

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ПРИРОДОКОРИСТУВАННЯ**

**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

КВАЛІФІКАЦІЙНА РОБОТА

першого (бакалаврського) рівня вищої освіти

на тему: **«Розробка компонентів інформаційної системи
планування використання технічного оснащення
фермерського господарства»**

Виконала: студент 4 курсу групи Кн-41сп

Спеціальності 122 «Комп'ютерні науки»

(шифр і назва)

Павлюх Павло Ігорович

(Прізвище та ініціали)

Керівник: к.т.н., в.о. доцента Татомир А.В.

(Прізвище та ініціали)

Рецензент: к.т.н., доцент Кригуль Р.Є.

(Прізвище та ініціали)

ДУБЛЯНИ-2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ПРИРОДОКОРИСТУВАННЯ
ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Перший (бакалаврський) рівень вищої освіти
Спеціальність 122 «Комп'ютерні науки»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____

д.т.н., проф. А. М. Тригуба

«____» _____ 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу студенту

Павлюху Павлу Ігоровичу

1. Тема роботи: «Розробка компонентів інформаційної системи планування використання технічного оснащення фермерського господарства»

Керівник роботи Татомир Андрій Володимирович, в.о. доцента
затверджені наказом по університету від 27.11.2023 року № 641/к-с.

2. Строк подання студентом роботи 10.06.2024 р.

3. Вихідні дані до роботи: вимоги до процесів планування використання технічного оснащення сільськогосподарського підприємства; засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, методика побудова планів.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити) _____

Вступ.

1. Аналіз предметної області.

2. Постановка задачі.

3. Проектування інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства.

4. Практичне використання інтелектуальної інформаційної системи планування використання технічного оснащення.

5.

Охорона праці.

Висновки та пропозиції.

Список використаної літератури.

5. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових креслень): особливості планування використання технічного оснащення сільськогосподарського підприємства; огляд інформаційних систем аналогів; результати вибору засобів реалізації проекту; планування робіт у проекті; результати проектування інтелектуальної інформаційної системи; особливості реалізації та практичного використання інтелектуальної інформаційної системи.

6. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3, 4	<i>Татомир А.В., в.о. доцента кафедри ІТ</i>		
5	<i>Городецький І.М., доцент кафедри фізики, інженерної механіки та безпеки виробництва</i>		

7. Дата видачі завдання

27 листопада 2023 р.

Календарний план

№ з/п	Назва етапів кваліфікаційної роботи	Терміни виконання етапів роботи	При-мітка
1	<i>Написання першого розділу</i>	<i>27.11-31.12.23</i>	
2	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>01.02-05.03.24</i>	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>06.03-14.04.24</i>	
4.	<i>Виконання четвертого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>15.04-10.05.24</i>	
5.	<i>Написання розділу «Охорона праці»</i>	<i>11.05-23.05.24</i>	
6.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	<i>24-31.05.24</i>	
7.	<i>Завершення роботи в цілому</i>	<i>01-10.06.24</i>	

Студент _____ Павлюх П.І.
(підпис)

Керівник роботи _____ Татомир А.В.
(підпис)

УДК 004.9 : 631.1

Розробка компонентів інформаційної системи планування використання технічного оснащення фермерського господарства.

Велика Ю.Я. Кафедра ІТ – Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 61с. текст. част., 12 рис., 3 табл., 10 арк. ілюстраційного матеріалу, 23 джерела.

Подано особливості планування використання технічного оснащення сільськогосподарського підприємства. Виконано аналіз існуючих інтелектуальних інформаційних систем аналогів. Виконано опис запропонованої ідеї проекту.

Сформульовано мету, задачі та використовуваний інструментарій проекту інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства. Здійснено вибір засобів для виконання проекту та проведено його планування. Виконано проектування інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства. Здійснено моделювання діяльності інтелектуальної інформаційної системи.

Наведено етапи розробки інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства, а також особливості практичного її використання.

Розроблено заходи із охорони праці під час використання інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства.

ЗМІСТ

ВСТУП	7
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1. Особливості планування використання технічного оснащення сільськогосподарського підприємства.....	8
1.2. Огляд інтелектуальних інформаційних систем для планування діяльності у сільськогосподарських підприємствах.....	10
1.3. Ідентифікація ідеї проекту.....	15
2. МАТЕМАТИЧНИЙ ОПИС ЗАДАЧІ ТА ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЕКТУ.....	17
2.1. Математичний опис задачі та вибір методу її розв’язання.....	17
2.2. Опис методу планування використання технічного оснащення	18
2.3. Вибір засобів для виконання проекту	21
3. ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЛАНУВАННЯ ВИКОРИСТАННЯ ТЕХНІЧНОГО ОСНАЩЕННЯ СІЛЬСЬКОГОСПОДАРСЬКОГО ПІДПРИЄМСТВА.....	24
3.1. Побудова моделі варіантів використання інтелектуальної інформаційної системи.....	24
3.2. Побудова моделі класів інтелектуальної інформаційної системи.....	26
3.3. Побудова моделі послідовності інтелектуальної інформаційної системи....	28
3.4. Побудова моделі діяльності інтелектуальної інформаційної системи.....	30
4. РОЗРОБКА КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЛАНУВАННЯ ВИКОРИСТАННЯ ТЕХНІЧНОГО ОСНАЩЕННЯ ФЕРМЕРСЬКОГО ГОСПОДАРСТВА	32
4.1. Створення вікна користувача інтелектуальної інформаційної системи ...	32
4.2. Створення модуля для формування діалогового вікна користувача інтелектуальної інформаційної системи	35
4.3. Створення модуля планування використання технічного оснащення та виведення результатів	37

	6
5. ОХОРОНА ПРАЦІ	41
5.1. Аналіз стану шуму у приміщенні	41
5.2. Заходи щодо електробезпеки	43
5.3. Заходи щодо покращення пожежної безпеки.....	45
5.4. Інструкція з охорони праці під час розроблення інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства	47
5.4.1. Загальні положення.....	47
5.4.2. Вимоги безпеки перед початком роботи.....	49
5.4.3. Вимоги безпеки під час роботи та її завершення	49
5.4.4. Вимоги безпеки в аварійних ситуаціях.....	50
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	51
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	55
ДОДАТКИ	57

ВСТУП

Існуючі інтелектуальні інформаційні системи у сільськогосподарському виробництві є важливою передумовою для взаємодії із багатьма структурними підрозділами зазначеного виробництва. Продуктивність технічних засобів та засобів виробництва у сільському господарстві можна підвищити за допомогою відповідних, надійних та корисних інформаційних систем. Такі системи допомагають керівникам сільськогосподарських підприємств приймати кращі та точні рішення. Тому існує потреба розроблення інтелектуальних інформаційних систем у сільськогосподарському виробництві, які враховуватимуть особливості функціонування конкретного сільськогосподарського підприємства, щоб керувати ними та вдосконалювати їх діяльність.

Досягнення сталого розвитку сільського господарства ґрунтується менше на матеріальних витратах (наприклад, насіння та добрива), ніж на техніці та людях, залучених до їх діяльності. Для досягнення бажаного ефекту прийняття правильних управлінських рішень необхідно зосередитися на інтелектуальних інформаційних системах, які плануватимуть діяльність технічного забезпечення та відповідно використання людських ресурсів. Це відповідно забезпечить розширення знань та обміну інформацією про сільськогосподарське виробництво, а також на відповідних комунікаційних засобів, каналів та інструментів.

У кваліфікаційній роботі обґрунтовано особливості проектування та використання інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства, яка є досить актуальною на даний час.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Особливості планування використання технічного оснащення сільськогосподарського підприємства

Точне планування використання технічного оснащення сільськогосподарського підприємства, зокрема визначення потреби в технічних засобах за їх видами, марочним складом та економічною ефективністю є важливою умовою успішного сільськогосподарського виробництва [3]. Якщо у сільськогосподарському підприємстві є технічне оснащення, що задовольняє їх потреби, то значно підвищується ефективність їх використання і зменшується вартість обладнання на одиницю роботи чи продукту.

Для розрахунку потреби в тракторах та сільськогосподарській техніці необхідно знати обсяги робіт, терміни виконання, норми виробітку в зміну, коефіцієнт мінливості виробничих та кліматичних умов, коефіцієнт технічної готовності тракторного парку.

Для визначення потреб у тракторах, комбайнах та інших технічних засобах використовують різні методи:

- за загальним обсягом механізованих робіт, як правило, у переведенні на еталонні гектари;
- завантаженість у найбільш напружені періоди;
- нормативний метод;
- метод лінійного програмування.

Обґрунтовуючи розміри та структуру тракторного парку, уважно й детально слід аналізувати природно-виробничі умови сільськогосподарського підприємства, перспективи розвитку рослинництва та тваринництва, а також вивчити чинники, що впливають на рівень технології (стан ґрунту, довжина та ширина поля, рельєф, метеорологічні умови тощо).

Потреба у сільськогосподарській техніці визначається на підставі порівняльної економічної оцінки технологічних карт на виконання механізованих робіт, порівняльної економічної оцінки техніки та тракторів за зведеним планом механізованих робіт, за графіком роботи технічного оснащення впродовж календарного року.

Технологічні карти передбачають відображення повного переліку робіт у хронологічному порядку, обсягів кожного виду робіт, календарних термінів та кількості робочих днів, склад технічного оснащення, продуктивність технічного оснащення, прямі витрати на оплату праці та витрати на паливно-енергетичні ресурси.

При плануванні технологічних карт на виробництво сільськогосподарської продукції відзначається підвищення ефективності трудомістких процесів завдяки впровадженню науки і передового досвіду. Карти складаються на основі наявного технічного оснащення та фактичного плану виконання робіт. Для кожного виду робіт підбирається найбільш ефективне технічне оснащення. Це завдання вирішується на основі аналізу даних економічної оцінки технічного оснащення, що дозволяє вибрати агрегат з найбільшим ефектом, а саме із забезпеченням найвищої продуктивності, забезпечення економії витрат ресурсів на одиницю виконаної роботи.

Потреба у технічному оснащенні визначається виходячи з обсягу робіт і часу їх виконання, відповідно до технологічних карт, заданих марок тракторів, сільськогосподарських машин, які поєднуються з певними тракторами. Сюди входять урожайність, норми виробітку за одиницю часу, планова кількість механізованих робіт та терміни їх виконання, продуктивність комплектних агрегатів.

Потреба в тракторах розраховується як у звичайних одиницях з використанням обсягу робіт на звичайний еталонний гектар, так і в натуральних для кожної марки тракторів з урахуванням інтенсивних періодів виконання польових робіт. Потреба в тракторах, комбайнах та інших технічних засобах розраховується за формулою:

$$K_m = \frac{O}{H \cdot G \cdot D \cdot K_{mg}}, \quad (1.1)$$

де K_m – кількість технічного оснащення, од.; O – обсяг виконання робіт, ум. ет. га.; H – норма виробітку технічного оснащення, ум. ет. га / год.; G – тривалість роботи (впродовж доби), год; D – кількість робочих днів; K_{mg} – коефіцієнт технічної готовності технічного оснащення (відношення кількості технічно справного технічного оснащення до загальної їх кількості у сільськогосподарському підприємстві).

Слід використовувати обсяг виконання робіт, що виконуються певними марками у заданий період, щоб визначати потребу в фірмових тракторах (комбайнах) [4]. При цьому розрахунок проводиться за формулою:

$$K_m = \frac{O}{H_{zm} \cdot K_{zm} \cdot D \cdot K_{mg} \cdot K_y}, \quad (1.2)$$

де K_m – кількість потрібного технічного оснащення певної марки, од.; O – обсяг виконання робіт в ум. ет. га. у напружений період їх виконання для цієї марки, га; H_{zm} – змінна норма виробітку технічного оснащення заданої марки, га; K_{zm} – коефіцієнт використання часу зміни; D – кількість робочих днів у напружений період; K_{mg} – коефіцієнт технічної готовності технічного оснащення певної марки; K_y – коефіцієнт метеорологічних умов.

1.2. Огляд інтелектуальних інформаційних систем для планування діяльності у сільськогосподарських підприємствах

На даний час спостерігається використання широкого спектру інтелектуальних інформаційних систем у діяльності сільськогосподарських підприємств. Це сприяє організації конкурентоспроможного виробництва, дає можливість знаходити сучасні технології та методи їх впровадження в довгостроковій перспективі, сприяє розробці бізнес-планів для виявлення виробничих проблем та обґрунтування методів для їх вирішення [13].

Інтелектуальні інформаційні системи для планування діяльності у сільськогосподарських підприємствах можна розділити на такі основні групи. Зокрема, інтелектуальні інформаційні системи управління на основі роботи електронної карти поля та доданої бази даних (Геоінформаційні системи).

У сільському господарстві ГІС забезпечує:

- швидке отримання інформації про економіку та виробничий процес;
- формування звітів і завдань;
- планування та моделювання технологічних операцій;
- надавати інформаційну підтримку в прийнятті рішень;
- у деяких випадках обробка даних із бортових комп'ютерів сільськогосподарського технічного оснащення та формування карт внесення для диференційованого внесення добрив.

