що саме їх системи прийшли на зміну «застарілим» системам класу ERP. Продавці APS твердять про те, що, системи класу APS легко можуть замінити як ERP, так і MES разом узяті.

Розглянемо існуючі функціональні відмінності між декількома розглянутими системами планування виробництва.

Система ERP виконує об'ємно-календарне планування (MPS), попутно вирішуючи завдання балансу виробничих потужностей (CPP).

Система APS уточнює розраховані в ERP планові завдання, розподіляючи (оптимізуючи) їх по робочих місцях. На цьому етапі можливе внесення корекцій у вихідний MPS. Оскільки послідовність цих завдань, як правило, залежить від залученого в контур планування ланцюжка поставок матеріалів і комплектуючих (SCM), то порушувати цю послідовність всередині цеху вже не можна без ризику зруйнувати основний виробничий план.

Система MES за рахунок можливостей багатокритеріальної оптимізації дозволяє, варіюючи пріоритети різних завдань, вирішити задачу максимізації швидкості їх проходження через робочі місця. При цьому допускається не тільки перевпорядкування завдань, а й перерозподіл деяких з них на інші взаємозамінні робочі місця. Відзначимо, що на MES-рівні ніяких змін у загальний виробничий план вносити не можна.

Програмне забезпечення, розробка якого розглядається у даній роботі, дозволяє вирішити задачу складання розкладу виконання робіт з відношенням

передування паралельними пристроями за критерієм мінімізації загального часу виконання робіт.

2 АНАЛІЗ МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧ ТЕОРІЇ РОЗКЛАДІВ

На даний час існують різні методи розв'язку задач теорії розкладу. Проведемо огляд запропонованих методів розв'язання.

Велику кількість методів дискретної оптимізації можна розділити на дві групи: точні та наближені. На даний момент цей поділ є умовним, адже багато точних методів можуть застосовуватися як наближені, а наближені методи при певних умовах можуть застосовуватися як точні або їх складова частина. Найбільшого розповсюдження отримали наступні підходи та методи:

— точні методи:

1) методи цілочисельного програмування:

- а) лінійне цілочисельне програмування;
- б) нелінійне цілочисельне програмування;
- с) булеве програмування;

2) послідовні алгоритми оптимізації:

- а) метод гілок та меж;
- б) динамічне програмування та методи аналізу і відсіву варіантів;
- с) методи теорії графів;

— наближені методи:

1) обмеження об'єму розрахунків у послідовних алгоритмах оптимізації;

2) методи випадкового пошуку:

- а) методи глобального випадкового пошуку,
- б) методи локальної варіації;

3) генетичні алгоритми та еволюційні стратегії;

4) евристичні методи;

5) гібридні алгоритми;

6) метаевристичні методи;

7) імітаційне моделювання.

До наближених методів рішення відносяться методи глобального випадкового пошуку і локальних варіацій. Характерною особливістю цього методу є те, що на кожній ітерації процес вибору нового рішення виконується не детерміновано, а в результаті реалізації деякого випадкового процесу, при чому стратегія вибору може залежати від передісторії пошуку. Одним із розвитком методів локальної варіації є Tabu-Search-аналіз.

Генетичні алгоритми є евристичними алгоритмами пошуку, які використовуються для рішення задач оптимізації та моделювання послідовності підбору, комбінації та варіації шуканих параметрів на основі механізмів, які нагадують біологічну еволюцію. Оскільки, алгоритм у процесі пошуку використовує деяке кодування множини параметрів замість самих параметрів, то він може ефективно застосовуватися для рішення задач теорії розкладів, дискретної оптимізації, визначених як на числових множинах, так і на кінцевих множинах вільної природи.

Достатньо широке розповсюдження в наш час отримали різні евристичні методи рішення задач. Основні підходи при побудові цих методів базуються на прийомах зниження вимог та використанні різних видів вирішальних правил, обґрунтованих при рішенні близьких за постановкою задач, але які не містять цілої множини додаткових складностей та обмежень, характеристик для реально розглянутої проблеми. Евристичні алгоритми використовують різні підходи без суворих обґрунтувань. Евристичні методи дозволяють знаходити прийнятні рішення навіть у дуже складних випадках: при неповноті, випадковості вихідних даних, відсутності адекватної математичної моделі, NP-складності вирішуваної задачі та відсутності точних методів їх рішення.

Також часто для отримання наближеного рішення задачі використовується «лівостороння» схема розгалуження у методі гілок та меж із зупинкою алгоритму у випадку отримання першого допустимого рішення задачі.

Математичні методи вирішення задач теорії розкладів можна поділити на наступні види:

- математичні методи цілочисельного математичного програмування;

– математичні методи задач теорії розкладів у вигляді побудови допустимих екстремальних шляхів на графі.

Більшість задач теорії розкладів можуть бути сформульовані як задачі цілочисельного математичного програмування. Частковим випадком яких є задача лінійного цілочисельного програмування. Динамічне програмування зазвичай застосовується до вирішенню задач, які можна розбити на послідовність задач меншої розмірності та більш простої структури.

Джонсон запропонував простий алгоритм знаходження точного рішення для задачі обробки n деталей на двох пристроях і для окремого випадку завдання обробки на трьох пристроях, сформульованої Беллманом і званої в даний час завданням Беллмана-Джонсона.

Не зважаючи на те, що більшість задач теорії розкладів є NP-важкі, практика потребує рішення таких задач. Для цього існує декілька підходів. Першим підходом є розробка поліноміальних евристичних алгоритмів. Евристичний алгоритм, або просто евристика — це алгоритм, спроможний видати прийнятне рішення проблеми серед багатьох рішень, але неспроможний гарантувати, що це рішення буде найкращим. Отже, такі алгоритми є приблизними і неточними. Зазвичай такі алгоритми знаходять рішення, близьке до найкращого і роблять це швидко. Іноді такий алгоритм може бути точним, тобто він знаходить дійсно найкраще рішення, але він все одно буде називатися евристичним доти, доки не буде доведено, що рішення дійсно найкраще. Серед евристичних алгоритмів важливу роль відіграють так звані жадібні алгоритми (іноді це носить назву принципу жадності). Цей принцип полягає у тому, щоб на кожному кроці вибирати з усіх можливих альтернатив ту, що забезпечує найменше значення критерію. Цей принцип може бути застосований як при знаходженні початкового розкладу, так і на етапі його дальшого покращення (наприклад, згаданим вище шляхом локального пошуку).

Типовим представником евристичних алгоритмів є метод локального пошуку. При цьому заздалегідь вибрана множина перестановок використовується для послідовного покращення початкового розкладу доти,

доки покращення є можливим. Таким чином, знаходиться локальний оптимум у даному класі перестановок.

Іншим напрямком евристичних методів у теорії розкладів формування функцій переваг (пріоритетів). Для кожної i -ї роботи обчислюється значення функції $f(i)$ переваги та вибирається та робота, для якої $f(i)$ досягає мінімуму чи максимуму.

Також евристичним підходом є так звані генетичні алгоритми, суть яких полягає у наступному. Нехай є два допустимі розклади. По цих розкладах за деякими встановленими правилами будується декілька розкладів, кожен з яких є “схожим” у деякому розумінні на ці два вихідні розклади (вони називаються нащадками). З одержаних нащадків вибираються найкращі за заданим критерієм і перевіряється, чи є вони кращими за “батьків”. Якщо так, то по цих розкладах знову будуються нащадки і т.д.

Зустрічаються і такого типу задачі, в яких евристичний алгоритм буде працювати або дуже довго, або видавати невірні результати, однак, такі парадоксальні приклади можуть ніколи не зустрітись на практиці через свою специфічну структуру. Для деяких евристичних алгоритмів відомі оцінки похибки одержуваного рішення. Такі алгоритми називаються наближеними. Існують наближені алгоритми, що гарантують як відносну похибку так і абсолютну похибку. Деякі NP-важкі задачі допускають існування так званої апроксимаційної схеми. В рамках даної схеми можна знайти наближене рішення з відносною похибкою не більше будь-якого заданого значення $\varepsilon > 0$ за час, поліноміально залежний від $1/\varepsilon$ і від розміру вхідної інформації задачі, — цілком поліноміально апроксимаційна схема.

Для задач теорії розкладів такі схеми розробили Алон, Мастроліллі та інші. Для завдань, які не мають апроксимаційної схеми, велике значення має встановлення граничного значення ε , для якого можливо знаходження ε -наближеного рішення за поліноміальний час, — поліноміальна апроксимаційна схема.

На даний момент широкого поширення мають метаевристичні алгоритми, які знаходять "гарне" рішення, близьке до оптимального, за прийнятний час. Недоліком таких алгоритмів є відсутність оцінок якості отриманого рішення. Невідомо, наскільки рішення відрізняється від оптимального в найгіршому випадку. Метаевристика є потужним і надзвичайно популярним класом оптимізаційних методів. дозволяють знаходити рішення для широкого кола завдань з різними додатками. Сила метаевристичних полягає в їх здатності рішення складних завдань без знання простору пошуку, саме тому ці методи дають можливість вирішувати складні для задачі оптимізації. Спрощено можна назвати метаевристичні алгоритми як алгоритми, що реалізують прямий випадковий пошук можливих рішень задачі, оптимальних або близьких до оптимальних, поки не буде виконана якась умова або досягнуто задане число ітерацій.

Проблеми планування на паралельних пристроях можна класифікувати за такими типами:

- проблема календарного планування на ідентичних паралельних пристроях;
- проблема календарного планування на рівномірних/пропорційних паралельних пристроях;
- проблема календарного планування на не пов'язаних паралельних пристроях.

Нехай, t_{ij} – час обробки роботи j на пристрої i , для та $i = 1, 2, 3, \dots, m$
 $j = 1, 2, 3, \dots, n$.

Тоді задаються три способи планування паралельних пристроїв за допомогою цього часу обробки.

1) Якщо $t_{ij} = t_{1j}$ для всіх i та j ,

то проблема називається проблемою, ідентичних паралельних пристроїв. Це означає, що всі паралельні пристрої ідентичні з точки зору їх швидкості. Кожна робота займає однакову кількість часу обробки на кожному з паралельних пристроїв.

2) Якщо $t_{ij} = \frac{t_{1j}}{s_i}$ для всіх i та j , де s_i - швидкість пристрою i , а t_{1j} - час обробки роботи j на пристрої 1, то проблема називається з рівномірним (пропорційним) паралельним пристроєм.

Це означає, що паралельні пристрої матимуть різну швидкість. Як правило, ми припускаємо швидкість $1, 2, \dots, m$ для паралельних пристроїв $1, 2, \dots, m$ відповідно з співвідношенням $1, 2, \dots, m$ $s_1 < s_2 < \dots < s_m$. Тобто пристрій 1 є найповільнішим пристроєм, а пристрій m - найшвидшим пристроєм. Для заданої роботи час його обробки на паралельних пристроях буде в співвідношеннях, наведених нижче:

$$\frac{1}{s_1} : \frac{1}{s_2} : \dots : \frac{1}{s_m}.$$

3) Якщо t_{ij} є довільним для всіх i та j , то це проблема незв'язаних паралельних пристроїв. У цьому типі планування не буде ніякого відношення між періодами обробки роботи на паралельних пристроях. Це може бути пов'язано з технологічними відмінностями пристроїв, різними характеристиками робочих місць тощо.