До даних інтелектуальних інформаційних систем відносяться:

- «ГЕО-Агро»;
- «ГІС Панорама Землеробство»;
- «Farm Works Site (Pro)»;
- «SST Summit»;
- «SMS Desktop Software (Advanced і Basic)»;
- «JD Reports MAP»;
- «АграрОфіс»;
- «Agro-Net NG»;
- «FarmView Record Keeper та інші».

Також є інтелектуальні інформаційні системи для кишенькового ПК. Як правило, такі інформаційні системи є доповненням до програмного забезпечення, встановленого на робочому столі. До них належать «Farm Truck Mate», «SST Stratus» та інші.

Інтелектуальна інформаційна система «Farm Works Site (Pro)» являє собою програмне забезпечення для управління, яке допомагає збирати, структурувати та обробляти інформацію про всі технологічні операції з вирощування сільськогосподарських культур. Це програмне забезпечення для

сільського господарства дозволяє створювати звіти, створювати електронні карти в області розподілу параметрів. «Farm Works Site (Pro)» зручно використовувати як у невеликому господарстві, так і великих агрохолдингах. На рис. 1.1 показаний зовнішній вигляд описаної вище інформаційної системи.

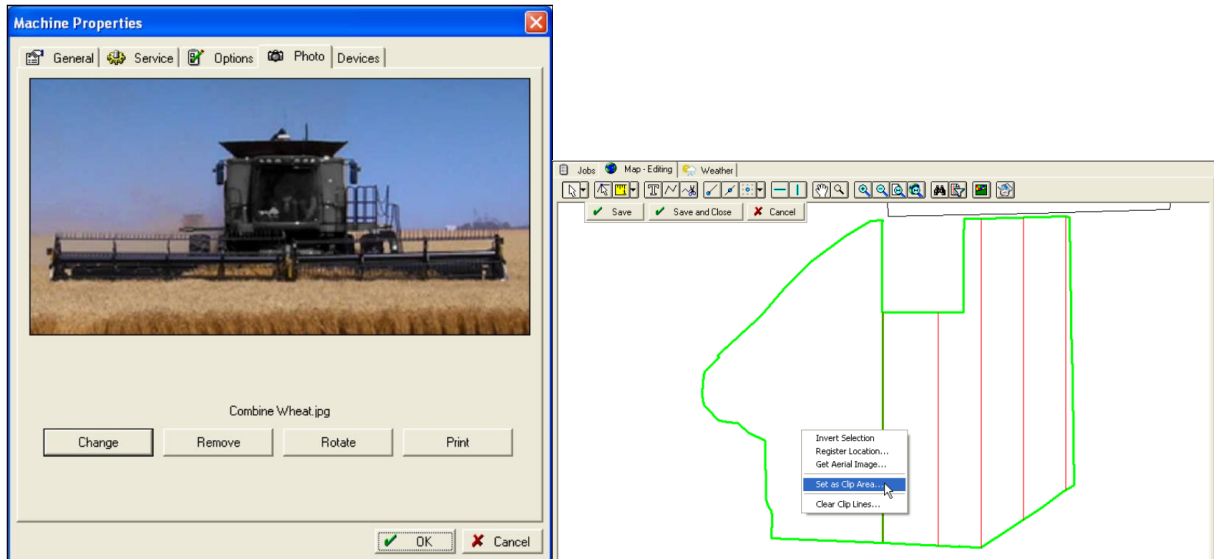


Рис. 1.1. Інтелектуальна інформаційна система «Farm Works Site (Pro)»

Інтелектуальна інформаційна система «Аграр Офіс» розроблена компанією «Land-Data Eurosoft» (Німеччина). Понад 30 років компанія «Land-Data Eurosoft» займається розробкою комп'ютерного програмного забезпечення для сільськогосподарських підприємств, розробляючи багато інноваційних проектів – від індивідуальних рішень автоматизації виробництва до створення комплексної системи управління сільськогосподарським виробництвом.

Інтелектуальна інформаційна система «Аграр Офіс» являє собою програмне забезпечення для всіх типів сільськогосподарських підприємств, яке є помічником у прийнятті рішень у сфері агробізнесу. «Land-Data Eurosoft» є лідером на європейському ринку програмного забезпечення. Ця програма дозволяє аналізувати виробничий процес, автоматично готувати необхідну документацію, точно і якісно складати плани виробництва. Вона збирає всю інформацію, необхідну для агротехнологічного планування.

Інженер, який отримує від системи інформацію про точний обсяг запланованих робіт, встановлює графік ремонту та обслуговування обладнання,

заміни робочих органів на інструменті. Система «Аграр-Офіс» включає ряд модулів, серед яких «Журнал полів» (рис. 1.2).

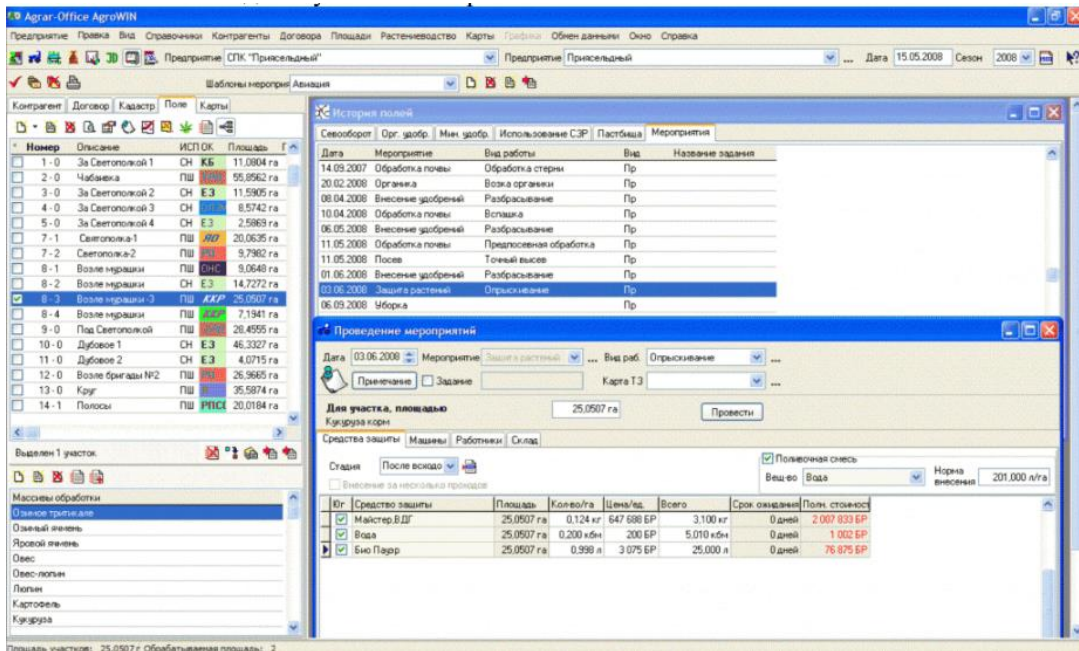


Рис. 1.2. Модуль польового журналу інтелектуальної інформаційної системи «Аграр Офіс»

Модуль польового журналу є сучасний ідеальний помічник для керівників, що дозволяє вести облік і аналізувати всі витрати, є гарною підтримкою в організації виробництва, включає повну документацію по рослинництву, дозволяє аналізувати галузь і на її основі приймати рішення. Дозволяє планувати та аналізувати будь-які дії в полі, розраховувати баланс поживних речовин, автоматично складати технологічні карти, розраховувати використані ресурси – фінансові та людські, зберігати історію.

Система Soft.Farm із використанням мобільних додатків показує, на якому полі працює техніка, яку площу, скільки гектарів планується обробити, скільки фактично зроблено, а скільки залишилося виконати робіт, а також витрачено пального (рис. 1.3).

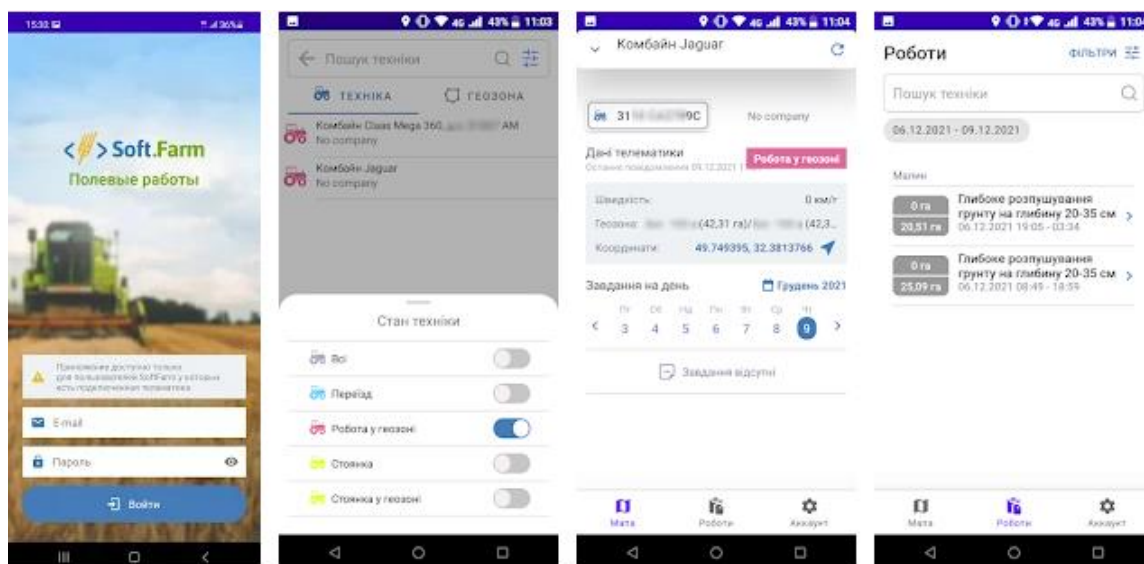


Рис. 1.3. Додаток «Полеві роботи» інтелектуальної інформаційної системи «Soft.Farm»

Використовуючи часові шкали у додатку «Полеві роботи» інтелектуальної інформаційної системи «Soft.Farm», ви можете переглядати хід операцій не лише в режимі онлайн, але й на певну дату чи зміну, а також за допомогою розширеного фільтра упорядковувати дані за термінами, географією, маркою технічного оснащення, операціями, відділами, змінами, статусами та керівниками.

Виконаний огляд існуючих інтелектуальних інформаційних систем для планування діяльності у сільськогосподарських підприємствах свідчить про те, що вони є вагомим інструментарієм для планування діяльності сільськогосподарських підприємств. Вони призначені для вирішення багатьох задач, які скеровані на планування діяльності у сільськогосподарських підприємствах. Однак, попри їх переваги з погляду планування використання технічного оснащення сільськогосподарського підприємства вони мають низку недоліків. Це зумовлює доцільність розроблення та проектування інтелектуальної інформаційної системи для планування використання технічного оснащення сільськогосподарського підприємства.

1.3. Ідентифікація ідеї проекту

Задля пришвидшення процесу розрахунку та уникнення помилок під час планування використання технічного оснащення сільськогосподарського підприємства була сформульована задача забезпечити автоматизацію процесу розрахунку використання технічного оснащення на підставі розроблення відповідної інтелектуальної інформаційної системи.

Коротко окреслимо ідею проекту створення інтелектуальної інформаційної системи. Вона забезпечує вирішення задач щодо планування використання технічного оснащення сільськогосподарського підприємства. Насамперед встановлюється комплекс робіт, що має загальний регламентований термін виконання. Також у сільськогосподарському підприємстві існує множина технічного оснащення, які можуть працювати паралельно, окремі із них є взаємозамінними і відрізняються між собою насамперед продуктивністю. Усе технічне оснащення можна ранжувати за їх продуктивністю, що дає можливість визначити тривалість виконання робіт заданого виду та обсягу. Для кожної марки технічного оснащення можна задати тривалість виконання окремих сільськогосподарських робіт.

Якщо припустити, що на кожен окрему добу відомо які види та обсяг робіт потрібно виконати, то приймаємо що всі роботи надходять одночасно в нульовий час, кожна робота виконується без зміни технічного оснащення до її завершення. Технічне оснащення можуть зупинятися із-за несприятливих виробничих чи природно-кліматичних умов у будь-який момент часу і почати працювати за сприятливих.

Пропонована інтелектуальна інформаційна система забезпечує формування графіка, який дозволить мінімізувати відхилення моментів часу відносно технологічно регламентованих термінів завершення виконання сільськогосподарських робіт.

Рішення цієї задачі здійснюється завдяки вирішенню трьох підзадач, які є частковими випадками виконання сільськогосподарських робіт залежно від кількості та виду технічного оснащення.