Основні характеристики проблеми календарного планування на рівномірних/пропорційних паралельних пристроях описані нижче [4]:

- є n окремих робіт;
- є m паралельних пристроїв з різною швидкістю ($s_1 < s_2 < \dots < s_m$);
- пристрої постійно доступні, і вони ніколи не залишаються вільними, поки робота чекає;
- t_{1j} - час обробки роботи j на пристрої 1 для $j = 1, 2, 3, \dots, n$;
- для кожної роботи час обробки на однорідних паралельних пристроях обернено пропорційний швидкості цих паралельних пристроїв $\left(\frac{1}{s_1} : \frac{1}{s_2} : \dots : \frac{1}{s_m}\right)$,

де s_1 - одиниця швидкості;

$$- t_{ij} = \frac{t_{1j}}{s_i} \text{ for } j = 1, 2, 3, \dots, n \text{ and } i = 1, 2, 3, \dots, m;$$

У роботі розглядається задача складання розкладу виконання робіт з відношенням передування паралельними пристроями (як ідентичних, так і різної продуктивності), тому проаналізуємо сучасні досягнення наукової спільноти у вирішенні задач цього класу із застосуванням різних математичних методів.

3 РОЗВ'ЯЗАННЯ ЗАДАЧІ ОПТИМІЗАЦІЇ ТА РОЗРОБКА ПРОЕКТУ ІС

Дана задача визначення мінімального загального часу виконання робіт із заданим відношенням передуванням паралельними пристроями може розглядатися як для ідентичних пристроїв та і для пристроїв, що мають різну продуктивність (рисунок 3.1).

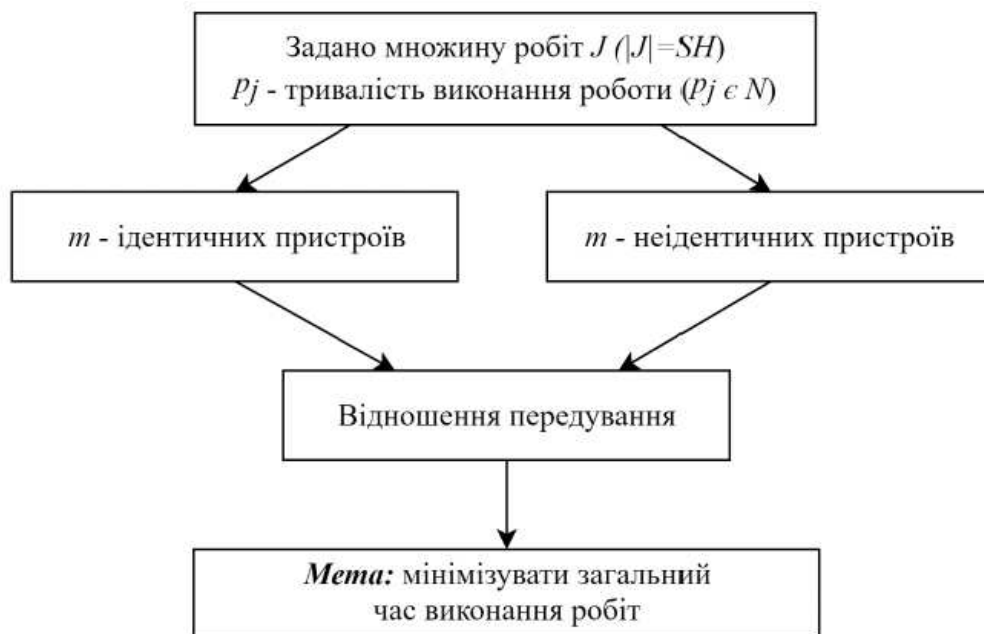


Рисунок 3.1 - Схематичне представлення задач дослідження

3.1 Математична постановка задачі

Задано множину робіт J ($J = S \sqcup H$), що розбиті на n ланцюгів, де $n = 2S$.

Ланцюг i складається з i n робіт, $i = 1, n$. Кількість пристроїв, що обробляють дані роботи – m . Кожна робота з J описується парою ij , де i – номер ланцюга, j – порядковий номер у ланцюзі, $1, ij = n$, і тривалістю виконання, $ij \in P \sqcup N$ (N – множина натуральних чисел). Усі пристрої починають свою роботу в один і той же момент часу (далі – момент запуску пристроїв). Пристрої працюють паралельно і є взаємозамінними у тому сенсі, що кожний з пристроїв може виконувати будь-яку роботу з множини J . Пристрої відрізняються один від одного продуктивністю виконання робіт і є одноманітними (пропорційними).

Тобто, пристрої можна впорядкувати за швидкістю виконання робіт і цей порядок однаковий для всіх робіт: для пристрою v заданий коефіцієнт $v k$ такий, що тривалість виконання роботи ij на пристрої v дорівнює $1, v ij k p v = t$. «Еталонним» будемо називати пристрій з коефіцієнтом продуктивності $k = 1$. У цьому сенсі величина $ij p$ є тривалістю виконання роботи ij на еталонному пристрої. Величину $v k$ будемо називати *коефіцієнтом продуктивності* (якщо $1 v k \square$, то пристрій v менш продуктивний, ніж еталонний, якщо $1 v k \square$ – більш продуктивний).

Передбачається, що всі роботи множини J надходять одночасно, процес обслуговування кожної роботи протікає без переривань. Усі пристрої працюють без переривань.

Необхідно побудувати такий розклад виконання робіт множини J , при якому досягає мінімуму загальний час виконання робіт:

$$C_{\max} = \max_{ij} C_{ij} = \max_i C_{in_i} \rightarrow \min, \text{ де}$$

C_{ij} – момент завершення роботи ij , C_{in_i} – момент завершення останньої роботи i – того ланцюга.

Далі, не втрачаючи загальності, будемо вважати, що еталонний пристрій є найбільш продуктивний (звідси випливає, що всі $1 v k \square$); всі $ij p$ є цілими числами.

Задача, що розглядається належить до класу NP.

3.2 Дослідження властивостей задачі

Визначимо теоретично мінімальний час, за який усі пристрої могли б виконати всі роботи в об'ємі

Цю величину можна отримати, якщо вважати, що усі пристрої можуть паралельно виконувати кожну з робіт. Для цього t паралельних пристроїв розглядаються як один, а фактичні тривалості робіт замінюються так званими узагальненими значеннями, рівними:

$$\tilde{p}_{ij} = \frac{1}{\sum_{v=1}^m \frac{1}{t_{vij}}},$$

де $t_{vij} = k_v p_{ij}$ – тривалість виконання роботи ij пристроєм v ($v = \overline{1, m}$; $j = \overline{1, n}$;

$j = \overline{1, n_i}$);

$\frac{1}{t_{vij}}$ – частка роботи ij , яка виконується пристроєм v за одну одиницю часу;

$\sum_{v=1}^m \frac{1}{t_{vij}}$ – частка роботи ij , яка виконуються всіма пристроями за одну одиницю

часу;

$\frac{1}{\sum_{v=1}^m \frac{1}{t_{vij}}}$ – тривалість виконання роботи ij всіма пристроями.

Тоді, з однієї сторони, мінімально можливий час, за який усі пристрої могли б виконати усі роботи, відповідає «рівномірному» розкладу, тобто розкладу, в якому кожен з пристроїв закінчує роботу в момент:

$$\sum_{i=1}^n \sum_{j=1}^{n_i} \frac{1}{\sum_{v=1}^m \frac{1}{k_v p_{ij}}}. \quad (3.1)$$

Однак, з урахуванням того, що в даній задачі роботи мають відношення передування у вигляді ланцюгів, то загальний час на виконання усіх робіт не може бути меншим, ніж час виконання робіт найдовшого з ланцюгів найпродуктивнішим пристроєм. Тобто нехай $P_i = \sum_{j=1}^{n_i} p_{ij}$, $i = \overline{1, n}$. Упорядкуємо ланцюги по незростанню їх сумарної тривалості робіт $P_1 \geq P_2 \geq \dots \geq P_n$, а пристрої за їх продуктивністю так, що $k_1 \leq k_2 \leq \dots \leq k_m$, $k = \overline{1, m}$. Тоді нижня оцінка критерію C^* з урахуванням (3.1):

$$C^* = \max \left\{ P_1 k_1; P_2 k_2; \dots; P_{m-1} k_{m-1}; \sum_{i=1}^n \sum_{j=1}^{n_i} \tilde{p}_{ij} \right\}$$

$$C^* = \max \left\{ P_1 k_1; P_2 k_2; \dots; P_{m-1} k_{m-1}; \sum_{i=1}^n \sum_{j=1}^{n_i} \frac{1}{\sum_{v=1}^m \frac{1}{k_v p_{ij}}} \right\} \quad (3.2)$$

Величина може бути як цілим так і не цілим числом. * С

Враховуючи (3.2), маємо два випадки, що є важливими для аналізу задачі:

- 1) коли максимум у (3.2) відповідає останньому елементу;
- 2) коли максимум у (3.2) відповідає одному з перших елементів.

Розглянемо деякий розклад. Позначимо в цьому розкладі:

$J_v(\sigma)$ – упорядкована множина робіт, що виконується пристроєм v , $v = \overline{1, m}$;

$C_v(\sigma) = \sum_{ij \in J_v(\sigma)} k_v p_{ij}$ – тривалість зайнятості пристрою v , $v = \overline{1, m}$;

$\Delta_v(\sigma) = \max\{0; C_v(\sigma) - C^*\} = \max\left\{0; \sum_{ij \in J_v(\sigma)} k_v p_{ij} - C^*\right\}$ – *виступ* пристрою v , $v = \overline{1, m}$;

$R_v(\sigma) = \max\{0; C^* - C_v(\sigma)\} = \max\left\{0; C^* - \sum_{ij \in J_v(\sigma)} k_v p_{ij}\right\}$ – *резерв* пристрою v , $v = \overline{1, m}$.

Позначимо через $C_v'(\sigma) = \frac{C_v(\sigma)}{k_v}$ *зведену тривалість зайнятості пристрою v*

($v = \overline{1, m}$) в розкладі σ – час зайнятості пристрою, розрахований у еталонних

тривалостях виконання робіт. Відповідні зведені величини:

$$\Delta_v'(\sigma) = \max\{0; C_v'(\sigma) - c_v^*\}; \quad v = \overline{1, m}; \quad R_v'(\sigma) = \max\{0; c_v^* - C_v'(\sigma)\}, \quad v = \overline{1, m}.$$

Визначимо:

$I_\Delta(\sigma)$ – множина пристроїв, для яких $\Delta_v(\sigma) > 0$;

$I_R(\sigma)$ – множина пристроїв, для яких $R_v(\sigma) > 0$;

$I_0(\sigma)$ – множина пристроїв, для яких $\Delta_v(\sigma) = R_v(\sigma) = 0$.

Із визначення величин $R_v(\sigma)$ і $\Delta_v(\sigma)$ слідує, що $R_v(\sigma)\Delta_v(\sigma) = 0$, $v = \overline{1, m}$.

З урахуванням позначень критерій оптимізації вихідної задачі (мінімізація загального часу виконання робіт) має вигляд:

$$C_{\max}(\sigma) = C^* + \max_{1 \leq v \leq m} \Delta_v(\sigma) \rightarrow \min$$

і є еквівалентним критерію мінімізації загального часу виконання всіх робіт:

$$C^* + \max_{1 \leq v \leq m} \Delta_v(\sigma) \rightarrow \min.$$

У свою чергу, з урахуванням того, що $C^* = \text{const}$, тобто не залежать від упорядкування, критерій спрощується і зводиться до мінімізації максимального виступу:

$$C_{\max}(\sigma) \rightarrow \min \sim \max_{1 \leq v \leq m} \Delta_v(\sigma) \rightarrow \min.$$

Для введених величин має місце наступне:

$$\sum_{ij \in J_v(\sigma)} k_v p_{ij} = C^* - R_v(\sigma) + \Delta_v(\sigma), \quad v = \overline{1, m}. \quad (3.3)$$

Розділимо v -те рівняння в (3.3) на k_v ($k_v \neq 0$):

$$\sum_{ij \in J_v(\sigma)} p_{ij} = \frac{C^*}{k_v} - \frac{R_v(\sigma)}{k_v} + \frac{\Delta_v(\sigma)}{k_v}, \quad v = \overline{1, m},$$

$$\sum_{ij \in J_v(\sigma)} p_{ij} = c_v^* - R'_v(\sigma) + \Delta'_v(\sigma), \quad v = \overline{1, m}.$$

Застосовуючи методологію побудови ПДС-алгоритмів, згідно з якою поліноміальна складова алгоритму породжується логіко-аналітичними умовами, виконання яких гарантує оптимальність отриманого рішення, визначимо достатні умови оптимальності (ДУО) розкладів.