Перша підзадача розглядає систему виконання сільськогосподарських робіт за умови, що є лише одна одиниця технічного оснащення, друга – систему виконання сільськогосподарських робіт із використанням багатьох однакових видів технічного оснащення, третя – систему виконання сільськогосподарських робіт із кількома видами технічного оснащення різної продуктивності, або ж із виконанням різних видів робіт.

Основною перевагою запропонованої інтелектуальної інформаційної системи є те, що нею враховуються як виробничі та природно-кліматичні умови виконання робіт у будь-який момент часу, так і можливості окремого сільськогосподарського підприємства залучити доступне технічне оснащення одного виду або багатовидове та залучення їх до виконання різних видів робіт.

Пропонована інтелектуальна інформаційна система планування використання технічного оснащення сільськогосподарського підприємства вирішує одну із інформаційних та інженерно-прикладних задач, що є досить актуальною на даний час для різних за розмірами сільськогосподарських підприємств. Інтелектуальна інформаційна система забезпечить візуалізацію отриманих результатів у вигляді графіків виконання робіт для заданого технічного забезпечення та матиме інтерфейс користувача.

РОЗДІЛ 2.

МАТЕМАТИЧНИЙ ОПИС ЗАДАЧІ ТА ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ ПРОЕКТУ

2.1. Математичний опис задачі та вибір методу її розв'язання

Метою кваліфікаційної роботи є розробка інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства завдяки створенню програмного продукту із можливістю використання на ПК, що підкреслює практичну цінність цієї системи.

Вирішення задачі планування використання технічного оснащення сільськогосподарського підприємства здійснюється на підставі заданих початкових даних:

- множини виконання сільськогосподарських робіт – $J = \{j_1, j_2, \dots, j_n\}$, яких може бути для заданої доби n ;
- тривалості виконання сільськогосподарських робіт – $P = \{p_1, p_2, \dots, p_n\}$;
- кількості m технічного оснащення заданої марки за умови що вони доступні ($m \geq l$);
- продуктивності технічного оснащення: – $H = \{h_1, h_2, \dots, h_n\}$;
- регламентований термін виконання сільськогосподарських робіт – d .

Цільова функція вирішення задачі планування використання технічного оснащення сільськогосподарського підприємства стосується мінімізації відхилення моментів часу відносно технологічно регламентованих термінів завершення виконання сільськогосподарських робіт:

$$L \rightarrow \min. \quad (2.1)$$

Методи та алгоритми, описані у роботах [14] та [16], будуть використані для розв'язування нашої задачі, умови якої наведені вище. Вони частково будуть удосконалені для умов планування використання технічного оснащення

сільськогосподарського підприємства із кількома їх видами. У наукових працях доведено, що запропоновані їх авторами алгоритми створюють оптимальні графіки використання техніки і тому можуть використовуватися для вирішення нашої задачі.

Також нами пропонується алгоритм розподілу окремих сільськогосподарських робіт між технічним оснащенням. Цей алгоритм не гарантує найкращого розподілу, але, крім повного пошуку, не існує алгоритму, який би призначав для окремих видів технічного оснащення найкращого набору робіт за заданий проміжок часу їх виконання. Оскільки неможливо виконати повний пошук із дуже великою кількістю вхідних впливів, метод описаний у цьому окремому розділі, буде використаний для вирішення кількох задач, пов'язаних із ефективним використанням технічного оснащення.

2.2. Опис методу планування використання технічного оснащення

Насамперед розглянемо випадок використання одного технічного оснащення. Це досить ідеалізоване завдання для сільськогосподарських підприємств, але воно стане основою для всіх подальших сценаріїв використання технічного оснащення. Таким чином, у процесі розв'язування задачі з регламентованим терміном виконання робіт та з однією одиницею технічного оснащення можуть виникнути дві ситуації, залежно від регламентованого терміну (відносно нульового терміну, тобто моменту планування робіт):

- регламентований період виконання робіт далекий від початку координат (рис. 1.1);
- регламентований період виконання робіт близький до початку їх відліку (початку виконання) (рис. 1.2).

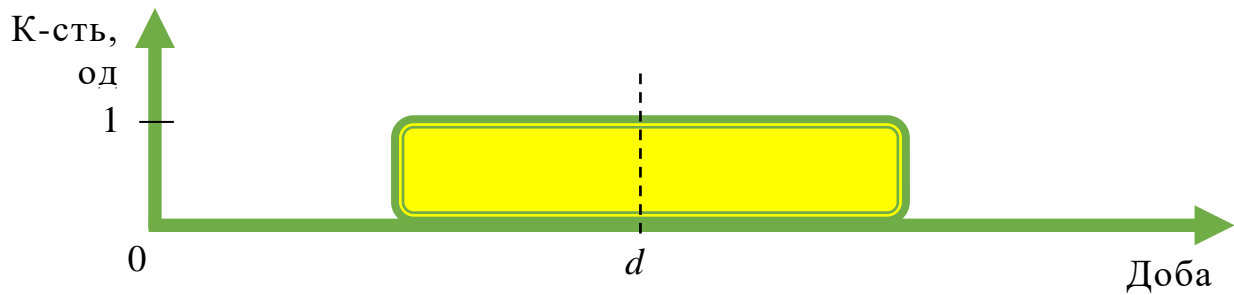


Рис. 2.1. Графічна інтерпретація регламентованого періоду виконання робіт за зміщення початку робіт відносно початку координат

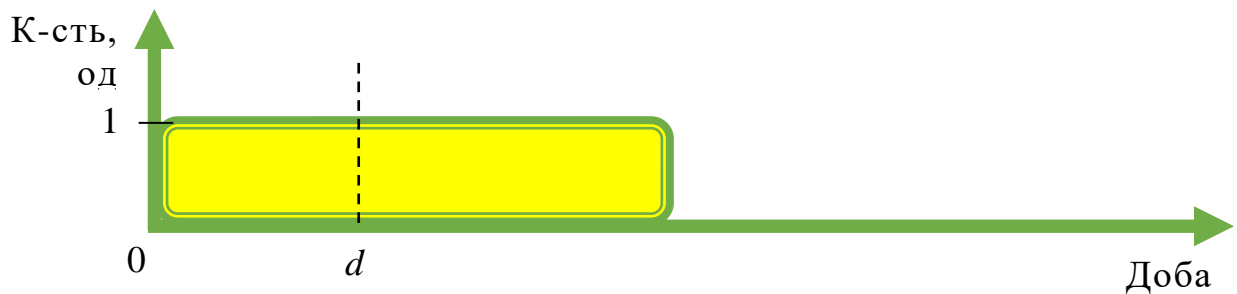


Рис. 2.2. Графічна інтерпретація регламентованого періоду виконання робіт за початку робіт близькому до початку координат

Коли ми говоримо «початок робіт близький до початку координат», це означає, що за умовами даного регламенту неможливо побудувати довільний графік використання технічного оснащення, які виконуватимуть будь-яку кількість завдань, що задовольняє регламенти.

На рисунках 2.1 та 2.2 показані умовні контури графіка – їх схематичні вигляди. Головне, на що варто звернути увагу на цих рисунках це розташування контурів відносно початку координат. Це визначає сценарій залучення технічного оснащення для виробничих умов, що розглядаються.

Наприклад, на рисунку 2.1 контур виконання робіт далекий від початку виконання робіт, тоді як на рисунку 2.2 контур виконання робіт близький до нього, і більша частина контуру знаходиться праворуч від регламентованого терміну. Це свідчить про те, що за першим алгоритмом неможливо

організувати роботу і, отже, виникають два сценарії – період регламентованого терміну досить малий, щоб побудувати довільний графік. У цьому випадку роботи починаються з моменту входу в систему.

Для кожного випадку розробляється свій алгоритм. Їх поєднання забезпечує складний алгоритм розв'язання поставленої задачі, повна схема якого показана на діаграмі у графічній інтерпретації виконання робіт (рис. 2.3).

Якщо у технічного оснащення різна продуктивність, до алгоритму додається ще один крок. Перш ніж розділити роботи, потрібно відсортувати технічне оснащення за зниженням продуктивності. Таким чином, найдовша робота на кожному етапі розподілу передається технічному оснащенню більшої продуктивності, врівноважуючи цим самим загальний контур побудованого графіка.

Розглянемо приклад, у якому задана множина виконання сільськогосподарських робіт $J = \{1, 2, 3, 3, 4, 5, 6, 7\}$. При цьому залучається технічне оснащення заданої марки у кількості $m = 3$ із продуктивністю $H = \{12; 10; 8\}$.

Початкові дані для проведення планування використання технічного оснащення сільськогосподарського підприємства подано у табл. 2.1.

Таблиця 2.1. Початкові дані для проведення планування використання технічного оснащення сільськогосподарського підприємства

Показник	Значення							
	1	2	3	3	2	1	1	2
Кількість технічного оснащення, од	1	2	3	3	2	1	1	2
Продуктивність технічного оснащення, га/зміну	12	10	8	8	10	12	12	10
Регламентована тривалість виконання робіт, діб	7	6	5	4	3	3	2	1
Тривалість використання технічного оснащення, діб	6,7	6	5	4,5	3	2,7	1,4	1

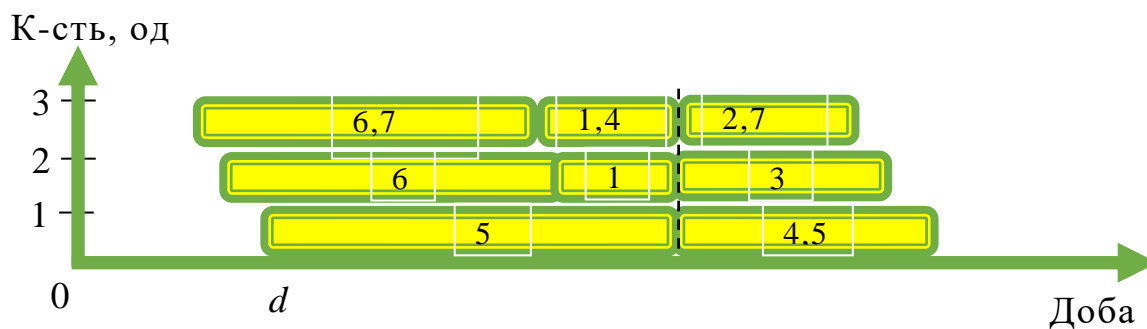


Рис. 2.3. Графічна інтерпретація прикладу побудови графіка виконання робіт після їх розподілу для технічного оснащення із різною продуктивністю

У цьому прикладі ви бачите, що порядок виконання робіт у графіку не змінився, хоча зовнішній вигляд самого графіка став більш компактним. І хоча графік, створений у цьому випадку, може бути неоптимальним, при збільшенні кількості робіт і неможливості виконати повний перебір, цей алгоритм дасть бажані результати для заданого проміжку часу.

2.3. Вибір засобів для виконання проекту

Для виконання проекту, зокрема завдань щодо створення програмного продукту інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства із віконним додатком для ПК, було здійснено вибір засобів для виконання проекту:

- Python 3.9;
- Tkinter;
- Matplotlib;
- Spyder4.

Для моделювання майбутньої інтелектуальної інформаційної системи використовувався Diagrams.net.

Python – це інтерпретована високорівнева об'єктно-орієнтована мова програмування зі суворою динамічною типізацією для загального використання [10].

Python широко використовується в багатьох областях, включаючи розробку програмного забезпечення. Низький поріг входу та простота роблять цю мову програмування основою будь-якого дослідження та експерименту. Тому обрали цю мову для написання коду програмного продукту. Її основними перевагами є:

- якісна та доступна документація;
- простота написання та чистота коду;
- велика спільнота розробників;
- динамізм;
- доступ до великої кількості бібліотек і фреймворків.

Python часто використовується для таких завдань, як моделювання або експериментування, оскільки його простота дозволяє швидко писати код для будь-якої системи та негайно тестувати її.

Python повністю підходить для проведення експериментів і тестування алгоритмів теорії планування.

Tkinter – це стандартний пакет Python, який містить усі необхідні інструменти для створення графічного інтерфейсу користувача [10]. Tkinter простий у використанні, але дуже потужний. Він забезпечує доступ до всіх необхідних функцій, які використовуються для створення графічного інтерфейсу користувача.

Matplotlib – графічна бібліотека мови Python. Найпопулярніша бібліотека для створення графіків та креслень, що забезпечує зручний інтерфейс для їх інтеграції в програмні продукти.

Spyder – це потужне наукове середовище, написане на Python, для Python, розроблене науковцями, інженерами даних та для них. Він має унікальну комбінацію передових функцій редагування, аналізу, налагодження та профілювання комплексного інструменту розробки з дослідженням даних,

інтерактивним виконанням, глибокою перевіркою та прекрасними можливостями візуалізації наукового пакету. Крім того, Spyder пропонує вбудовану інтеграцію з багато популярних наукових пакетів, включаючи NumPy, SciPy, Pandas, IPython, QtConsole, Matplotlib, SymPy та інші. Крім багатьох вбудованих функцій, Spyder можна ще більше розширити за допомогою плагінів сторонніх розробників.

Diagrams.net – це інтуїтивно зрозумілий і сучасний інструмент моделювання системи, який використовує стандарти UML. Саме з його допомогою були створені всі схеми, представлені в цій роботі.

РОЗДІЛ 3.

ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЛАНУВАННЯ ВИКОРИСТАННЯ ТЕХНІЧНОГО ОСНАЩЕННЯ СІЛЬСЬКОГОСПОДАРСЬКОГО ПІДПРИЄМСТВА

3.1. Побудова моделі варіантів використання інтелектуальної інформаційної системи

Для побудови моделі варіантів використання інтелектуальної інформаційної системи використовуємо UML, яка являє собою уніфіковану мову моделювання. UML підходить для широкого спектру розроблених програмних систем, для різних сфер застосування, типів організацій, рівнів компетенції, розмірів проекту. UML описує об'єкт за допомогою єдиного синтаксису, тому незалежно від того, де ви намалюєте діаграму, її правила будуть зрозумілі кожному, хто знає графічну мову.

Для побудови моделі варіантів використання інтелектуальної інформаційної системи нами використано безкоштовний сервіс Diagrams.net. Але закладений в нього функціонал значно розширює можливості сервісу. Насправді, використовуючи diagrams.net, можна змоделювати та уявити будь-який процес, який стосується проектування інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства. Величезний вибір попередньо встановлених піктограм і схематичних зображень сервісу Diagrams.net значно скорочує час, необхідний для створення діаграм.

Побудована нами структурна схема варіантів використання інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства подана на рис. 3.1.

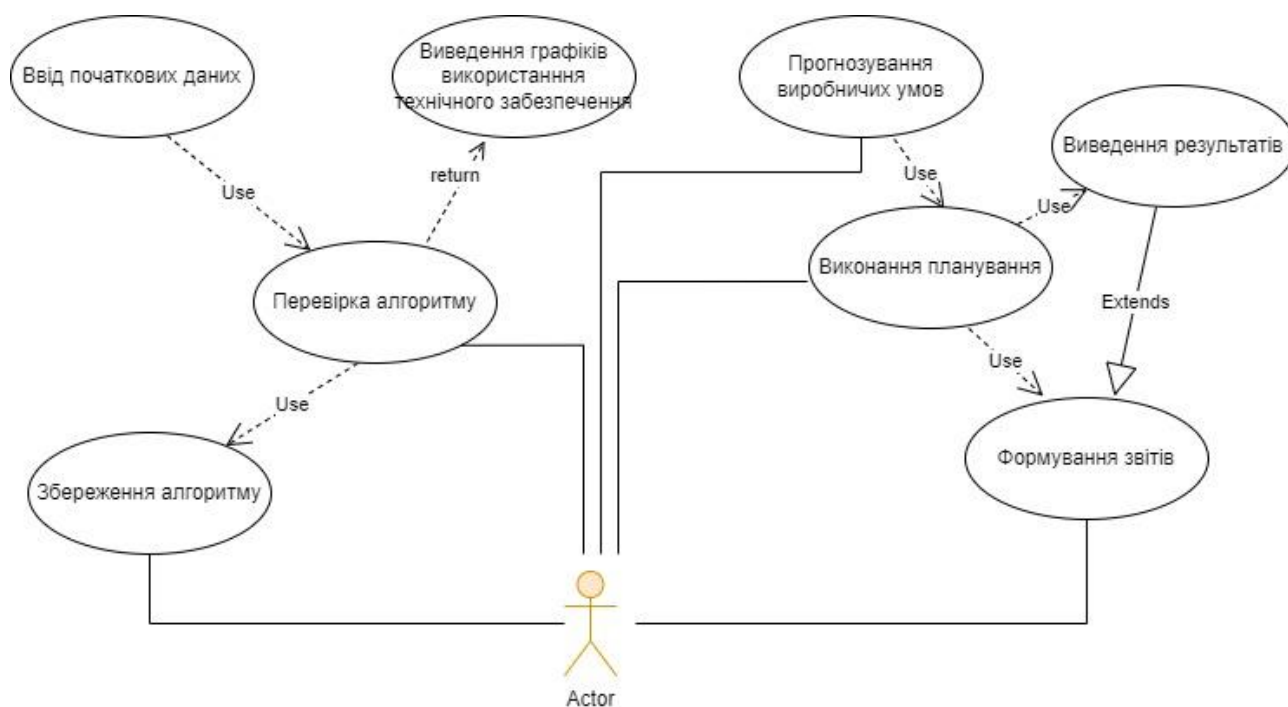


Рис. 3.1. Схема структурна варіантів використання інтелектуальної інформаційної системи

Для розробки моделі варіантів використання інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства було визначено суб'єкти:

Actor – користувач інтелектуальної інформаційної системи, який її використовує для потреб планування використання технічного оснащення сільськогосподарського підприємства;

Перевірка алгоритму – здійснюється для адекватного формування планів використання технічного оснащення в умовах сільськогосподарського підприємства;

Виконання планування – здійснюється на підставі обґрунтованого методу та алгоритмів для побудови графіків використання технічного оснащення в умовах сільськогосподарського підприємства;

Прогнозування виробничих умов – стосується початкових даних щодо регламентованих термінів виконання робіт, природно-кліматичних умов та обсягів робіт.

3.2. Побудова моделі класів інтелектуальної інформаційної системи

Програмний продукт інтелектуальної інформаційної системи розділений на дві складові, які представляють графічний інтерфейс користувача та функціональний блок.

Графічний інтерфейс користувача розроблений для зручності використання і представлений у вигляді вікон. Кожне вікно є окремим етапом виконання планування використання технічного оснащення в умовах сільськогосподарського підприємства або тестуванні алгоритмів.

Структурну схему класів інтелектуальної інформаційної системи подано на рис. 3.2.

Опис представлених класів інтелектуальної інформаційної системи, назви та їх описи наведено в таблиці 3.1.

Таблиця 3.1. Класи інтелектуальної інформаційної системи, назви та їх опис

Клас	Опис класів
<i>App</i>	Вхідна точка програми, яка містить посилання на інші класи
<i>MainPage</i>	Головне вікно програми, яке забезпечує подальший вибір дій користувача
<i>Test</i>	Вікно вибору файла із початковими даними з метою тестування алгоритмів
<i>ResearchOneDueDate</i>	Вікно введення характеристик генератора з метою перевірки алгоритму із одним регламентом на виконання робіт
<i>ResearchTwoDueDates</i>	Вікно введення характеристик генератора з метою перевірки алгоритму із двома регламентами на виконання робіт

Основний функціональний блок програми насамперед відповідає за виконання процесу тестування та планування використання технічного

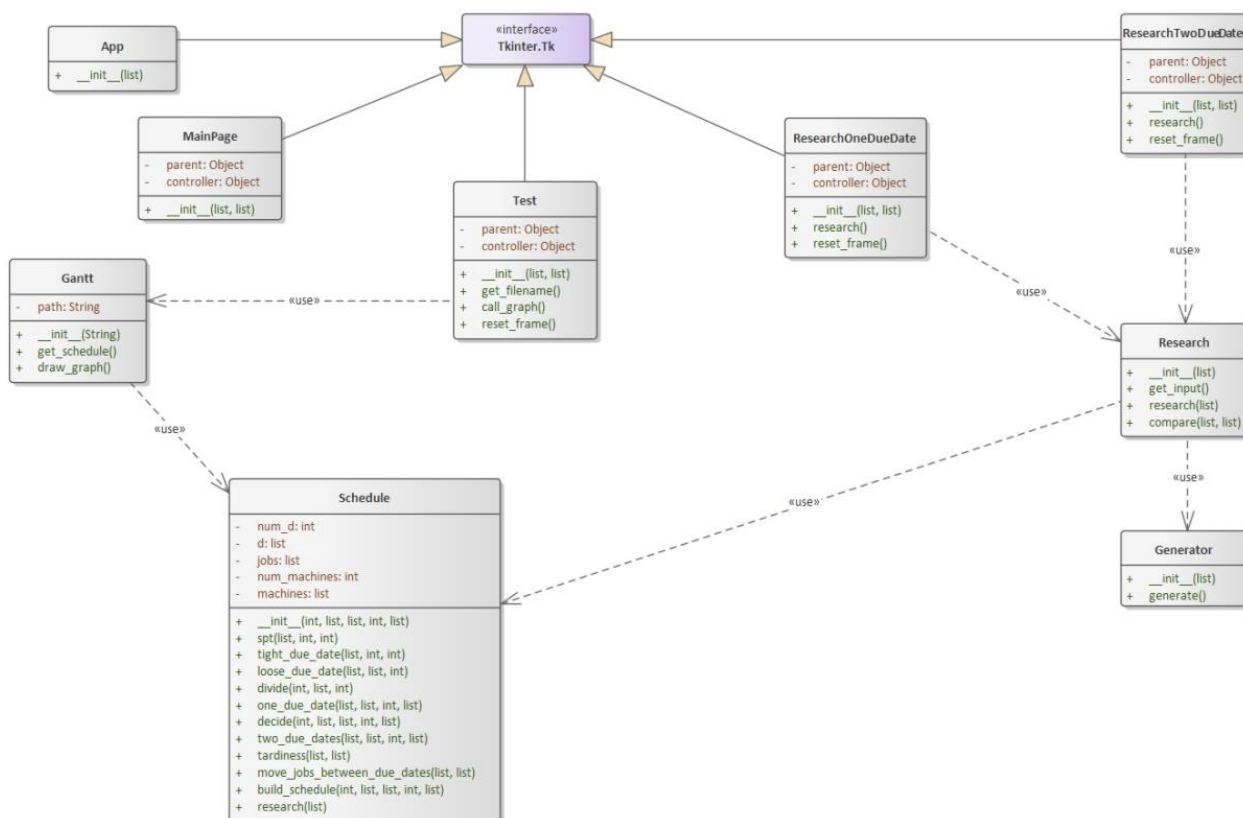


Рис. 3.2. Схема структурна класів інтелектуальної інформаційної системи

оснащення в умовах сільськогосподарського підприємства. Це основне ядро програми, яке описане класами, що представлено у таблиці 3.2.

Таблиця 3.2. Класи функціонального блоку програми інтелектуальної інформаційної системи та їх основні функції

Клас	Опис класів
<i>Schedule</i>	Забезпечує побудову графіків використання технічного забезпечення завдяки обґрунтованим алгоритмам
<i>Gantt</i>	Забезпечує візуалізацію побудованого графіка Ганта із використанням технічного забезпечення
<i>Generator</i>	Забезпечує генерацію вхідних даних
<i>Research</i>	Забезпечує планування за алгоритмами та виведення результату планування

Побудована схема структурна класів інтелектуальної інформаційної системи (рис. 3.2) відображає взаємозв'язки між вище описаними класами.

3.3. Побудова моделі послідовності інтелектуальної інформаційної системи

Як вже було сказано, програма інтелектуальної інформаційної системи може працювати в двох режимах – тестування і планування. Режим планування також поділяється на два незалежні процеси – планування виконання робіт технічним забезпеченням із одним або двома регламентами. Таким чином, користувач може виконати одне з наступного в один момент часу – тестування алгоритму, планування виконання робіт технічним забезпеченням із одним або двома регламентами. Кожен із двох процесів представлений окремою альтернативою у моделі послідовності інтелектуальної інформаційної системи на рис. 3.3.