3.3 Опис програмної реалізації системи

Для складання розкладу виконання робіт на паралельних пристроях розроблено програмний продукт.

Даний програмний продукт представляє собою web-застосування.

Архітектуру програмного продукту наведено на рисунку 3.2.

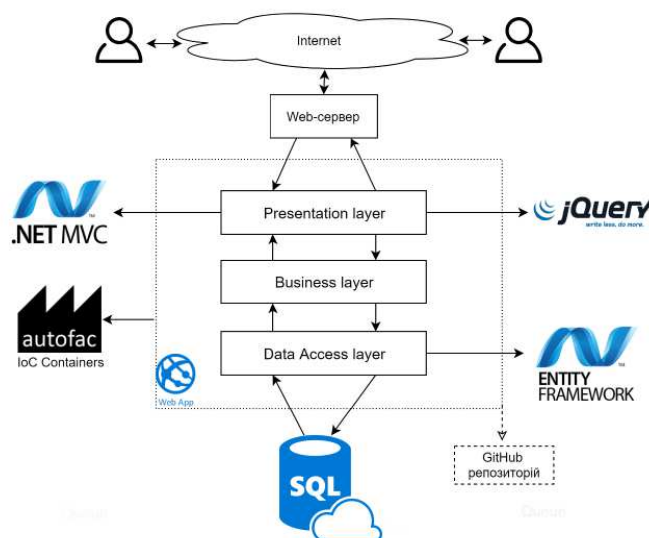


Рисунок 3.2 – Архітектура програмного продукту

Така архітектура дозволяє отримати наступні переваги:

- відокремлення логічних частин програми, що дозволяє їх окремо легко змінювати не коригуючи роботу тих частин, що не підлягали змінам;
- ізоляція сервера БД від клієнта шляхом розробки підсистеми уніфікованого доступу до даних, таким чином, забезпечується захист серверу від несанкціонованого доступу;
- розгортання програмного продукту на Windows Azure platform, що дозволяє не встановлювати його на комп'ютер користувача.

Компонентами програмного продукту є:

- Web сервер. У роботі, у його якості, виступає основний компонент IIS (Internet Information Services) – веб-сервер, який дозволяє розміщувати в Інтернеті сайти. IIS – це набір серверів для декількох служб Інтернету від компанії Майкрософт. IIS підтримує різні протоколи передачі даних;
- сервер БД для роботи з даними Windows Azure SQL Databases, що базується на Microsoft SQL Server. Це хмарний сервіс від корпорації Microsoft, що надає можливість зберігання і обробки реляційних даних. Мова, що використовується для запитів – Transact-SQL. Перевагою такого збереження даних є стабільність та надійність збереження даних;
- репозиторій контролю версій GitHub для зберігання програмного коду. Github є одним із найбільших репозиторіїв для спільної розробки програмного забезпечення. Він дозволяє переглядати збережений код, має функцію підсвічування синтаксису для різних мов програмування. Github працює з системою контролю версій Git;
- сервер застосувань, в основу реалізації якого покладені розроблені в роботі методи, що описані в розділі 3 цієї роботи. Він складається з застосування складання розкладу виконання робіт з відношенням передування паралельними пристроями за критерієм.

Розглянемо детальніше архітектуру сервера застосувань. Він представляє собою трирівневу архітектуру, наведену на рис. 3.2, яка складається з наступних рівнів:

а) Рівень представлення даних. Це є рівень, з яким безпосередньо взаємодіє користувач. Цей рівень включає компоненти для користувача інтерфейсу, механізм отримання введення від користувача. Стосовно до ASP.Net MVC на даному рівні розташовані всі компоненти, які складають призначений для користувача інтерфейс (стилі, статичні сторінки html, javascript), а також моделі уявлень, контролери, об'єкти контексту запиту;

б) Бізнес рівень. Він містить набір компонентів, які відповідають за обробку отриманих від рівня уявлення даних, реалізує всю необхідну логіку застосунку, все обчислення, взаємодію з базою даних і передає рівню подання результати обробки;

с) Рівень доступу до даних. Даний рівень зберігає моделі, що описують суті, що використовуються, також тут розміщуються клас контексту даних Entity Framework. Тут також зберігаються репозиторії, через які рівень бізнес-логіки взаємодіє з базою даних.

Вся логіка рівнів була написана за допомогою мови C#.

3.4 Обґрунтування вибору технологій розробки

Програмний продукт розроблений з використанням наступних технологій:

- високорівнева мова програмування C#;
- реляційна база даних Windows Azure SQL Databases;
- ASP.NET MVC Framework;
- IoC-контейнер Autofac;
- Entity Framework.

В якості інтегрованого середовища розробки було використано Visual Studio 2017. Дане середовище має широкий спектр різних засобів інтеграції з іншими засобами для розробки програмного забезпечення як системи контролю версій, web-сервери, покриття коду, фреймворки та ін.

Система контролю версій, яка була використана під час написання коду Git, є однією з найефективніших, надійних і високопродуктивних систем керування версіями, що надає гнучкі засоби нелінійної розробки, що базуються на відгалуженні та злитті гілок.

C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. Одна з найпотужніших мов, що швидко розвивається. C# є мовою з Сі-подібним синтаксисом і близький в цьому відношенні до C++ і Java. Сформулюємо основні переваги даної мови:

- об'єктно-орієнтована мова програмування;
- компонентно-орієнтований підхід до програмування;
- сприяє меншій машинно-архітектурній залежності результуючого програмного коду, гнучкості, переносимості і легкості повторного використання (фрагментів) програм;
- орієнтація на безпеку коду;
- уніфікована система типізації;
- розширена підтримка подієво-орієнтованого програмування.

Windows Azure SQL Databases – реляційна база даних, що базується на Microsoft SQL Server. Це хмарний сервіс від корпорації Microsoft, що надає можливість зберігання і обробки реляційних даних. Мова, що використовується для запитів – Transact-SQL. Перевагою такого збереження даних є стабільність та надійність збереження даних.

ASP.NET MVC Framework – фреймворк для створення веб-додатків, який реалізує шаблон Model-View-Controller. Платформа ASP.NET MVC базується на взаємодії трьох компонентів: контролера, моделі та представлення. Контролер приймає запити, обробляє введені дані користувача, взаємодіє з моделлю і представленням і повертає користувачу результат обробки запиту. Модель представляє рівень, що описує логіку організації даних у програмі. Представлення отримує дані з контролера і генерує елементи призначеного для користувача інтерфейсу для відображення інформації. Перевага даного використання даного фреймворка в наступному:

- більшість помилок видно на стадії розробки;
- призначені для користувача елементи управління дозволяють виділяти часто використовувані шаблони, такі як меню сайту;
- набір елементів управління і бібліотек класів дозволяє швидше розробляти додатки;
- можливість кешування даних, що використовуються на сторінці;
- можливість поділу візуальної частини та бізнес-логіки по різних файлах;
- розширена модель обробки запитів;
- наявність master-сторінок для завдання шаблонів оформлення сторінок;
- вбудована підтримка AJAX;
- має перевагу в швидкості в порівнянні з іншими технологіями, заснованими на скриптах.

ІоС-контейнер Autofac – фреймворк, який дозволяє використовувати архітектурне рішення інтеграції, яке спрощує розширення можливостей системи, при якому контроль над потоком управління програми залишається за каркасом.

.NET Entity Framework (EF) - об'єктно-орієнтована технологія доступу до даних, є object-relational mapping (ORM) рішенням для .NET Framework від Microsoft. Надає можливість взаємодії з об'єктами як за допомогою LINQ у вигляді LINQ to Entities, так і з використанням Entity SQL.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Вступ

На даний момент сучасні обчислювальні засоби широко використовуються в повсякденній діяльності людини. Сьогодні обчислювальна техніка дозволяє вирішувати найрізноманітніші задачі: від звільнення людини від ручної праці до розробки складних технічних розрахунків, що не під силу виконати одній людині. Сучасні засоби програмування дозволяють вирішувати задачі, що стояли в нещодавно перед цілими колективами наукових співробітників.

Розвиток нових технологій і ріст застосування обчислювальної техніки привели до широкого застосування ЕОМ для вирішення проблем, пов'язаних з обробкою інформації, її контролем і аналізом. Сьогодні на основі засобів обчислювальної техніки розробляються і впроваджуються різні автоматизовані інформаційні системи (проектування, технологічної підготовки виробництва).

Використання даного програмного продукту приведе до значної економії, що виражається в наступних показниках:

- підвищення продуктивності праці;
- прискорення обробки інформації;
- економія часу на обробку інформації.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування вартості програмного продукту. Очевидно, що вона являє собою дуже специфічний товар з безліччю особливостей.

Створення інформаційної системи вимагає одноразових витрат на її розробку, придбання необхідних технічних засобів і поточних витрат на функціонування системи. Економія від функціонування ІС визначається з урахуванням витрат на її експлуатацію. Відношення цієї економії до витрат на створення системи характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються по діючим на момент розрахунку оптовим цінам, тарифам і ставкам заробітної плати.

4.2 Розрахунок витрат на створення та експлуатацію інформаційної системи

Витрати на розробку системи складаються з витрат на зарплату розробника, на амортизацію ЕОМ, на якій виконується розробка, на експлуатацію цієї ЕОМ, на засоби розробки та витрат на матеріали і комплектуючі.

Виходячи з того, що в середньому заробітна плата розробників програмного забезпечення становить 8000 грн./місяць, вартість сучасного комп'ютера, який відповідає висунутим вимогам в третьому розділі, становить 15000 грн. і вартість кіловат-годин електроенергії становить 0,90 грн, розрахуємо вартість розробки системи. Витрати на допоміжні матеріали наведені в таблиці 4.1.