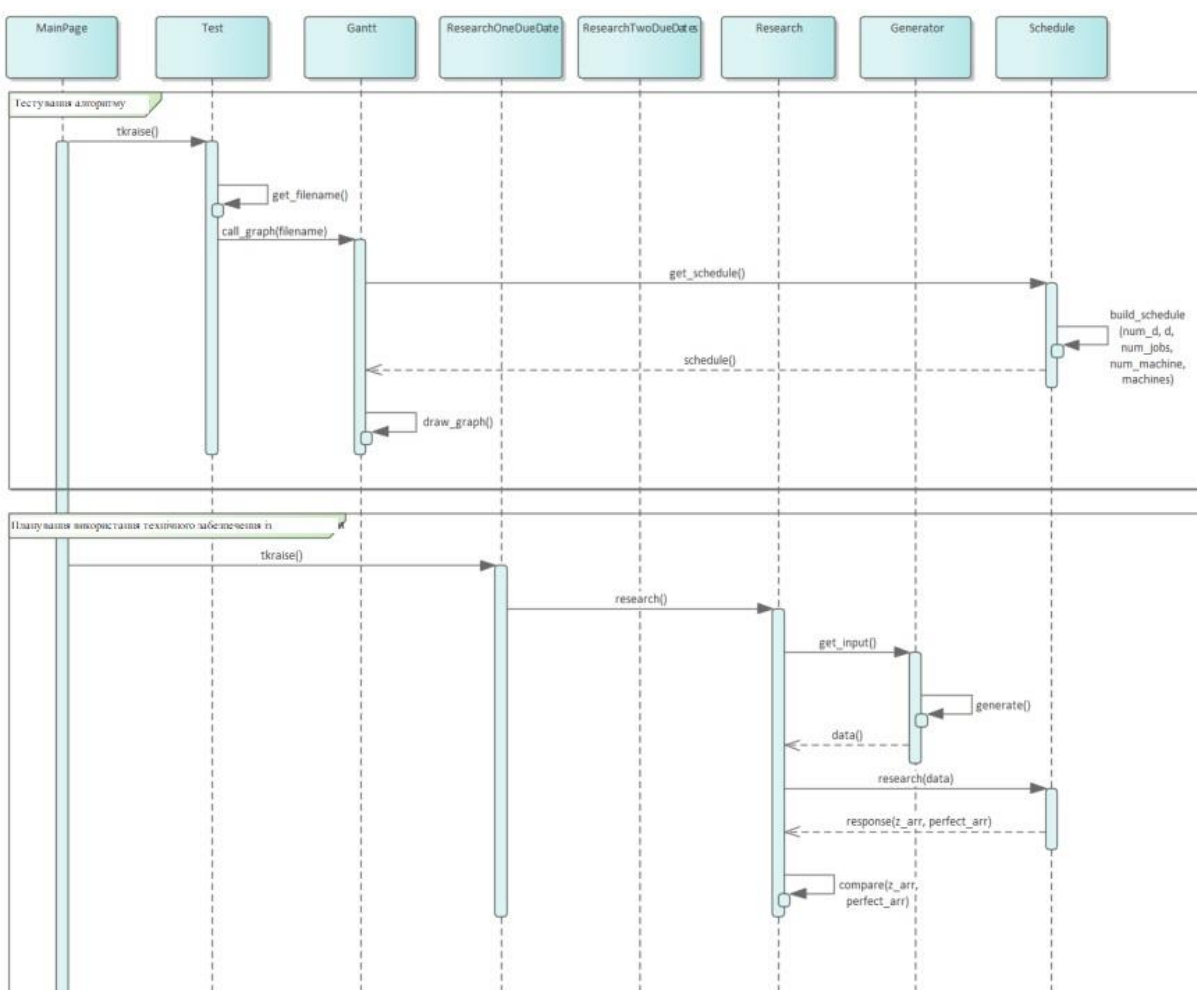


Рис. 3.3. Схема структурна послідовності інтелектуальної інформаційної системи

Схема структурна послідовності інтелектуальної інформаційної системи показує, як програмні об'єкти взаємодіють один з одним і послідовність викликів функцій у кожному процесі. Ось коротка послідовність дій, які виконує користувач у кожному з цих процесів. На початку користувач входить у перше вікно (представлене класом MainPage), де він може вибрати, що він хоче робити. Після цього послідовність дій відрізняється двома режимами (тоді як послідовність дій розв'язання задач планування для однієї та двох регламентів однакова – відрізняються лише вікна, які викликаються в кожному процесі).

У тестовому режимі користувач спочатку вибирає вхідний файл у вікні GUI, представленому класом Test, після чого будується графік Ганта для заданих умов, зчитує його та передає вхід до об'єкта Schedule, який відповідно до цього потребує вхідні дані. У подальшому користувач вибирає потрібний алгоритм і будує план використання технічного оснащення сільськогосподарського підприємства, а потім повертається до класу Gantt, який встановлює план і виводить його у вихідний файл.

У режимі планування користувач вводить початкові налаштування генератора або вибирає у вікні інтерфейсу опцію «повне планування», представлену класом ResearchOneDueDate або ResearchTwoDueDates (залежно від виду завдань планування з одним або двома регламентами). Налаштування параметрів генератора передаються до об'єкта класу Research, який спочатку створює унікальні вхідні набори даних за допомогою об'єкта генератора, а потім передає їх об'єкту Schedule, який встановлює графіки для цього входу, а також ідеальний (але неможливих) сценаріїв залучення технічного оснащення.

Випадки з однаковими вхідними даними (означає збільшення регламентованих умов для застосування до кожного набору алгоритмів даних, які гарантовано створюють оптимальні плани використання технічного забезпечення для заданих сільськогосподарських робіт. Отримані значення повертаються у об'єкт класу Research, який відображає значення цільових функцій побудованих графіків та їх нижні межі (для кожного набору вхідних

даних отримуємо два значення цільової функції – дійсне та ідеальне). Цей графік є результатом застосування алгоритмів для планування виконання робіт технічним забезпеченням.

3.4. Побудова моделі діяльності інтелектуальної інформаційної системи

Нами виконано побудову моделі діяльності інтелектуальної інформаційної системи, яка представлена на рис. 3.4.

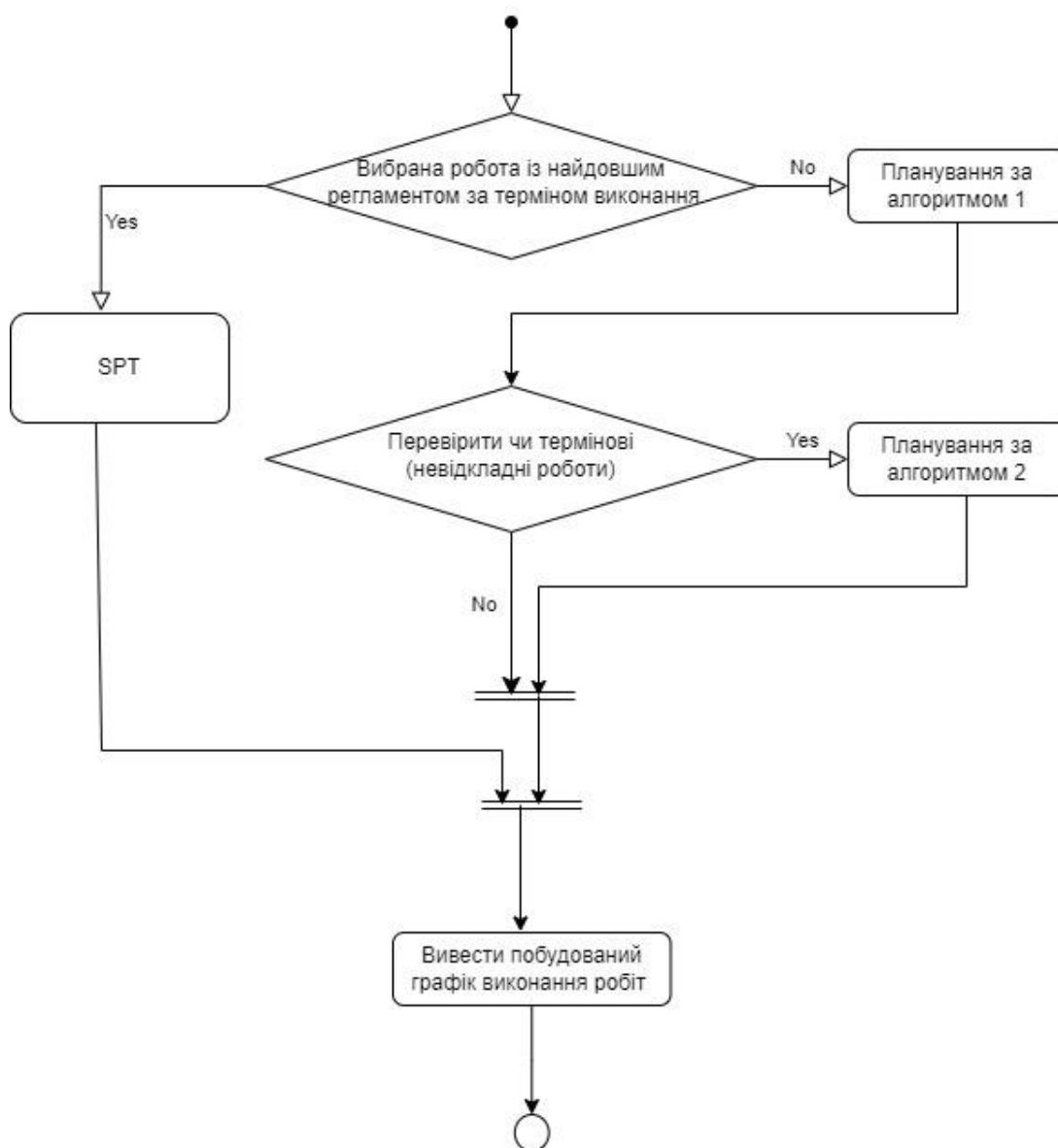


Рис. 3.4. Схема структурна діяльності інтелектуальної інформаційної системи

Як зазначено вище, програмний продукт інтелектуальної інформаційної системи може працювати в двох режимах. І відповідно результати роботи програми у кожного є різні.

При тестуванні алгоритму результатом роботи є графік використання технічного оснащення сільськогосподарського підприємства, а також його візуалізація – діаграма Ганта.

При плануванні результатом є графічна форма відображення, яка порівнює значення цільових функцій графіків, побудованих для згенерованих вхідних даних, та їх нижніх меж (збільшення регламентованих термінів виконання робіт) для заданого технічного забезпечення.

РОЗДІЛ 4.

РОЗРОБКА КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПЛАНУВАННЯ ВИКОРИСТАННЯ ТЕХНІЧНОГО ОСНАЩЕННЯ ФЕРМЕРСЬКОГО ГОСПОДАРСТВА

4.1. Створення вікна користувача інтелектуальної інформаційної системи

Насамперед перед розробкою інтелектуальної інформаційної системи планування використання технічного оснащення для заданих умов сільськогосподарського підприємства використовуючи мову програмування Python, слід розробити прототип пропонованого діалогового вікна для користувача. Він повинен вміщувати усі необхідні блоки, які лежать в основі формування відповідних програмних функцій. Запропонований нами прототип діалогового вікна для користувача відображено нижче (рис. 4.1), де подано розміщення відповідних блоків.

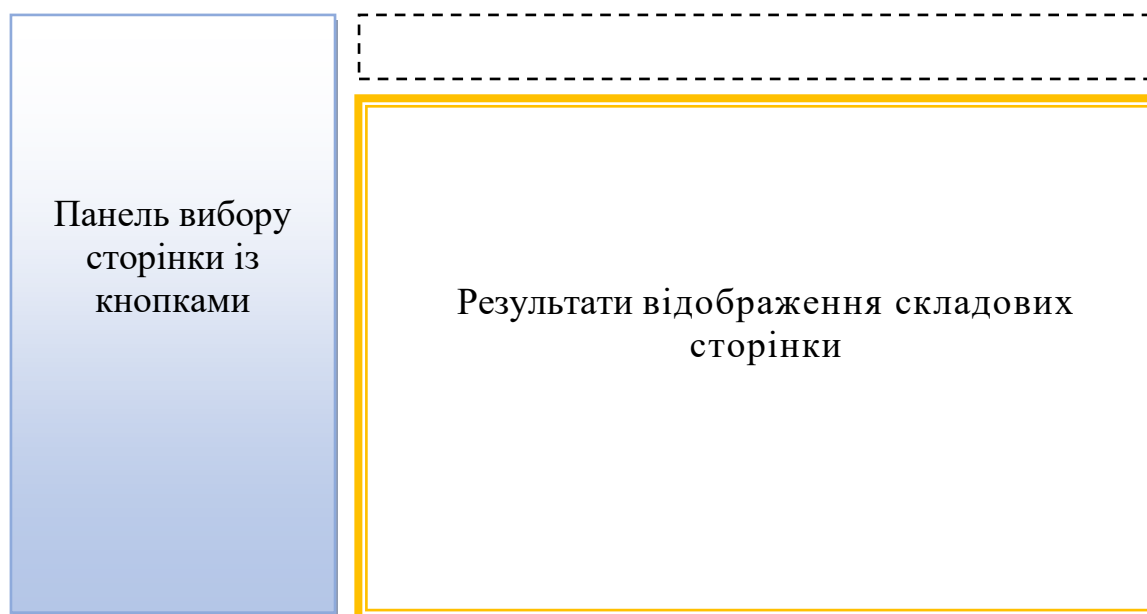


Рис. 4.1. Діалогове вікно для користувача інтелектуальної інформаційної системи планування використання технічного оснащення для заданих умов сільськогосподарського підприємства

На панелі вибору сторінки із окремими кнопками пропонується користувачам здійснити вибір потрібної сторінки. При цьому кнопки розташовано у порядку виконання планування використання технічного оснащення для заданих умов сільськогосподарського підприємства. Зокрема, пропонуються кнопки «Завантажити початкові дані», «Тестування», «Планування із 1 регламентом», «Планування із 2 регламентами» та «Зберегти результати планування як ...» (рис. 4.2).