Таблиця 4.1

Витрати на допоміжні матеріали

Найменування витрат	Сума, грн.
Папір	100
Заправлення картриджа до принтера	140
Компакт-диски	40
Непередбачені витрати	200
Разом матеріали і комплектуючі вироби	480

Капітальні витрати на розробку системи K_{π} розраховуються за формулою:

$$K_{\pi} = C + (Z_{\text{пр}} + Z_{\text{е}}) \cdot T + Z_{\text{д}} \quad (1)$$

де C - первісна вартість програмного продукту, грн.;

$Z_{\text{пр}}$ - заробітна плата розробника, грн.;

$Z_{\text{е}}$ – експлуатаційні витрати, грн.;

T - тривалість розробки, міс.;

$Z_{\text{д}}$ - додаткові витрати, грн.;

Заробітна плата фахівця розраховується за формулою:

$$Z_{\text{пр}} = Z_{\text{озп}} \cdot \left(1 + \frac{A_c}{100}\right) \quad (2)$$

де $Z_{\text{озп}}$ – основна заробітна плата фахівця;

A_c - відсоток відрахувань на соціальне страхування;

$$Z_{\text{пр}} = 8000 \cdot (1 + 0,34) = 10720$$

Розрахуємо Z_e при споживанні потужності 0,3 Вт, тривалості роботи на місяць рівної $21 \cdot 8 = 168$ годин і вартості кіловат-години електроенергії 0,90 грн.,

$$Z_e = 168 \cdot 0,90 \cdot 0,3 = 45,36 \text{ грн.}$$

Поточні витрати на розробку системи представленні в таблиці 4.2

Таблиця 4.2

Витрати на розробку інформаційної системи

Найменування витрат	Одиниця виміру	Кількість
Тривалість розробки	міс.	4
Заробітна платня	грн.	8000
Відрахування на соціальні заходи	грн.	2720
Витрати на допоміжні матеріали	грн.	480
Витрати на електроенергію	грн.	45,36

Відповідно до формули (1) капітальні витрати на розробку системи складають:

$$K_{\text{п}} = 15000 + (10720 + 45,36) \cdot 4 + 480 = 58541,44 \text{ грн}$$

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік:

$$A_{\text{об}} = 15000 \cdot 0,6 = 9000 \text{ грн}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали визначаються в розмірі 5% вартості основного устаткування:

$$B_{\text{м}} = 15000 \cdot 0,05 = 750 \text{ грн}$$

Річний обсяг робіт ПЕОМ у годинах визначається в такий спосіб:

$$\Phi_{\text{м}} = 253,3 \cdot T_3$$

де T_3 - це середнє місячне завантаження устаткування (близько 7 годин), 253,3
- середня кількість робочих днів у році:

$$\Phi_M = 253,3 \cdot 7 = 1773,1 \text{ годин}$$

Витрати на електроенергію Z_e при 1773,1 годинах роботи устаткування в рік складуть:

$$Z_e = 1773,1 \cdot 0,3 \cdot 0,90 = 478,7 \text{ грн}$$

4.3 Економічна ефективність розробки і впровадження системи

Критерієм ефективності створення і впровадження нових засобів автоматизації є очікуваний економічний ефект. Він визначається за формулою:

$$E = \Delta P - E_n \cdot K_n \quad (3)$$

де ΔP - економія від підвищення продуктивності праці;

E_n - нормативний коефіцієнт ($E_n = 0.15$);

K_n - капітальні витрати на впровадження системи.

Якщо користувач системи при економії i -го виду робіт із застосуванням програми економить T_i годин, то підвищення продуктивності праці P_i (у %) визначається за формулою:

$$P_i = \left(\frac{\Delta T_j}{F_j - \Delta T_j} \right) \cdot 100\% \quad (4)$$

де F_j - час, який планувалося затратити користувачем для виконання роботи j -виду до впровадження системи.

В таблиці 4.3 представлена інформація про вид робіт, виконуваних користувачем, та час який він витрачає.

Таблиця 4.3

Таблиця робіт користувача

Вид робіт	До автоматизації, хв. F_j	Економія часу, хв. ΔT_j	Підвищення продуктивності праці P , %
Введення інформації	40	20	100
Проведення розрахунків	90	70	350

Аналіз і вибірка даних	50	25	100
Підготовка та друк звітів	30	15	100

Економія, пов'язана з підвищенням продуктивності праці користувача Р визначимо за формулою:

$$\Delta P = Z_{\pi} \cdot \sum_i \frac{P_i}{100}$$

де Z_{π} - середня заробітна плата менеджера.

Заробітна плата менеджера становить відповідно до формули (2)

$$Z_{\pi} = 5000 \cdot (1 + 0,34) = 6700 \text{ грн.}$$

Відповідно до формули (4) економія, пов'язана з підвищенням продуктивності праці користувача:

$$\Delta P = 6700 \cdot 6,5 = 43550 \text{ грн.}$$

У підсумку, відповідно до формули (4), отримуємо наступну очікувану економічну ефективність:

$$E = 43550 - 0,15 \cdot 58541,44 = 34768,78 \text{ грн.}$$

Строк окупності системи розраховується за формулою:

$$T = \frac{K_{\pi}}{\Delta P} = \frac{58541,44}{43550} \approx 1,34 \text{ року}$$

Отже строк окупності системи складає приблизно 1 рік 4 місяці.

5 ОХОРОНА ПРАЦІ

5.1 Вступ

Охорона праці – це система правил і заходів, які забезпечують безпечну роботу на даному виробництві.

Законодавство про працю містить норми і вимоги з техніки безпеки і виробничої санітарії, норми, що регулюють робочий час і час відпочинку, звільнення та переведення на іншу роботу, норми праці щодо жінок, молоді, гігієнічні норми і правила тощо.

Загальний нагляд за додержанням норм охорони праці покладено на прокуратуру, спеціальний — на професійні спілки. Контроль за безпекою праці здійснюють також державні й відомчі спеціалізовані інспекції.

У сучасному житті комп'ютер широко застосовується в житті людини: і удома, і в офісі, і в магазині, і у виробництві, і навіть в побутовій техніці — іншими словами комп'ютери міцно увійшли до повсякденного життя людей і їх використання постійно збільшується.

В офісах комп'ютери в основному використовуються як допоміжні засоби обробки інформації, і таке введення комп'ютерних технологій принципово змінило характер праці офісних працівників і вимоги до організації і охорони праці.

Недотримання вимог безпеки приводить до того, що через деякий час роботи за комп'ютером співробітник починає відчувати певний дискомфорт: у нього виникають головні болі і різь в очах, з'являються втома і дратівливість. У деяких людей порушується сон, погіршується зір, починають хворіти руки, шия, поперек і тому подібне.

5.2 Аналіз шкідливих та небезпечних факторів

При роботі з комп'ютером шкідливими і небезпечними чинниками є:

- електростатичні поля;
- електромагнітне випромінювання;
- наявність потужних іонізуючих випромінювань;
- локальне стомлення, загальне стомлення;
- стомлюваність очей;
- небезпека поразки електричним струмом;
- пожежонебезпека.

Всі ці чинники викликані двома основними причинами: неправильною роботою за обладнанням і неправильним вибором обладнання.

Негативний вплив комп'ютера на людину є комплексним.

По-перше, комп'ютер (особливо його дисплей, або монітор, сконструйований на основі ЕПТ (електронно-променевої трубки)) є джерелом електростатичного поля; електромагнітних випромінювань в низькочастотному, наднизькочастотних та високочастотному діапазонах (2 Гц - 400 кГц); випромінювання оптичного діапазону (ультрафіолетового, інфрачервоного і видимого світла), рентгенівського випромінювання.

Вчені встановили, що випромінювання низької частоти в першу чергу негативно впливає на центральну нервову систему, викликаючи головний біль / запаморочення, нудоту, депресію, безсоння, відсутність апетиту, стрес.

Низькочастотне електромагнітне поле може з'явитися причиною шкірних захворювань (вугрі, екзема, рожевий лишай та ін), хвороб серцево-судинної системи та шлунково-кишкового тракту; воно впливає на білі кров'яні тілця, що призводить до виникнення пухлин, у тому числі і злоякісних.

Електростатичне поле великої напруженості здатне змінювати і переривати клітинне розвиток, а також в окремих випадках викликати катаракту.

Дисплеї з ЕПТ завдяки електростатичного поля інтенсивно притягають і збирають пил, а значить, можуть бути причиною появи дерматитів на шкірі обличчя, загострення бронхіальної астми, подразнення слизових оболонок.

По-друге, нерухома і напружена поза.

По-третє, інтенсивна робота з клавіатурою викликає больові відчуття в ліктьових суглобах, передпліччях, зап'ястях, в кистях та пальцях рук.

Відповідно до законодавчих актів площа робочого місця користувача ПК повинна складати не менше 4,5м². У приміщеннях повинні проводитися щоденне вологе прибирання і систематичне провітрювання після кожної години роботи. Устаткування, що виробляє багато шуму (друкуючі пристрої, сканери, сервери і тому подібні), рівні шуму якого перевищують нормативні, повинне розміщуватися поза робочими місцями співробітників.

Робочі столи слід розміщувати так, щоб монітори були орієнтовані бічною стороною до світлових отворів, щоб природне світло падало переважно зліва.

При розміщенні робочих місць відстань між робочими столами має бути не менше 2,0 м-коду, а відстань між бічними поверхнями відеомоніторів — не менше 1,2 м. Робочі місця співробітників, що виконують творчу роботу і що вимагає значної розумової напруги або високої концентрації уваги, рекомендується ізолювати один від одного перегородками заввишки від 1,5 м.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні використовуваного устаткування. Висота робочої поверхні столу повинна складати 725 мм, робоча поверхня столу повинна мати ширину 800..1400 мм і глибину 800..1000 мм. Робочий стіл повинен мати простір для ніг заввишки не менше 600 мм, шириною — не менше 500 мм, завглибшки на рівні колін — не менше 450 мм і на рівні витягнутих ніг — не менше 650 мм.

Конструкція робочого стільця або крісла повинна забезпечувати підтримку раціональної робочої пози працівника і дозволяти змінювати позу з метою зниження статичної напруги м'язів шийно-плечової області і спини. Робочий стілець або крісло мають бути підйнятно-поворотними, регульованими по

висоті і кутам нахилу сидіння і спинки, а також відстані спинки від переднього краю сидіння, при цьому регулювання кожного параметра має бути незалежним, легко здійснюваним і мати надійну фіксацію.

Клавіатуру слід розташовувати на поверхні столу на відстані 100..300 мм від краю, зверненого до користувача, або на спеціальній поверхні, відокремленій від основної.

Екран відеомонітора повинен знаходитися від очей користувача на відстані 600..700 мм, але не ближче 500.

5.3 Розрахунок системи штучного освітлення приміщення відділу інформаційних технологій

У відділі інформаційних технологій використовується загальне штучне освітлення, тобто освітлення приміщень штучним світлом за допомогою електричних ламп, яке використовується при недостатньому природному освітленню.

Необхідно зробити розрахунок загального освітлення виробничого приміщення. Довжина $A = 40$ м, ширина $B = 14$ м, висота $H = 3,5$ м. Висота виробничої поверхні $h_p = 1$ м. Для освітлення приймаємо світильники типу НСП 07. Мінімальна освітленість лампи розжарювання за нормами $E_{\min} = 100$ лк. Коефіцієнт відображення стелі $S_n = 70\%$, стін $S_c = 50\%$, робочої поверхні $S_p = 30\%$. Напруга мережі 220 В.

Розрахунок здійснюємо за методом коефіцієнта використання світлового потоку.

Відстань від стелі до робочої поверхні

$$H_o = H - h_p = 3,5 - 1 = 2,5 \text{ м.}$$

Відстань від стелі до світильника

$$h_c = 0,2 \cdot H_o = 0,2 \cdot 2,5 = 0,5 \text{ м.}$$

Висота підвішування світильника над освітлювальною поверхнею

$$h = H_o - h_c = 2,5 - 0,5 = 2 \text{ м.}$$

Висота підвішування світильника над підлогою

$$H_n = h + h_p = 2 + 1 = 3 \text{ м.}$$

Для досягнення найбільшої рівномірності освітлення приймаємо відношення $L/h = 1,5$. Тоді відстань між центром світильників складає

$$L = 1,5 \cdot h = 1,5 \cdot 2 = 3 \text{ м.}$$

Необхідна кількість світильників

$$N = S/L^2 = (40 \cdot 14) / 3^2 = 62,22 \text{ світ.}$$

Приймаємо 63 світильники - 7 рядів по 9 світильників (рисунок 5.1).

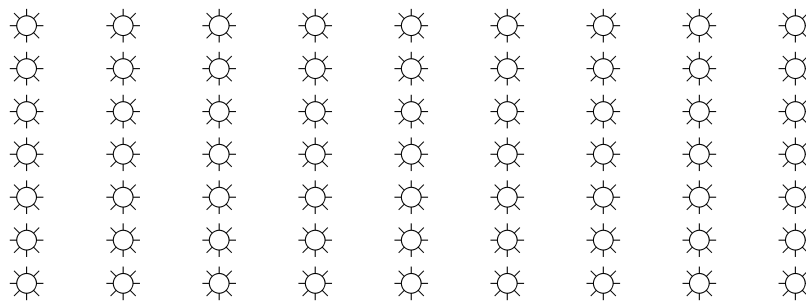


Рисунок 5.1 - Розташування світильників у виробничому приміщенні

Індекс приміщення дорівнює

$$i = A \cdot B / (h \cdot (A + B)) = (40 \cdot 14) / (2 \cdot (40 + 14)) = 560 / 108 = 5,185$$

При $i = 5,185$; $S_n = 70\%$, $S_c = 50\%$, $S_p = 30\%$ для світильника типу НСП 07 коефіцієнт використання світлового потоку $\eta = 0,9$.

Світловий потік однієї лампи

$$\Phi = (E_{min} S K_z Z) / (N \eta) = (100 \cdot 560 \cdot 1,3 \cdot 1,15) / (63 \cdot 0,9) = 1476,54 \text{ лм,}$$

де $K_z = 1,3$ за таблицею значень коефіцієнту запасу; $Z = 1,15$ для ламп розжарювання.

За знайденим світловим потоком з таблиці параметрів ламп розжарювання загального призначення вибираємо лампу потужністю 100 Вт, що має світловий потік $\Phi_{л} = 1450$ лм, найбільш близький до розрахункового.

При цьому фактична освітленість

$$E_{\phi} = E_{min} \cdot (\Phi_{л} / \Phi_p) = 100 \cdot (1450 / 1476,54) = 98,2 \text{ лк.}$$

Загальна потужність освітлювальної установки

$$P_3 = P_{\text{л}} \cdot N = 100 \cdot 63 = 6300 \text{ Вт} = 6,3 \text{ кВт.}$$

Отже для оптимальної роботи в приміщенні необхідно встановити 7 рядів по 9 світильників. Загальна потужність усіх світильників складає 6,3 кВт.

5.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів при роботі з персональним комп'ютером

5.4.1 Заходи щодо зменшення впливу

Заходи щодо усунення небезпеки поразки електричним струмом зводяться до правильного розміщення устаткування та електричних кабелів. Інші заходи щодо забезпечення електробезпеки, збігаються з загальними заходами пожежоготова електробезпеки.

В якості профілактичних заходів для забезпечення пожежної безпеки слід використовувати скриту електромережу, надійні розетки з пожежобезпечних матеріалів, силові мережі живлення устаткування виконувати кабелями, розрахованими на підключення в 3-5 разів більшого навантаження, включати й виключати живлення обладнання за допомогою штатних вимикачів. Треба регулярно робити очистку внутрішніх частин комп'ютерів, іншого устаткування від пилу, розташовувати комп'ютери на окремих не спалюваних столах. Для запобігання іскріння необхідно рідше встромляти і виймати штепсельні вилки з розеток.

Екран дисплея повинен бути розташованим перпендикулярно до напрямку погляду. Якщо він розташований під кутом, то стає причиною сутулості. Відстань від дисплея до очей повинна трохи перевищувати звичну відстань між книгою та очима. Перед екраном монітора, особливо старих типів, повинен бути спеціальний захисний екран. При його відсутності треба сидіти на відстані витягнутої руки від монітора. Ще одним моментом, який стосується зору, є необхідність створення неоднорідного поля зору. Для цього можна розвісити на

поверхнях (стінах) плакати та картини, виконані у спокійних тонах. Наприклад, пейзажі.

Важливою є форма спинки крісла, яка повинна повторювати форму спини. Висота крісла повинна бути такою, щоб користувач не відчував тиску на куприк або стегна. Крісло бажано обладнати бильцями. Його потрібно встановити так, щоб не треба було тягтися до клавіатури. Періодично користувачу необхідно рухатися, вчасно змінювати положення тіла і робити перерви у роботі.

При напруженій роботі за комп'ютером щогодини необхідно робити перерву на 15 хвилин через кожну годину і треба займатися іншою справою. Декілька разів на годину бажано виконувати серію легких вправ для розслаблення.

5.4.2 Техніка безпеки при роботі на комп'ютері

Загальні положення:

1) При виконанні робіт на персональних комп'ютерах (ПК) необхідно дотримуватись вимог загальної та даної інструкцій з охорони праці.

2) До самостійної роботи на комп'ютерах допускаються особи, які пройшли медичний огляд, навчання з професії, ввідний інструктаж з охорони праці і первинний інструктаж з охорони праці на робочому місці. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя.

3) При роботі на комп'ютерах, до складу яких входять відеодисплейні термінали (ВДТ), на працівників можуть діяти наступні шкідливі та погрожуючі життю і здоров'ю виробничі фактори:

а) фізичні :

- підвищений рівень шуму (від вентилятора мікропроцесора та плат);
- високий рівень напруги в електричній мережі, торкання до якої може викликати нещасний випадок;
- підвищений рівень статичної електрики;
- підвищений рівень електромагнітного випромінювання;

- підвищений рівень напруженості електричного поля;
- пряма і відбита від екрана блескiсть;
- несприятливе розподiлення яскравостi об'єктiв в полі зору.

б) психофізіологічні:

- фізичні перевантаження статичної та динамічної дії;
- нервово-психічні перевантаження (розумове перенапруження, перенапруження зорових аналізаторів, монотонність праці, емоційні перевантаження).

4) Основним обладнанням робочого місця користувача ПК є: ВДТ, системний блок, клавіатура, друкуючий пристрій (принтер), зовнішні пристрої пам'яті, сканер, "миша", блок безперервного живлення, робочий стіл, стілець та інше; допоміжним – шафи, полиці, сейф та ін.

Робочі місця з ВДТ повинні бути розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін – на відстані 1 м, між собою – на відстані не менше 1,5 м. Відносно вікон робочі місця доцільно розташувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

5) Робочі місця, обладнані ВДТ, слід розташовувати таким чином, щоб запобігти попаданню в очі прямого світла. Джерела освітлення слід розташовувати з обох боків екрану, паралельно напрямку зору.

Для запобігання світлових відблисків від екрана, клавіатури в напрямку очей користувача від освітлювачів загального призначення або сонячних променів необхідно застосовувати антивідблискові сітки, спеціальні фільтри, захисні козирки, на вікнах регульовані жалюзі.

Фільтри із металевої або нейлонової сітки використовувати не рекомендується тому, що сітка спотворює зображення внаслідок інтерференції світла. Найкраща якість зображення забезпечується скляними поляризаційними фільтрами – вони усувають практично усі відблиски і забезпечують чітке та контрастне зображення.

6) При роботі з текстовою інформацією (в режимі введення даних, редагування тексту і читання з екрану ВДТ) найбільш фізіологічним є зображення чорних знаків на світлому (білому) фоні.

7) Розташовувати ВДТ на робочому місці слід так, щоб поверхня екрану знаходилась в центрі поля зору на відстані 400-700 мм від очей користувача. Пропонується розташовувати елементи робочого місця таким чином, щоб підтримувалась однакова відстань від очей користувача до екрана, клавіатури, пулітра.

8) Робота комп'ютера супроводжується електромагнітним випромінюванням різних частот і рівнів, інтенсивність котрого зменшується пропорційно квадрату відстані до екрану. Оптимальною для працюючого за ПК є відстань 50 см від екрану ВДТ.

9) Раціональною робочою позою працюючого за ПК вважається таке розташування тіла, при якому ступні працівника розташовані горизонтально на підлозі або на підставці для ніг, стегна зорієнтовані в горизонтальній площині, верхні кінцівки рук – вертикально, кут ліктьового суглоба коливається в межах 70-90 град., зап'ястя зігнуті під кутом, не перевищує 20 град., нахил голови – в межах 15-20 град.

10) Для нейтралізації зарядів статичної електрики в приміщенні, де виконуються роботи на комп'ютерах, в тому числі на лазерних та світлодіодних принтерах, пропонується збільшувати вологість повітря за допомогою кімнатних зволожувачів. Не рекомендується носити одяг з синтетичних матеріалів.

Вимоги безпеки перед початком роботи:

1) Перевірити надійність встановлення апаратури на робочому столі. ВДТ не повинен стояти на краю столу. Повернути ВДТ так, щоб було зручно дивитись на екран – під прямим кутом (а не збоку) і трохи згори вниз, а екран повинен бути нахиленим під невеликим кутом (нижній край ближче до оператора).

2) Перевірити загальний стан апаратури, справність проводки електромережі, з'єднувальних шнурів, штепсельних вилок та розеток, заземлення захисного екрана.

4) Відрегулювати освітленість робочого місця.

5) Відрегулювати і зафіксувати висоту стільця (крісла), зручний для користувача нахил його спинки.

6) Під'єднати до системного блока необхідну апаратуру (принтер, сканер і т.п.). Всі кабелі, що з'єднують системний блок з іншими пристроями слід вставляти і виймати тільки при вимкненому електричному живленні.

7) Увімкнути апаратуру ПК вмикачами на корпусах в такій послідовності: стабілізатор (або блок безперервного живлення), ВДТ, системний блок, принтер (якщо передбачається друкування).

8) Відрегулювати яскравість світіння екрану ВДТ, фокусировку, контрастність. Не слід робити яскравість екрану занадто великою, тому що це втомлює очі (і зменшує термін роботи електронно-променевої трубки).

Рекомендується:

- яскравість світіння екрану – не менше 100 кд/ кв.м (кандел на кв.м);
- відношення яскравості екрану ВДТ до яскравості оточуючих його поверхонь в робочій зоні – не більше 3:1;
- мінімальний розмір точки світіння – не більше 0,4 мм для монохромного ВДТ і не більше 0,6 мм для кольорового;
- контрастність зображення знака – не менше 0,8.

9) При появі несправностей комп'ютерної техніки повідомити керівника.

Вимоги безпеки при закінчення роботи на ПК:

1) Закінчити і зберегти в пам'яті ПК файли, які знаходились у роботі. Виконати всі дії для коректного завершення роботи в оперативній системі.

2) Вимкнути принтер та інші периферійні пристрої, вимкнути ВДТ і системний блок. Вимкнути стабілізатор (або блок безперервного живлення при

його наявності). Штепсельні вилки витягнути з розеток. Накрити клавіатуру кришкою для попередження попадання в неї пилу.

3) Навести порядок на робочому місці.

4) Вимкнути освітлення та електроприлади.

Вимоги безпеки при аварійних ситуаціях:

1) При раптовому припиненні подачі електричної енергії вимкнути всі пристрої ПК в такій послідовності: периферійні пристрої, ВДТ, системний блок, стабілізатор (або блок безперервного живлення). Витягнути вилки з розеток.

2) При наявності ознак горіння (дим, запах горілого) необхідно вимкнути всі пристрої ПК. Знайти місце загоряння і виконати всі можливі заходи для його ліквідації, попередивши терміново про це керівництво.

3) У випадку виникнення пожежі негайно попередити про це пожежну частину та керівництво, виконати усі можливі заходи по евакуації людей з приміщення і розпочати гасіння пожежі первинними засобами пожежогасіння.

6 ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ

Охорона навколишнього середовища — система заходів щодо раціонального використання природних ресурсів, збереження особливо цінних та унікальних природних комплексів і забезпечення екологічної безпеки, також це сукупність державних, адміністративних, правових, економічних, політичних і суспільних заходів, спрямованих на раціональне використання, відтворення і збереження природних ресурсів землі, обмеження негативного впливу людської діяльності на навколишнє середовище.

Відносини у галузі охорони навколишнього середовища регулюються Законом України «Про охорону навколишнього природного середовища», а також земельним, водним, лісовим законодавством, законодавством про надра, про охорону атмосферного повітря, про охорону і використання рослинного і тваринного світу та іншим спеціальним законодавством.