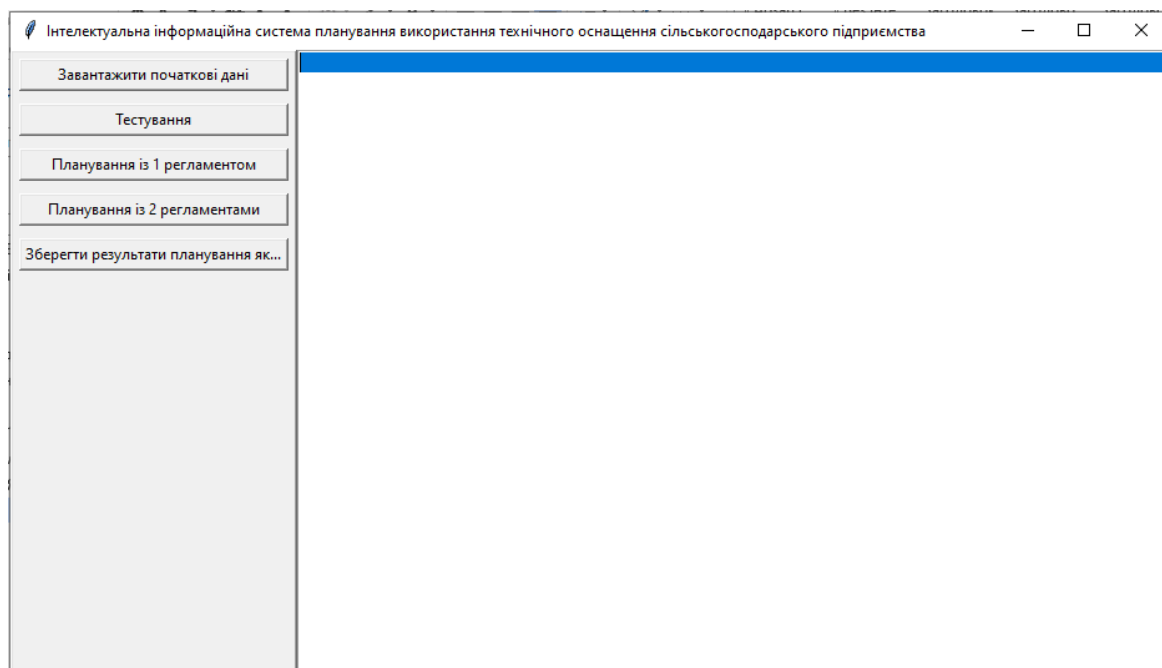


Рис. 4.2. Панелі вибору сторінок із відображенням кнопок

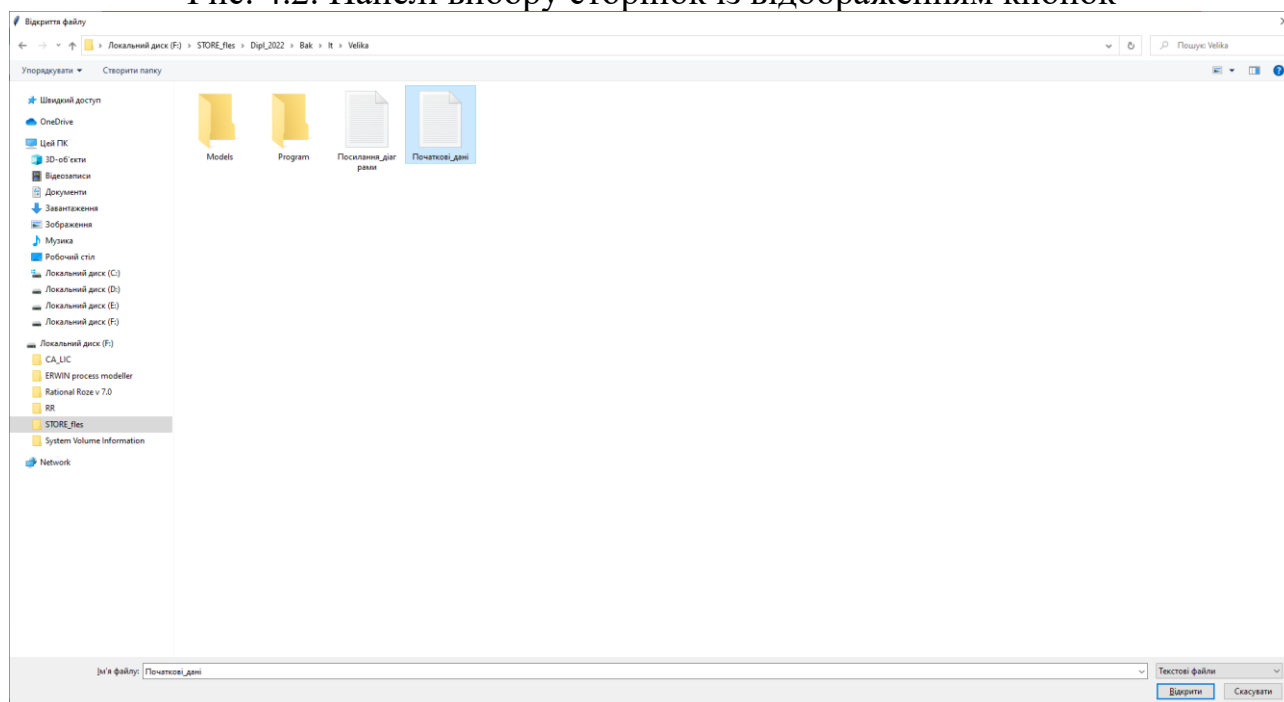


Рис. 4.3. Панель вибору файла із початковими даними

Сторінка «Завантажити початкові дані» забезпечує вибір файла із початковими даними (рис. 4.3).

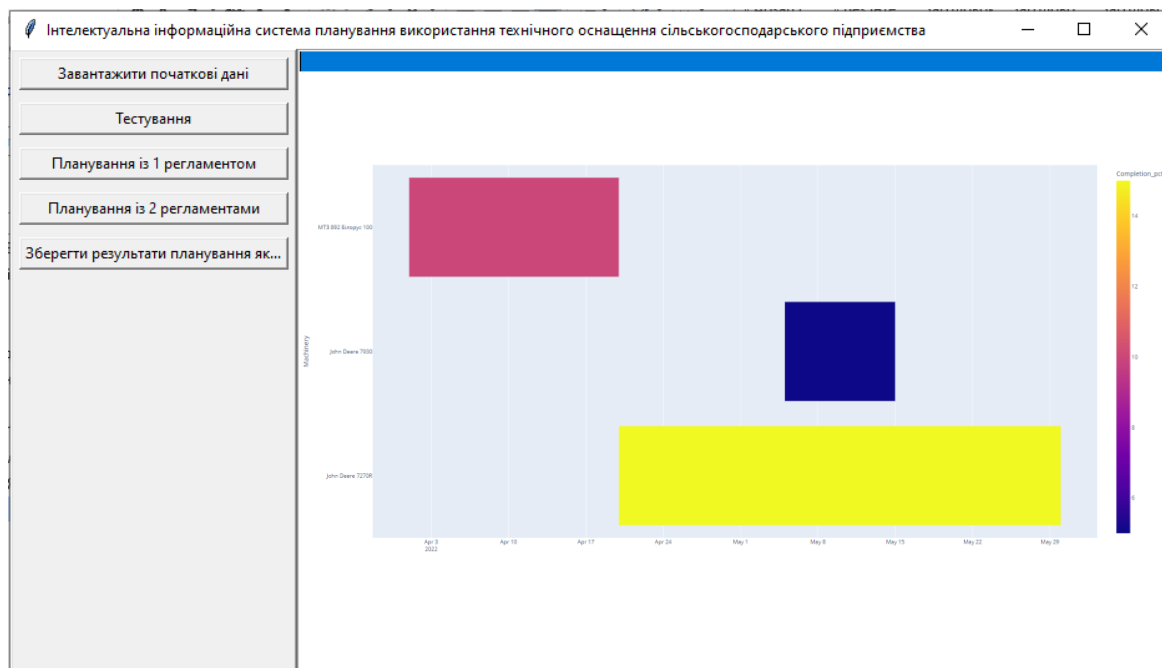


Рис. 4.4. Панель результатів планування використання технічного оснащення для заданих умов сільськогосподарського підприємства

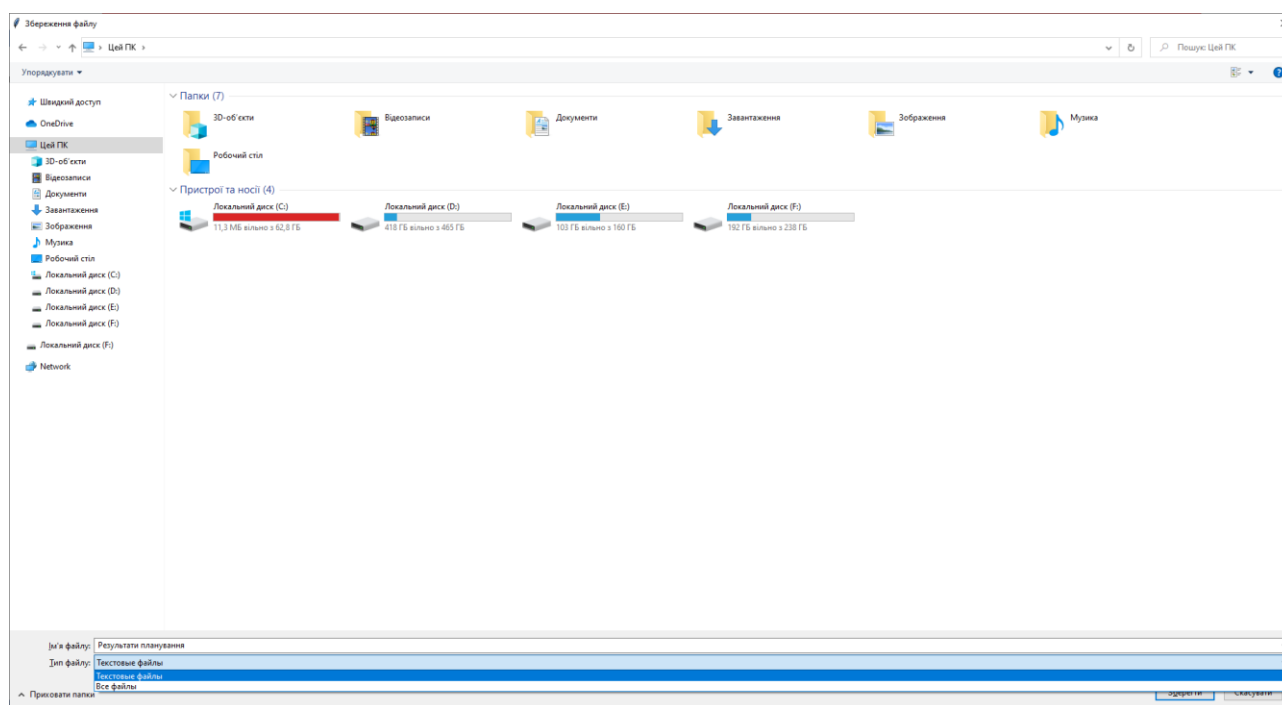


Рис. 4.5. Панель із збереженням у результатів планування

Наступна сторінка «Планування із 2 регламентами» відображена у вигляді графіка Ганта із результатами планування, що забезпечує візуальне представлення результатів планування та розрахунків щодо використання

технічного оснащення для заданих умов сільськогосподарського підприємства (рис. 4.4).

Остання сторінка «Зберегти результати планування як ...» відображає панель із збереженням у результатів планування (рис. 4.5). У цьому файлі подано результати планування використання технічного оснащення для заданих умов сільськогосподарського підприємства.

4.2. Створення модуля для формування діалогового вікна користувача інтелектуальної інформаційної системи

Переважно бібліотеку Tkinter використовують для створення інтерфейсу користувача за допомогою коду мови Python. Це досить зручно для програмних додатків Windows, тому її ми вибрали для нашого проекту. Завдяки цій бібліотеці створено діалогове вікно для користувача інтелектуальної інформаційної системи планування використання технічного оснащення для заданих умов сільськогосподарського підприємства, що представлено на рис. 4.2-4.5.