Метою охорони навколишнього середовища є протидія негативним змінам у навколишньому середовищі, які мали місце в минулому, відбуваються зараз або можуть бути.

Природоохоронні заходи мають на меті поліпшення стану навколишнього природного середовища або створення умов для цього. Ознаками природоохоронних заходів є:

- підвищення екологічності продукції, що випускається;
- скорочення споживання природних ресурсів на одиницю продукції, що випускається та здійснення господарської діяльності;
- зниження забруднення природних комплексів викидами, стоками, відходами, випромінюваннями;
- зниження концентрації шкідливих речовин у викидах, стоках, відходах;
- поліпшення стану середовища існування людей.

Актуальність охорони довкілля, що перетворилася в глобальну проблему, пов'язана зі зростаючим антропогенним впливом. Це зумовлено демографічним вибухом, урбанізацією, що прискорюється розвитком комунікацій, забрудненням навколишнього середовища відходами, надмірним навантаженням на орні землі, пасовища, ліси, водойми.

6.2 Забруднення навколишнього середовища

При виробництві різних композиційних матеріалів відбувається викид великої кількості техногенних відходів в навколишнє середовище, що суттєво забруднює атмосферу.

Оскільки на підприємствах для виробництва КМ використовуються різноманітні технології, то і в довкілля потрапляють різні забруднюючі речовини.

Основними джерелами утворення і викидів в атмосферу забруднюючих речовин є наступні технологічні процеси: підготовка сировини (збагачування, агломерація), виробництво коксу, доменне виробництво, виробництво сталі, виробництво прокату.

Наприклад бетонні виробництва потребують близько 20 технологічних процесів (розмелювання, висушування, подрібнення, нагрівання в печах, охолодження в баштах, транспортування на стрічкових конвеєрах, транспортування готового продукту тощо), що супроводжуються забрудненням повітря, причому кожен з них має особливості і створює власні технічні проблеми. Найбільш складними технологічними блоками є хімічні підприємства. На одному заводі ряд виробничих процесів може спричинити викиди різних забруднюючих речовин, включно з газоподібними (наприклад, при виробництві азотної, сірчаної кислот, віскози та добрив, а також теплової енергії в котельнях).

Для аглофабрики кількість агломераційних газів, що викидаються в атмосферу, становить - $1,83-106 \text{ м}^3 / \text{год}$, їх запиленість $8-10 \text{ г} / \text{м}^3$, зміст CO_2 - $4,2\%$, CO - $0,6-0,9\%$, SO_2 - $0,05\%$. Під час агломерації руди відбувається вигорання сірки з тиритів. Сульфідні руди містять до 10% сірки, а після агломерації її залишається лише $0,2 - 0,8\%$. Викид сірчистого газу при агломерації може бути у розмірі 190 кг . на 1 т . руди, тобто одна стрічкова машина дає близько 700 т . сірчистого газу за добу.

У доменному виробництві мають місце викиди в атмосферу продуктів згорання з доменної печі (колошникових газів) у процесах завантаження шихти, випуску чавуну, охолодження шлаку. Також разом з доменним газом в атмосферу в невеликих кількостях викидаються також сполуки миш'яку, фосфору, сурми, свинцю, пари ртуті і рідких металів, ціанистий водень і смолисті речовини.

У загальному балансі пилогазових викидів сталеплавільних агрегатів на частку виробництва сталі припадає 75% пилу, 85% оксидів азоту, 90% діоксиду сірки. Запиленість димів газів становить $0,5-2 \text{ г} / \text{м}^3$.

Значну роль у забрудненні атмосфери грають викиди мартенівських і конверторних сталеплавильних цехів. При веденні мартенівського процесу пил утворюється з металевої шахти (сталевий брухт при її окислюванні, зі шлаку, руди, вапняку, що йдуть на окислювання домішок шихти, і з доломіту, що застосовується для заправлення підлоги печі). У період кипіння сталі виділяються також пари окисів шлаку і металу, газу. Розміри часток пилу не перевищує 3 мкм . Переважна частина пороши мартенівських печей складається з триокису заліза (67%) і триокису алюмінію ($8,7\%$). При безкисневому процесі на 1 т . мартенівської сталі виділяється $3000 - 4000 \text{ м}^3$ газів з концентрацією пилу в середньому $0,5 \text{ г/м}^3$. У період подачі кисню в зону розплавленого металу пилоутворення багаторазово збільшується, досягаючи $15-52 \text{ г/м}^3$. Крім того, плавлення сталі супроводжується вигоранням деякої кількості вуглецю і сірки, у зв'язку з чим у газах мартенівських печей, що відходять при кисневому дутті, міститься до 60 кг . окису вуглецю і до 3 кг . сірчистого газу в розрахунку на 1 т .

виданої сталі. Головною особливістю конверторного процесу є одержання сталі з рідкого чавуну без застосування палива. Виплавка сталі за таким принципом здійснюється в конвертерах ємністю 50,100, 250 і більше тонн шляхом продування рідкого чавуну киснем. Це забезпечує вигорання небажаних домішок, наприклад, марганцю, фосфору і вуглецю, що містяться у чавуні. Процес одержання конверторної сталі носить циклічний характер і при кисневому дутті триває 25-30 хвилин. Димові гази, що утворюються, складаються з часток окисів кремнію, марганцю і фосфору. У складі диму міститься значна кількість окису вуглецю - до 80%. Концентрація пилу в газах що відходять, складає приблизно 15 г/м³. Основна маса пилу -90%, складається з часток розміром 0,2 -1 мкм.

У процесі виробництва коксу мають місце викиди в атмосферу коксового газу. Коксохімічні підприємства забруднюють атмосферне повітря пилом і сумішшю летких сполук. Процес коксування полягає в термічній, без доступу повітря, обробці кам'яновугільної шихти при температурі 900 - 1100вС. При коксуванні 1т. вугілля утворюється 300-320 м³ коксового газу. У його сполуки входять: водень - 50-62% (об'ємних), метан -20-34%, окис вуглецю - 4,5-4,7%, вуглекислий газ -1,8-4%, азот - 5-10%, вуглеводні - 2-2,6% і кисень -0,2-0,5%.

Основна маса коксового газу вловлюється і направляється на хімічну переробку й утилізацію, близько 6% газу надходить в атмосферу внаслідок втрат під час завантаження і вивантаження печей і негерметичності апаратури. У деяких випадках, наприклад, при порушенні режиму роботи батарей коксування чи допоміжних відділень утилізації, в атмосферу викидаються значні кількості неочищеного коксового газу.

6.3 Розробка заходів щодо зменшення забруднення навколишнього середовища

Охорона навколишнього середовища на підприємстві характеризується комплексом вжитих заходів, які спрямовані на попередження негативного

впливу людської діяльності підприємства на навколишню природу, що забезпечує сприятливі та безпечні умови людської життєдіяльності. Враховуючи стрімкий розвиток науково-технічного прогресу, перед людством постала складна задача — охорона найважливіших складових навколишнього середовища (земля, вода, повітря), схильних сильному забрудненню техногенними відходами і викидами, що призводить до окислення ґрунту і води, руйнування озонового шару землі та кліматичним змінам. Промислова політика всього світу привела до таких незворотних і суттєвих змін в навколишньому середовищі, що це питання (охорона навколишнього середовища на підприємстві) став загальносвітовою проблемою і примусив державні апарати розробити довгострокову екологічну політику зі створення внутрішньодержавного контролю.

Основними умовами для поліпшення екології в країні є: раціональне використання, охорона і трата запасів природного резерву, забезпечення безпеки екології та протирадіаційні заходи, підвищення і формування екологічного мислення у населення, а також контроль над екологією в промисловості.

Охорона навколишнього середовища на підприємстві визначила ряд заходів для зниження рівня забруднень, що виробляється підприємствами:

- виявлення, оцінка, постійний контроль та обмеження викиду шкідливих елементів в атмосферу, а також створення технологій і техніки, які охороняють і зберігають природу і її ресурси.
- розробка правових законів, спрямованих на охоронні заходи навколишнього середовища та матеріальне стимулювання виконаних вимог і профілактики комплексу природоохоронних заходів.
- профілактика екологічної обстановки шляхом виділення спеціально відведених територій (зон).

Суттєвого зменшення забруднення навколишньої середовища підприємствами можна досягти впровадженням таких заходів:

1) Реалізація засад ресурсозбереження в усьому ланцюзі суспільного виробництва на шляху його інтенсифікації і зниження ресурсомісткості (водо-, земле-, метало-, енерго-);

2) Впровадження безвідходних технологій, що означає (і має своїм наслідком): комплексне використання сировини і енергоносіїв, створення замкнених газо- і водооборотних систем, застосування принципово нових підходів до вилучення, збагачення, перероблення сировини і матеріалів – біотехнологій, геотехнологій.

При вирішенні проблеми безвідходності виробництва слід мати на увазі дві сторони єдиного процесу. Перше – це найбільш раціональних видобуток та повне використання ресурсів і як наслідок зменшення утворення відходів. Друге – це розширення використання відходів, що утворюються. Ці шляхи не виключають, а взаємно доповнюють один одного.

Поняття безвідходних технологій дещо умовним, оскільки повної безвідходності досягти практично не можливо. Більш коректним буде говорити про маловідходні технології. При цьому мається на увазі можливість створення технологічних систем, вплив яких на природу не буде перевищувати її відновлювального потенціалу.

В основі концепції безвідходних технологій лягли три основні положення, а саме:

- створення максимально замкнених систем, організованих за аналогією з природними екосистемами;
- раціональне використання всіх компонентів сировини;
- неминучі впливи на навколишнє середовище не повинні порушувати його функціонування.

ВИСНОВОК

У представленій роботі вирішено актуальну науково-практичну задачу підвищення продуктивності праці та якості виробів за рахунок впровадження інформаційної системи для оптимізації планування випуску м'ясної продукції компанії.

У результаті виконання магістерської роботи було вирішено наступні задачі:

- Проведено аналіз процесу виробництва ковбасних виробів;
- Проведено аналіз процесу планування випуску продукції;
- Проведено системний аналіз методів оптимізації в якості інструмента для планування;
- Проведено дослідження існуючих систем управління виробництва.
- Розроблено математичні моделі, які можуть використатися для розробки ІС.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Андейчук, В.Г. Економіка аграрних підприємств: Підручник. - 2-изд.
2. Антонов А.Н. Основы современной организации производства / А.Н. Антонов, Л.С. Морозова. – М.: Дело и сервис, 2004. – 578 с.
3. Бабук, І.М. Економіка підприємства: навч. посібник для студентів технічних спеціальностей [Текст] / І.М.Бабук. - Мінськ: "ІВЦ Мінфіну", 2006.
4. Буч, Г. Язык UML. Руководство пользователя [Текст] / Г.Буч, Дж. Рамбо, А. Джекобсон. – К.: ДМК пресс, 2007. – 496 с.
5. Горнев В.Ф. Оперативное управление в ГПС / В.Ф. Горнев, В.В. Емельянов, М.В. Овсянников. – М.: Машиностроение, 1990. – 256 с.
6. Джигирей, В. С. Екологія та охорона навколишнього природного середовища: Навчальний посібник [Текст] / В. С. Джигирей. - 3-є вид., випр. і доп. - К.: Т-во "Знання", КОО, 2004. - 309 с.
7. Екологічний менеджмент: Навчальний посібник [Текст] / В. Ф. Семенов, О. Л. Михайлюк, Т. П. Галушкіна, Г. В. Крусір та ін.; За ред. В. Ф. Семенова, О. Л. Михайлюк ; М-во освіти і науки України, ОДЕУ. - К. : Центр навчальної літератури, 2004. - 407 с.
8. Ємельянова, Ф. Н. Організація переробки сільськогосподарської продукції. Навчальний посібник і практикум [Текст] / Ф. Н. Ємельянова, М. К. Кирилов – М.: Асоціація авторів, і видавців “ТАНДЕМ”. Видавництво “ЭКМОС”, 2000. – 384 с.
9. Жукова И.С. Проблемы организации производства в условиях смены технологических укладов / И.С. Жукова // Теоретические основы и практика организации производства: Юбилейный сб. науч. трудов. — Воронеж : ВГТУ , 2010. – С. 13 - 20.
10. Закон України про охорону навколишнього природного середовища [Текст] : затверджено і введено в дію Постановою Верховної Ради України №1268-ХІІ від 26 червня 1991, чинний, поточна редакція від 18 листопада 2012.

11. Інструкція з охорони праці при роботі на комп'ютерах [Текст] : затверджено і введено в дію наказом Мінінформа України №75 від 05 червня 1996 р.
12. Ковалев В. В. Анализ деятельности предприятия: учеб. [Текст] / В. В. Ковалев, О. Н. Волкова. — М.: издательство «Проспект», 2004. — 424 с.
13. Коробин, В. И. Екологія: Підручник для вузів [Текст] / В. И. Коробин, Л. В. Передельський. — Ростовн / Д: вид-во «Фенікс», 2000. — 576 с.
14. Мизюн, В. А. Управление производственными системами и процессами [Електронне видання] / В.А. Мизюн. — Самара: Издательство СНЦ РАН, 2012. — 211 с. — Режим доступу до ресурсу: http://www.cfin.ru/management/manufact/manufacturing_sys-03.shtml .
15. Мизюн, В. А. Модель конкурентоспособного производства / В. А. Мизюн // Аудит и финансовый анализ. — 2009. — № 5. — С. 314 — 344.
16. Мильнер, Б. З. Теория организации. / Б.З. Мильнер. — 6-е изд. (перераб. и доп.). — М.: ИНФРА-М, 2008. — 797 с.
17. Мыльник, В. В. Исследование систем управления / В.В. Мыльник, Б.П. Титаренко, В.А. Волочиенко. — 4-е издание. — М.: Академический проект, 2006. — 352 с.
18. Родионова, В. Н. Понятие и механизм синхронизации производственных процессов / В.Н. Родионова // Организатор производства. — М.: Экономика и финансы, 2010. — № 3. — С.15–18.
19. Румянцева, Е. Л. Информационные технологии [Текст] / Е. Л. Румянцева, В. В. Слюсарь. - М.: Форум, Инфра-М, 2007. — 256 с.
20. Трофимова, Л. А. Управленческие решения (методы принятия и реализации) : учебное пособие [Текст] / Л. А. Трофимова, В. В. Трофимов. - СПб. : Изд-во СПбГУЭФ, 2011. — 190 с.
21. Тубальцев, А. М. Виробниче освітлення та його розрахунок: Навчальний посібник [Текст] / А. М. Тубальцев. — Миколаїв: УДМТУ, 2001. — 84 с.

ДОДАТОК А – Технічне завдання

1 Загальні положення

Назва інформаційної системи: «ІС ЯрмарОК смаколиків».

Створюваний програмний продукт являє собою ІС для оптимізації планування випуску м'ясної продукції. Дана автоматизована система забезпечує оптимізацію шляхів транспортування товару від постачальника до споживача.

Підстави для розробки: Наказ №1232-уч від 12.11.2021 «Про затвердження тем магістерських робіт».

2 Призначення та мета створення системи

2.1 Призначення

Дана система призначена для оптимізації шляхів транспортування товару від постачальника до споживача.

2.2 Мета створення

Метою створення даної системи є підвищення продуктивності праці та якості виробів за рахунок впровадження інформаційної системи для оптимізації планування випуску м'ясної продукції компанії.

3 Характеристика об'єкту автоматизації

Об'єктом автоматизації даного проекту є оптимізація шляхів транспортування товару від постачальника до споживача. Під автоматизацію підпадає функції ведення бази даних продукції компанії, а також генерація

альтернатив по оптимізації планування. Дані процеси описані в розділі 1 та 2 даної роботи.

4 Цілі та завдання розробки системи

Метою даної ІС є:

- збільшення кількості користувачів, що працюють із системою;
- зведення до мінімуму суб'єктивних помилок при оптимізації;
- підвищення достовірності даних;
- розмежування доступу до сховища даних по правах користувачів;
- створення єдиної інформаційної системи;
- підвищення надійності функціонування.

Створювана ІС повинна виконувати наступні функції:

- Ведення бази даних продукції;
- Ведення бази даних сировини;
- Ведення бази даних постачальників;
- Генерація альтернатив.

5 Вимоги до системи

5.1 Вимоги до функціональних характеристик

Програмне забезпечення повинно передбачати наступні функції:

- Додавати інформацію про продукцію;
- Проводити пошук альтернативних заготівок;
- Корегувати знайдений шаблон планування;
- Проектувати технологічні операції;
- Генерувати альтернативні плани випуску;

- Зберегти шаблон випуску;
- Затвердити план випуску;
- Роздрукувати документацію по проекту.

5.2 Вимоги до надійності

Для коректного функціонування програмного додатку необхідно забезпечувати:

- Безперебійне живлення технічних засобів;
- Використання ліцензійного програмного забезпечення;
- Випробування програмних засобів на наявність комп'ютерних вірусів.

Необхідно слідкувати за коректністю введення даних. Якщо обов'язкові поля не заповнені програма повинна видати про це повідомлення і не продовжувати подальшу роботу то того часу як дані поля не будуть заповнені. Також якщо введена інформація не відповідає типу поля програма повинна видати про це попередження. При введенні помилкової інформації програма повинна забезпечувати її редагування.

5.3 Вимоги до експлуатації

Треба враховувати що відділ технологів може розташовуватись в цехових приміщеннях. Комп'ютер технолога повинний бути спеціально призначений для експлуатації в жорстких цехових умовах. Ця спецтехніка має найвищу ступінь захисту від пилу і відповідає рівню IP65 по стандарту ISO9001, що забезпечує пило-ізоляцію та безпроблемний контакт з водою.

Від користувача вимагається знання інструкції користувача, та загальні вміння користуватися ПК.

5.4 Вимоги до складу та параметрів технічних засобів

- серверна платформа CPU AMD ATHLON 64 3800+ (2.4 ГГц), RAM 4 Гб, HDD 500 Гб і серверне програмне забезпечення (MS Windows Server);
- клієнтські платформи : CPU AMD Sempron 3000+ (1,6 ГГц), RAM 4 Гб, HDD 500 Гб та клієнтське програмне забезпечення (MS Windows 7);
- мережне середовище (100 Mbit);

Апаратне забезпечення повинне максимально повно охоплювати існуючий парк комп'ютерної техніки на підприємстві.

6 Вимоги до програмної документації

Програмна документація повинна містити:

- Технічне завдання;
- Технічний опис;
- Програму та методики випробувань;
- Інструкцію користувача;

7 Стадії та етапи розробки

Роботи по створенню системи включають в себе наступні етапи та стадії розробки:

- Перед проектне дослідження;
- Аналіз предметної галузі;
- Оформлення вимог користувача до системи;
- Розробка концепції системи;
- Розробка варіантів концепцій системи та їх функціональний аналіз;
- Вибір варіанту концепції системи.

- Ескізне проектування;
- Прийняття загальносистемних рішень;
- Розробка попередніх рішень по вибору варіантів інформаційного, технічного та програмного забезпечення системи.

- Технічне проектування;
- Прийняття проектних рішень по системі та її частинам;
- Розробка технічної документації по проекту.
- Ввід в дію:
- Виготовлення та налагодження компонентів системи;
- Розробка акту передачі.

8 Порядок прийому та контролю

Порядок контролю і здачі даного програмного продукту виконується представником замовника у присутності представника розробника згідно програмі і методиці випробування.

Для контролю і здачі даного програмного продукту повинна бути представлена розроблена ІС, методика випробувань і інструкція користувача.

За результатами здачі складається акт, який підписується представником замовника і представником розробника, і затверджується керівниками організації-замовника і організації-розробника.

Додаток В – Текст програми

```
package YarSmak;

public class Main {

    mainForm mf;

    public static void main(String[] args) {
        Main prog = new Main();
        prog.run();
    }

    public void run() {
        mf = new mainForm();
        mf.setVisible(true);
    }
}

package YarSmak;

public class mainForm extends javax.swing.JFrame {

    projectForm pf;

    public mainForm() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox();
        jTextField1 = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
);

        jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(new
```

```

javax.swing.border.LineBorder(new java.awt.Color(0, 0, 0), 1,
true), " IC Ярмарок смаколиків "));

jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

    jLabel1.setText("Загальна інформація:");

    jLabel2.setText("Назва проекту:");

    jLabel3.setText("Автор:");

    javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGroup(jPanel2Layout.createParallelGroup(
                javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addComponent(jLabel2)
                    .addPreferredGap(javax.swing.La
                        outStyle.ComponentPlacement.RELATED)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 162,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addComponent(jLabel3)
                    .addPreferredGap(javax.swing.La
                        outStyle.ComponentPlacement.RELATED)
                    .addComponent(jComboBox1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
            .addContainerGap())
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.La
                outStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel2)
            .addPreferredGap(javax.swing.La
                outStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel3)
            .addPreferredGap(javax.swing.La
                outStyle.ComponentPlacement.RELATED)
            .addComponent(jComboBox1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap())
    );
    jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGroup(jPanel2Layout.createParallelGroup(
                javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel1)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addComponent(jLabel2)
                    .addPreferredGap(javax.swing.La
                        outStyle.ComponentPlacement.RELATED)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 162,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addComponent(jLabel3)
                    .addPreferredGap(javax.swing.La
                        outStyle.ComponentPlacement.RELATED)
                    .addComponent(jComboBox1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
            .addContainerGap())
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.La
                outStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel2)
            .addPreferredGap(javax.swing.La
                outStyle.ComponentPlacement.RELATED)
            .addComponent(jLabel3)
            .addPreferredGap(javax.swing.La
                outStyle.ComponentPlacement.RELATED)
            .addComponent(jComboBox1, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addContainerGap())
    );
    jPanel2Layout.setDefaultCloseOperation(
        javax.swing.WindowConstants.EXIT_ON_CLOSE);
    jPanel2Layout.setSize(new Dimension(400, 300));
    jPanel2.setVisible(true);
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
}

```

```

        .addContainerGap()
        .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.BASELINE)
        .addComponent(jLabel3)
        .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    jButton1.setText("X");
    jButton1.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Початок оптимізації");
    jButton2.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton2ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

```

```

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
    .addGroup(jPanell1Layout.createSequentialGroup())
        .addContainerGap()

    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayou
t.Alignment.LEADING)
        .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addGroup(jPanell1Layout.createSequentialGroup()
        .addComponent(jButton1)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE, 222, Short.MAX_VALUE)))
        .addContainerGap())
);
jPanell1Layout.setVerticalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
    .addGroup(jPanell1Layout.createSequentialGroup())
        .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

    .addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayou
t.Alignment.BASELINE)
        .addComponent(jButton1)
        .addComponent(jButton2))

    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
        .addGroup(layout.createSequentialGroup())

```



```

        .addContainerGap()
        .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

    pack();
} // </editor-fold>

private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
}

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    pf = new projectForm();
    pf.setVisible(true);
    this.setVisible(false);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {
            new mainForm().setVisible(true);
        }
    });
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JLabel jLabel1;