Нами створено блокнот у Tkinter із використанням віджету Frame, який забезпечує групування та керування іншими наявними віджетами (рис. 4.6). Він виконує роль контейнера, відповідальний за організацію розташування решти віджетів. Саме це забезпечує відображення прямокутних областей на екранах користувачів інтелектуальної інформаційної системи для налаштування макету і заповнення зазначених віджетів. Frame також використовують як базові класи для складних віджетів.

```

window = tk.Tk()
window.title("Інтелектуальна інформаційна система планування використання технічного оснащення сільськогосподарського підприємства")
window.rowconfigure(0, minsize=500, weight=1)
window.columnconfigure(1, minsize=700, weight=1)

txt_edit = tk.Text(window)
fr_buttons = tk.Frame(window, relief=tk.RAISED, bd=2)
btn_open = tk.Button(fr_buttons, text="Завантажити початкові дані", command=open_file)
btn_test = tk.Button(fr_buttons, text="Тестування", command=open_file)
btn_plan_1 = tk.Button(fr_buttons, text="Планування із 1 регламентом", command=open_file)
btn_plan_2 = tk.Button(fr_buttons, text="Планування із 2 регламентами", command=open_file)
btn_save = tk.Button(fr_buttons, text="Зберегти результати планування як...", command=save_file)

btn_open.grid(row=0, column=0, sticky="ew", padx=5, pady=5)
btn_test.grid(row=1, column=0, sticky="ew", padx=5, pady=5)
btn_plan_1.grid(row=2, column=0, sticky="ew", padx=5, pady=5)
btn_plan_2.grid(row=3, column=0, sticky="ew", padx=5, pady=5)
btn_save.grid(row=4, column=0, sticky="ew", padx=5, pady=5)

fr_buttons.grid(row=0, column=0, sticky="ns")
txt_edit.grid(row=0, column=1, sticky="nsew")

```

Рис. 4.6. Програмний код для виконання групвань та керування віджетами

Представлений вище макет сторінки «Завантажити початкові дані» забезпечує вибір файла із початковими даними на підставі програмного коду, який подано на рис. 4.7.

```

def open_file():
    """Відкриваємо файл із початковими даними"""
    filepath = askopenfilename(
        filetypes=[("Текстові файли", "*.txt"), ("Всі файли", "*.*")]
    )
    if not filepath:
        return
    txt_edit.delete("1.0", tk.END)
    with open(filepath, "r") as input_file:
        text = input_file.read()
        txt_edit.insert(tk.END, text)
    window.title(f"Інтелектуальна інформаційна система планування
        використання технічного оснащення сільськогосподарського
        підприємства - {filepath}")

```

Рис. 4.7. Програмний код функції open_file

Результати планування у діалоговому вікні користувача інтелектуальної інформаційної системи записують у файл із кодом формату .txt. Для цього написано функцію під назвою save_file (рис. 4.8).

```

def save_file():
    """Зберігаємо поточний файл як новий файл."""
    filepath = asksaveasfilename(
        defaulttextextension="txt",
        filetypes=[("Текстові файли", "*.txt"), ("Всі файли", "*.*")],
    )
    if not filepath:
        return
    with open(filepath, "w") as output_file:
        text = txt_edit.get("1.0", tk.END)
        output_file.write(text)
    window.title(f"Результати планування використання технічного
    оснащення сільськогосподарського підприємства - {filepath}")

```

Рис. 4.8. Програмний код функції save_file

4.3. Створення модуля планування використання технічного оснащення та виведення результатів

Створення модуля планування використання технічного оснащення та виведення результатів виконували завдяки прописуванню коду класу *ResearchOneDueDate*, який забезпечує введення характеристик генератора з метою перевірки алгоритму із одним регламентом на виконання робіт (рис. 4.9).

```

class ResearchOneDueDate(tkinter.Frame):
    def __init__(self, parent, controller):
        tkinter.Frame.__init__(self, parent)
        self.controller = controller

        arr = ['S', 'M', 'L']

        self.button = tkinter.Button(self, text='Назад', font=font,
                                     bg=light_grey_color, fg=dark_grey_color,
                                     command=lambda: self.reset_frame()).grid(row=0,
                                     columnspan=2, sticky='W')
        tkinter.Label(self, text='Параметри генератора',
                      font=font_header).grid(row=1, columnspan=2, padx=275, pady=50)

        tkinter.Label(self, text='Кількість виконуваних робіт', font=font).grid(row=2, column=0, padx=20, pady=10, sticky='ew')

        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=2, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Тривалість регламентованого терміну', font=font).grid(row=3, column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=3, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Обсяг робіт', font=font).grid(row=4, column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=4, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Кількість технічного оснащення', font=font).grid(row=5, column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=5, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Продуктивності технічного оснащення', font=font).grid(row=6, column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=6, column=1, padx=10, pady=10, sticky='w')

```

Рис. 4.9. Програмний код класу *ResearchOneDueDate*

Наступний етап написання програмного коду виконується для створення класу *Test*, який забезпечує вибір файлу із початковими даними з метою тестування алгоритмів (рис. 4.10).

```
class Test(tkinter.Frame):
    def __init__(self, parent, controller):
        tkinter.Frame.__init__(self, parent)
        self.controller = controller

        self.button = tkinter.Button(self, text='Назад', font=font, bg=light_grey_color, fg=dark_grey_color,
                                     command=lambda: self.reset_frame()).place(x=0, y=0)
        tkinter.Button(self, text='Оберіть файл з вхідними даними', font=font, bg=blue_color, fg='white',
                       command=lambda: self.get_filename()).pack(pady=25)

        self.literal = tkinter.Label(self, text='Обраний файл', font=font)
        self.text = tkinter.StringVar()
        self.text.set('')
        self.label = tkinter.Label(self, textvariable=self.text, font=font,
                                   borderwidth=3, relief="groove", pady=20)
        self.button = tkinter.Button(self, text='Продовжити', font=font,
                                     bg=blue_color, fg='white',
                                     command=lambda: self.call_graph())

    def get_filename(self):
        filename = askopenfilename()

        self.literal.pack(pady=10)
        self.label.pack()
        self.text.set(filename)
        self.button.pack(pady=20)

    def reset_frame(self):
        from MainPage import MainPage
        self.controller.frames[MainPage].tkraise()

    def call_graph(self):
        gantt = Gantt(self.text.get())
        gantt.draw_graph()
```

Рис. 4.10. Програмний код класу *Test*

Наступний етап передбачає написання коду для проведених математичних розрахунків використовуючи модель, що подана у розділі 3 цієї роботи та побудови графіка Ганта. Написання програмного коду виконується для створення класу *Research*, який забезпечує планування за алгоритмами та виведення результату планування.

Написання програмного коду виконується для створення класу *Schedule*, який забезпечує побудову графіків використання технічного забезпечення завдяки обґрунтованим алгоритмам (рис. 4.11).

```

from copy import deepcopy
class Schedule:
    def __init__(self):
        pass

    def spt(self, jobs, machine=1, start=0):
        jobs = [job/machine for job in sorted(jobs)]
        schedule = []

        for job in jobs:
            schedule.append([start, job])
            start += job

        return schedule

    def tight_due_date(self, jobs, d, machine=1, start=0):
        jobs = [job/machine for job in sorted(jobs, reverse=True)]
        schedule_before_due_date = []
        schedule_after_due_date = []
        tay1 = d
        tay2 = sum(jobs) - d

        for job in jobs:
            if tay1 >= tay2:
                schedule_before_due_date.append([start, job])
                start += job
                tay1 -= job
            else:
                schedule_after_due_date.insert(0, [0, job]) # setting temporary time of start to 0
                tay2 -= job

        # setting times when the job is starting
        for job in schedule_after_due_date:
            job[0] = start
            start += job[1]
        return schedule_before_due_date + schedule_after_due_date

    def loose_due_date(self, jobs, d, machine=1):
        jobs = [job/machine for job in sorted(jobs, reverse=True)]
        schedule_before_due_date = [[0, job] for job in jobs[::-2]]
        schedule_after_due_date = [[0, job] for job in jobs[1::2][::-1]] #[::-1] is used to reverse the list

        duration = sum([job[1] for job in schedule_before_due_date]) # sum of all jobs that go before due date
        for job in schedule_before_due_date:
            job[0] = d - duration
            duration -= job[1]

        # at this point duration must be equal to 0
        duration = d

```

Рис. 4.11. Програмний код класу *Schedule*

Написання програмного коду виконується для створення класу *Gantt*, який забезпечує візуалізацію побудованого графіка Ганта із використанням технічного забезпечення (рис. 4.12).

```

class Gantt:
    def __init__(self, path):
        self.filename = path

        with open(self.filename, 'r') as f:
            self.num_d = int(f.readline())
            self.num_jobs = []
            self.jobs = []
            self.d = []

            for _ in range(self.num_d):
                self.num_jobs.append(int(f.readline()))
                self.jobs.append([int(job) for job in f.readline().split()])
                self.d.append(int(f.readline()))

            self.num_machines = int(f.readline())
            self.machines = [int(machine) for machine in f.readline().split()]

    def get_schedule(self):
        return Schedule().build_schedule(self.num_d, self.d, self.jobs, self.num_machines, self.machines)

    def draw_graph(self):
        schedule = self.get_schedule()

        fig, gnt = plt.subplots()
        gnt.set_title('Побудований план використання технічного оснащення')
        gnt.set_ylim(0, (self.num_machines + 1) * 10 + 10)
        gnt.set_xlim(0, max([job[0] + job[1] for machine in schedule for due_date in machine for job in due_date]) + 5)
        gnt.set_xlabel('Час')
        gnt.set_ylabel('Технічне оснащення')
        gnt.grid(True)

```

Рис. 4.12. Програмний код класу *Gantt*

Побудований графік Ганта, який є складовою плану використання технічного оснащення сільськогосподарського підприємства виводиться у вікно результатів, як показано на рис. 4.4.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

Нами розглянуто питання із безпеки та охорони праці програміста, який працює над розробленням інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства. У приміщенні де працює програміст використовується серверне обладнання та комп'ютерна техніка.

Метою розділу є аналіз впливу шкідливих чинників і умов праці, а також описані норми та заходи, які будуть направлені на усунення потенційно небезпечних виробничих чинників, які можуть негативно вплинути на організм програміста в процесі розробки інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства.

5.1. Аналіз стану шуму у приміщенні

Характеристика джерел шуму у приміщенні наведена в таблиці 5.1.

Таблиця 5.1 – Характеристика джерел шуму

№ п/п	Найменування	Джерело небезпеки	Причини небезпек	Наслідки небезпеки
1	Сервер Dell PowerEdge T330	Елемент охолодження сервера (куллер)	Шум елемента охолодження	Емоційна роздратованість працівника, що може призвести до зменшення продуктивності його роботи

Реальні та нормативні значення рівня шуму в приміщенні наведені в таблиці 5.2.

Таблиця 5.2 – Реальні та нормативні значення рівня шуму

№ п/п	Назва параметра	Реальне значення	Нормативне значення
1	Рівень шуму, дБ	56-60	50

Засоби та заходи захисту від впливу шуму наведено у таблиці 5.3.

Таблиця 5.3 – Засоби та заходи захисту від небезпеки

№ п/п	Вид заходу	Зміст заходу	Результат заходу
1	Технічні	підтримка серверної шафи у закритому вигляді; встановлення шумоізоляційної перегородки між серверною та робочою зоною; додаткова шумоізоляція серверної шафи за допомогою ущільнювачів дверей і швів та резинових підставок під ноги шафи.	Мінімізування рівня шуму
2	Організаційні	перерви для працівника; перевірка на належний стан куллерів та шумоізоляційних матеріалів перегородки і серверної шафи.	Запобігання впливу на організм працівника шуму
3	Експлуатаційні	своєчасний ремонт куллерів та заміна шумоізоляційних елементів	Захист від високого рівня шуму

5.2. Заходи щодо електробезпеки

Приміщення програміста, який працює над розробленням інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства, не відноситься до приміщень з підвищеною небезпекою. Обладнання не завдає великого навантаження на мережу. Джерела небезпеки наведено у табл. 5.4.

Таблиця 5.4 – Джерела електробезпеки

№ п/п	Назва	Джерело небезпеки	Причини небезпеки	Наслідки небезпеки
1	Персональний комп'ютер Zevs PC i7 2600	Блок живлення	Пошкодження блоку живлення, кабеля живлення.	Ураження струмом
2	Сервер Dell PowerEdge T330	Блок живлення, деталі сервера, що знаходяться під напругою	Пошкодження блоку живлення, кабеля живлення.	Ураження струмом
3	Роутер Asus RT-N11P	Блок живлення	Пошкодження блоку живлення, кабеля живлення	Ураження струмом
4	Джерело безперебійного живлення LogicPower LPT-W-10000RD	Блок вхідного живлення, деталі, що знаходяться під напругою	Пошкодження блоку живлення, вхідного кабеля живлення.	Ураження струмом

Дані про споживання напруги наведено у таблиці 5.5.

Таблиця 5.5 – Реальні та нормативні фактори небезпеки

№ п/п	Чинник небезпеки	Реальне значення	Нормативні значення
1	Максимальний струм	>1 А	50 мА
		220 В	45 В

Для зниження ймовірності настання небезпечної ситуації, необхідно дотримуватись заходів безпеки, які наведені в таблиці 5.6.

Таблиця 5.6 – Засоби захисту від електротравм

№ п/п	Група номенклатурних заходів з ОП	Вид заходу	Результат заходу
1	Технічні	<p>Використання ізоляційних матеріалів для сервера та серверної шафи (ізоляційні лаки для усунення витоків струсу).</p> <p>Захисні заземлення (наземні комунікації).</p> <p>Захисне розділення електромереж (використання розділювального трансформатора).</p> <p>Увімкнення техніки в мережу через заземлені фільтри (мережевий фільтр Defender DFS 151).</p> <p>Своєчасна заміна елементів, що працюють під напругою, які вийшли з ладу</p>	Уникнення пробою, витоків струму уникнення контакту зі струмопровідним и частинами
2	Організаційні	<p>Інструктаж з правил електробезпеки.</p> <p>Підтримка сухого, незапиленого приміщення з вологістю не вище 75%</p>	Доступність знань щодо безпеки експлуатації
3	Режимні	Перевірка несправностей техніки та мережі тільки у відключеному стані;	Уникнення контакту з елементами під напругою
4	Експлуатаційні заходи	Своєчасна заміна будь-яких пошкоджених елементів	Забезпечення безпечної роботи з об'єктом

У приміщенні програміста, який працює над розробленням інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства, виявлена наявність електробезпеки, яка може проявлятися у вигляді витoku струму з електромережі та надання травм організму при контакті з джерелом небезпеки.