```

```

        private javax.swing.JLabel jLabel2;
        private javax.swing.JLabel jLabel3;
        private javax.swing.JPanel jPanel1;
        private javax.swing.JPanel jPanel2;
        private javax.swing.JTextField jTextField1;
        // End of variables declaration
    }
    package YarSmak;

    public class projectForm extends javax.swing.JFrame {

        mainForm mf;
        processForm pf;

        public projectForm() {
            initComponents();
        }

        @SuppressWarnings("unchecked")
        // <editor-fold defaultstate="collapsed" desc="Generated
Code">
        private void initComponents() {

            jPanel1 = new javax.swing.JPanel();
            jPanel2 = new javax.swing.JPanel();
            jLabel1 = new javax.swing.JLabel();
            jLabel2 = new javax.swing.JLabel();
            jTextField1 = new javax.swing.JTextField();
            jLabel3 = new javax.swing.JLabel();
            jLabel4 = new javax.swing.JLabel();
            jComboBox1 = new javax.swing.JComboBox();
            jComboBox2 = new javax.swing.JComboBox();
            jLabel5 = new javax.swing.JLabel();
            jComboBox3 = new javax.swing.JComboBox();
            jButton1 = new javax.swing.JButton();
            jButton2 = new javax.swing.JButton();
            jButton3 = new javax.swing.JButton();
            jButton4 = new javax.swing.JButton();
            jPanel3 = new javax.swing.JPanel();
            jScrollPane1 = new javax.swing.JScrollPane();
            jTextArea1 = new javax.swing.JTextArea();
            jLabel6 = new javax.swing.JLabel();
            jCheckBox1 = new javax.swing.JCheckBox();
            jComboBox4 = new javax.swing.JComboBox();

            setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
);

            jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(jav

```

```

ax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0,
0)), " Оптимізація "));

jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

        jLabel1.setFont(new java.awt.Font("Tahoma", 1, 11)); //
NOI18N
        jLabel1.setText("Інформація про продукцію:");

        jLabel2.setText("Назва:");

        jLabel3.setText("ДСТУ:");

        jLabel4.setText("Матеріал:");

        jLabel5.setText("Модель:");

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                        .addGroup(jPanel2Layout.createParallelGroup(
                        javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(jPanel2Layout.createParallelGroup(
                                javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(jPanel2Layout.createSequentialGroup()
                                                .addComponent(jLabel5)

                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
                                        .addComponent(jComboBox3, 0, 192,
Short.MAX_VALUE))
                                                .addComponent(jLabel1)

                                .addGroup(jPanel2Layout.createSequentialGroup()
                                        .addGroup(jPanel2Layout.createSequentialGroup()
                                                .addComponent(jLabel2)

                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
                                        .addComponent(jTextField1,
javax.swing.GroupLayout.DEFAULT_SIZE, 237, Short.MAX_VALUE))

                                .addGroup(jPanel2Layout.createSequentialGroup()
                                        .addGroup(jPanel2Layout.createSequentialGroup()
                                                .addComponent(jLabel3)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
                                .addComponent(jComboBox2, 0, 239,
Short.MAX_VALUE))

.addGroup(jPanel2Layout.createSequentialGroup()
                                .addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
                                .addComponent(jComboBox1, 0, 221,
Short.MAX_VALUE))
                                .addContainerGap())
);
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                                .addGroup(jPanel2Layout.createSequentialGroup()
                                    .addContainerGap()
                                    .addComponent(jLabel11)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
                                .addComponent(jLabel2)
                                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
                                .addComponent(jLabel3)
                                .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
                                .addComponent(jLabel4)

```

```

        .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.BASELINE)
        .addComponent(jLabel5)
        .addComponent(jComboBox3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    jButton1.setText("<");
    jButton1.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jButton2.setText("Проектувати ТО");
    jButton2.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton2ActionPerformed(evt);
        }
    });

    jButton3.setText("X");
    jButton3.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            jButton3ActionPerformed(evt);
        }
    });

    jButton4.setText("Підібрати заготівку");
    jButton4.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {

```

```

        jButton4ActionPerformed(evt);
    }
});

jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

    jTextArea1.setColumns(20);
    jTextArea1.setRows(5);
    jScrollPane1.setViewportView(jTextArea1);

    jLabel6.setFont(new java.awt.Font("Tahoma", 1, 11)); //
NOI18N
    jLabel6.setText("Альтернативи заготовок:");

    javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
    jPanel3.setLayout(jPanel3Layout);
    jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(10, 10, 10)
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.Grou
pLayout.Alignment.LEADING)
                .addComponent(jLabel6)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 153, Short.MAX_VALUE))
            .addContainerGap(10, true)
            .addComponent(jCheckBox1)
            .addContainerGap(10, true)
            .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(10, true)
        );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel6)
            .addGap(10, 10, 10)
            .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 153, Short.MAX_VALUE)
            .addGap(10, 10, 10)
            .addComponent(jCheckBox1)
            .addGap(10, 10, 10)
            .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(10, true)
        );
    jPanel3Layout.setDefaultCloseOperation(
    javax.swing.WindowConstants.EXIT_ON_CLOSE);
    jPanel3Layout.setVisible(true);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
}

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayou
t.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayou
t.Alignment.LEADING)
                .addComponent(jCheckBox1)
                .addComponent(jComboBox4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(91, 91, 91))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap()))
);

    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayou
t.Alignment.TRAILING, false)
                    .addComponent(jPanel3,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addComponent(jPanel2,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel1Layout.createSequentialGroup()
                    .addComponent(jButton3)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

        .addComponent(jButton1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

        .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addComponent(jButton4,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );
    jPanell1Layout.setVerticalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanell1Layout.createSequentialGroup())
        .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

        .addComponent(jButton4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

        .addComponent(jPanel3,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayou
t.Alignment.BASELINE)
        .addComponent(jButton3)
        .addComponent(jButton1)
        .addComponent(jButton2))
        .addContainerGap())
    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());

```



```

        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            );
        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanell1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
            );

        pack();
    } // </editor-fold>

    private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
    }

    private void
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        mf = new mainForm();
        this.setVisible(false);
        mf.setVisible(true);
    }

    private void
jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

    }

    private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        pf = new processForm();
        this.setVisible(false);
        pf.setVisible(true);
    }

```

```

    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                new projectForm().setVisible(true);
            }

        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JCheckBox jCheckBox1;
    private javax.swing.JComboBox jComboBox1;
    private javax.swing.JComboBox jComboBox2;
    private javax.swing.JComboBox jComboBox3;
    private javax.swing.JComboBox jComboBox4;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
package YarSmak;

public class processForm extends javax.swing.JFrame {

    public processForm() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated
Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();

```

```

jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jCheckBox1 = new javax.swing.JCheckBox();
jComboBox1 = new javax.swing.JComboBox();
jPanel3 = new javax.swing.JPanel();
jLabel2 = new javax.swing.JLabel();
jScrollPane2 = new javax.swing.JScrollPane();
jTextArea2 = new javax.swing.JTextArea();
jLabel3 = new javax.swing.JLabel();
jScrollPane3 = new javax.swing.JScrollPane();
jTextArea3 = new javax.swing.JTextArea();
jButton3 = new javax.swing.JButton();
jCheckBox2 = new javax.swing.JCheckBox();
jComboBox2 = new javax.swing.JComboBox();
jButton4 = new javax.swing.JButton();
jButton5 = new javax.swing.JButton();
jButton6 = new javax.swing.JButton();
jButton7 = new javax.swing.JButton();
jButton8 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE
);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(jav
ax.swing.BorderFactory.createLineBorder(new java.awt.Color(0, 0,
0)), " Оптимізація "));

jPanel2.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

    jLabel1.setText("Планування випуску:");

    jTextArea1.setColumns(20);
    jTextArea1.setRows(5);
    jScrollPane1.setViewportView(jTextArea1);

    jButton1.setText("Затвердити план");

    jButton2.setText("Корегувати шаблон");

    javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup())
            .addContainerGap()

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jLabel1,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE, 193, Short.MAX_VALUE)

.addGroup(jPanel2Layout.createSequentialGroup()
        .addComponent(jCheckBox1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jComboBox1, 0, 170,
Short.MAX_VALUE))
        .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE, 193, Short.MAX_VALUE))
        .addContainerGap()
    );
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup())

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(jCheckBox1))

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D, 40, Short.MAX_VALUE)
        .addComponent(jButton1)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jButton2))
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 113, Short.MAX_VALUE))
        .addContainerGap()
    );

jPanel3.setBorder(javax.swing.BorderFactory.createLineBorder(new
java.awt.Color(0, 0, 0)));

    jLabel2.setText("Згенеровані альтернативи:");

    jTextArea2.setColumns(20);
    jTextArea2.setRows(5);
    jScrollPane2.setViewportView(jTextArea2);

    jLabel3.setText("Порядок планування:");

    jTextArea3.setColumns(20);
    jTextArea3.setRows(5);
    jScrollPane3.setViewportView(jTextArea3);

    jButton3.setText("Копіювати");

    jButton4.setText("Затвердити план");

    jButton5.setText("Зберегти у якості шаблону");

    javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
    jPanel3.setLayout(jPanel3Layout);
    jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
            .addContainerGap()

    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayou
t.Alignment.TRAILING)
        .addComponent(jButton5,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 317, Short.MAX_VALUE)

```

```

        .addComponent(jButton4,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 317, Short.MAX_VALUE)

        .addGroup(jPanel3Layout.createSequentialGroup())

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.LEADING)
            .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 140, Short.MAX_VALUE)
            .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.LEADING)
            .addComponent(jScrollPane3,
javax.swing.GroupLayout.DEFAULT_SIZE, 171, Short.MAX_VALUE)
            .addComponent(jButton3,
javax.swing.GroupLayout.DEFAULT_SIZE, 171, Short.MAX_VALUE)
            .addComponent(jLabel3,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

        .addGroup(jPanel3Layout.createSequentialGroup())
            .addComponent(jCheckBox2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
            .addComponent(jComboBox2, 0, 294,
Short.MAX_VALUE))
        .addContainerGap())
    );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup())
            .addContainerGap()
            .addComponent(jLabel2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

        .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.LEADING, false)

        .addGroup(jPanel3Layout.createSequentialGroup())
            .addComponent(jLabel3)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jScrollPane3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jButton3)
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 147,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout
t.Alignment.TRAILING)
        .addComponent(jCheckBox2)
        .addComponent(jComboBox2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jButton4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATE
D)
        .addComponent(jButton5)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
    );

    jButton6.setText("<");

    jButton7.setText("X");

    jButton8.setText("Роздрукувати документацію по проекту");

    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup())
        .addContainerGap()

```

```

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel3,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(jPanel2,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanell1Layout.createSequentialGroup()
    .addComponent(jButton6)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jButton8,
        javax.swing.GroupLayout.DEFAULT_SIZE, 247, Short.MAX_VALUE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jButton7))
    .addContainerGap()
);
jPanell1Layout.setVerticalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanell1Layout.createSequentialGroup()
        .addComponent(jPanel2,
            javax.swing.GroupLayout.PREFERRED_SIZE,
            javax.swing.GroupLayout.DEFAULT_SIZE,
            javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jPanel3,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jButton6)
    .addComponent(jButton7)
    .addComponent(jButton8))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
    Short.MAX_VALUE))

```



```

    );

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanell1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

    pack();
} // </editor-fold>

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {

        public void run() {
            new processForm().setVisible(true);
        }

    });
}
// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton6;

```

```
private javax.swing.JButton jButton7;
private javax.swing.JButton jButton8;
private javax.swing.JCheckBox jCheckBox1;
private javax.swing.JCheckBox jCheckBox2;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JScrollPane jScrollPane3;
private javax.swing.JTextArea jTextArea1;
private javax.swing.JTextArea jTextArea2;
private javax.swing.JTextArea jTextArea3;
// End of variables declaration
}
```