5.3. Заходи щодо покращення пожежної безпеки

Характеристика джерел небезпеки, які наявні у кабінеті особи, яка приймає управлінські рішення, наведено в таблиці 5.7.

Таблиця 5.7 – Джерела пожежної небезпеки

№ п/п	Найменування	Джерело небезпеки	Причини небезпек	Наслідки небезпеки
1	Персональний комп'ютер Zevs PC i7 2600	Блок живлення, деталі під напругою	Коротке замикання	Виникнення пожежі
2	Сервер Dell PowerEdge T330	Блок живлення, деталі під напругою	Коротке замикання	
3	Роутер Asus RT-N11P	Блок живлення	Коротке замикання	
4	Джерело безперебійного живлення LogicPower LPT-W-10000RD	Блок вхідного живлення, деталі під напругою	Коротке замикання	
5	Матеріали і речовини, що схильні до займання	Загорання матеріалів	Зовнішнє загорання	
6	Щільність проводки	Оплавлення ізоляції	Коротке замикання	

Характеристика вибухонебезпечності та пожежної небезпеки у кабінеті особи, яка приймає управлінські рішення, наведено у таблиці 5.8.

Таблиця 5.8 – Характеристика вибухонебезпечності та пожежної небезпеки

№ п/п	Назва	Значення
1	Клас пожежі	А,Е
2	Клас зони приміщення по пожежній безпеці	Клас II-IIIa
3	Категорія пожежної небезпеки	В

Засоби та заходи захисту від вибухонебезпечності та пожежі у кабінеті особи, яка приймає управлінські рішення, наведено у таблиці 5.9.

Таблиця 5.9 – Засоби та заходи захисту від пожежної небезпеки

№ п/п	Заходи	Реалізація	Результат заходу
1	Технічні	Розташування у кімнаті порошкового вогнегасника ОПУ-10. Встановлення протипожежної теплового сповіщувача FT-B	Гасіння первинних осередків займання
2	Організаційні	Проведення навчань, інструктажів з пожежної безпеки. Створення планів евакуації.	Навчання з питань безпеки при пожежі, запобігання людських жертв
3	ЗІЗ	Протигази, респіратори та маски, захисний одяг.	Запобігання отруєнню та опіків
4	Експлуатаційні	Своєчасний ремонт та заміна обладнання.	Захист від пожеж, що можуть бути викликані технічними несправностями
5	Режимні	Встановлення пропускового режиму для серверної	Захист від зовнішніх осередків займання

Для забезпечення пожежної безпеки у кабінеті особи, яка приймає управлінські рішення, присутній порошковий вогнегасник ОПУ-10, та протипожежний тепловий сповіщувач FT-B.

5.4. Інструкція з охорони праці під час розроблення інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства

5.4.1. Загальні положення

Дія Інструкції з охорони праці під час розроблення інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства (далі - Інструкція) поширюється на всі структурні підрозділи підприємства.

По даній Інструкції працюючий перед початком роботи повинен пройти інструктаж. Результати інструктажу заносяться в «Журнал реєстрації інструктажів з питань охорони праці на робочому місці», в журналі після проходження інструктажу повинен бути підпис особи, яка інструктує та працівника, а також відмітка у стовпчику 12 про допуск до роботи.

Комп'ютери повинні мати передбачений заводом - виробником захист від ураження електричним струмом. Комп'ютер встановлюють на відстані не менше 1 м від стін, між собою на відстані не менше 1,5 м. Виключають можливість прямого засвічування екрану джерелом природного освітлення. Поверхня екрану повинна знаходитись на відстані 400-700 мм, від очей користувача. Висота робочої поверхні столу повинна становити 680-800 мм, ширина не менше 500 мм. Стілець повинен мати висоту 280-320 мм, ширину не менше як 380мм.

Працюючим на комп'ютерах забороняється:

- ✓ торкатися обірваних та оголених електричних проводів;

- ✓ користуватися пошкодженими розетками та вилками;
- ✓ працювати при знятому кожусі на моніторі чи системному блоці;
- ✓ працювати на комп'ютерах, монітори яких розташовані один проти одного в межах кімнати (приміщення).

Під час роботи, а також отриманою інформацією рекомендується працювати на світлому (білому) фоні з чорними знаками.

При роботі з комп'ютером можуть виникнути небезпечні та шкідливі виробничі фактори: електростатичне поле; недостатнє освітлення; психоемоційна напруга при тривалій роботі з екраном відеомонітору.

Працюючі за комп'ютером повинні дотримуватися наступного режиму праці:

- ✓ При введенні даних, читанні інформації з екрану безперервна тривалість роботи не повинна перевищувати 4-х годин при 8-годинному робочому дні;

- ✓ Через кожну годину роботи необхідно робити перерву на 5-10 хвилин, а через 2 години на 15 хвилин;

Перерви використовувати для зорового та фізичного розвантажування:

- ✓ стоячи або сидячи робити похитування головою ліворуч-праворуч (темп швидкий);

- ✓ стоячи або сидячи робити нахили голови уперед-назад (темп помірний);

- ✓ масаж лоба, злегка його погладжуючи, а також погладжуючи ділянки над бровами у напрямку до скронь;

- ✓ стоячи або сидячи проводити самомасаж шиї та потилиці, погладжуючи потилицю та шию у напрямку до торса.

5.4.2. Вимоги безпеки перед початком роботи

Перед початком роботи перевірити:

- ✓ Розміщення складових частин комп'ютера (монітор, принтер, та інших блоків);
- ✓ Монітор слід розташовувати таким чином, щоб кут зору на екран монітора становив 10-15 градусів, а відстань до екрана 400-800 мм;
- ✓ Цілісність, з'єднувальних кабелів;
- ✓ Підключення складових частин комп'ютера згідно схеми з'єднання;
- ✓ Наявність і стан захисних кожухів. Підготовку комп'ютера до роботи проводити у відключеному стані;
- ✓ Забезпечити освітлення робочого місця таким чином, щоб не утворювались відблиски від клавіатури та екрану відео монітора в напрямку очей працюючого.

5.4.3. Вимоги безпеки під час роботи та її завершення

Підключення комп'ютера проводити послідовно, згідно з інструкцією по експлуатації. Не підключати та не відключати з'єднувачі електроживлення при включеному комп'ютері. Під час перерв у роботі виключити монітор комп'ютера. Не залишати працюючий комп'ютер без нагляду.

Виключити комп'ютер в послідовності згідно з інструкцією по експлуатації. Доповідати керівнику за технічний стан комп'ютера, про всі зауваження і несправності в роботі комп'ютера, якщо вони мали місце. Привести в порядок робоче місце, прибрати зайві предмети та сміття. Покласти носії інформації до місць збереження.

5.4.4. Вимоги безпеки в аварійних ситуаціях

При появі шумів, скрипів запаху гарі та диму, які можуть привести до аварії, необхідно терміново відключити комп'ютер від електромережі і повідомити керівника та відповідального за технічний стан комп'ютера. Самостійна ліквідація несправності не дозволяється. При необхідності надати першу медичну допомогу потерпілим.

ВИСНОВКИ І ПРОПОЗИЦІЇ

У практичній сільськогосподарських підприємств існує управлінська задача планування використання технічного оснащення сільськогосподарського підприємства. На даний час у переважній більшості сільськогосподарських підприємств відповідні рішення приймають інтуїтивно, що переважно є однією із причин помилкових прийнятих рішень та відповідно витрат коштів. Для якісного планування використання технічного оснащення сільськогосподарського підприємства пропонується проектування та використання інтелектуальної інформаційної системи.

Виконаний огляд існуючих інтелектуальних інформаційних систем для планування діяльності у сільськогосподарських підприємствах свідчить про те, що вони є вагомим інструментарієм для планування діяльності сільськогосподарських підприємств. Вони призначені для вирішення багатьох задач, які скеровані на планування діяльності у сільськогосподарських підприємствах. Однак, попри їх переваги з погляду планування використання технічного оснащення сільськогосподарського підприємства вони мають низку недоліків.

Задля пришвидшення процесу розрахунку та уникнення помилок під час планування використання технічного оснащення сільськогосподарського підприємства була сформульована задача забезпечити автоматизацію процесу розрахунку використання технічного оснащення на підставі розроблення відповідної інтелектуальної інформаційної системи.

Основною перевагою запропонованої інтелектуальної інформаційної системи є те, що нею враховуються як виробничі умови виконання робіт у будь-який момент часу, так і можливості окремого сільськогосподарського підприємства залучити доступне технічне оснащення до виконання різних видів робіт. Окрім того, вона забезпечує формування графіка Ганта, який дозволить мінімізувати відхилення моментів часу відносно технологічно регламентованих термінів виконання сільськогосподарських робіт.

У процесі розв'язування задачі з регламентованим терміном виконання робіт за заданого технічного оснащення можуть виникнути дві ситуації: 1) регламентований період виконання робіт далекий від початку координат (рис. 1.1); регламентований період виконання робіт близький до початку їх відліку (початку виконання) (рис. 1.2). На рисунках варто звернути увагу на розташування контурів відносно початку координат. Це визначає сценарій залучення технічного оснащення для виробничих умов, що розглядаються.

На прикладі розглянуто варіант, у якому задана множина виконання сільськогосподарських робіт $J = \{1, 2, 3, 3, 4, 5, 6, 7\}$. При цьому залучається технічне оснащення заданої марки у кількості $m = 3$ одиниць із продуктивністю $H = \{12; 10; 8\}$. У результаті планування використання технічного оснащення за запропонованим методом побудовано графік виконання робіт після їх розподілу за технічним оснащенням із різною продуктивністю.

Для виконання проекту, зокрема завдань щодо створення програмного продукту інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства із віконним додатком для ПК, було здійснено вибір засобів для виконання проекту: Python 3.9; Tkinter; Matplotlib; Spyder4. Для моделювання майбутньої інтелектуальної інформаційної системи використовувався Diagrams.net

Для побудови моделі варіантів використання інтелектуальної інформаційної системи використовуємо UML, яка являє собою уніфіковану мову моделювання. Для побудови моделі варіантів використання інтелектуальної інформаційної системи нами використано безкоштовний сервіс Diagrams.net. Побудована нами структурна схема варіантів використання інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства передбачає дві групи об'єктів, які стосуються перевірки алгоритму та виконання планування.

Програмний продукт інтелектуальної інформаційної системи розділений на дві складові, які представляють графічний інтерфейс користувача та функціональний блок. Структурну схему класів інтелектуальної інформаційної

системи подано на рис. 3.2. Графічний інтерфейс користувача розроблений для зручності використання і представлений у вигляді вікон. Кожне вікно є окремим етапом виконання планування використання технічного оснащення в умовах сільськогосподарського підприємства або тестуванні алгоритмів.

Схема структурна послідовності інтелектуальної інформаційної системи (рис. 3.3) показує, як програмні об'єкти взаємодіють один з одним і послідовність викликів функцій у кожному процесі. Ось коротка послідовність дій, які виконує користувач у кожному з цих процесів. У режимі планування користувач вводить початкові налаштування генератора або вибирає у вікні інтерфейсу опцію «повне планування», представлену класом *ResearchOneDueDate* або *ResearchTwoDueDates* (залежно від виду завдань планування з одним або двома регламентами).

Нами виконано побудову моделі діяльності інтелектуальної інформаційної системи, яка представлена на рис. 3.4. Під час планування результатом є графічна форма відображення графіка Ганта, яка порівнює значення цільових функцій, побудованих для згенерованих вхідних даних, та їх нижніх меж (збільшення регламентованих термінів виконання робіт) для заданого технічного забезпечення.

Запропонований нами прототип діалогового вікна для користувача відображено нижче (рис. 4.1), де подано розміщення відповідних блоків. На панелі вибору сторінки із окремими кнопками пропонується користувачам здійснити вибір потрібної сторінки. При цьому кнопки розташовано у порядку виконання планування використання технічного оснащення для заданих умов сільськогосподарського підприємства. Зокрема, пропонуються кнопки «Завантажити початкові дані», «Тестування», «Планування із 1 регламентом», «Планування із 2 регламентами» та «Зберегти результати планування як ...» (рис. 4.2).

Наступна сторінка «Планування із 2 регламентами» відображена у вигляді графіка Ганта із результатами планування, що забезпечує візуальне представлення результатів планування та розрахунків щодо використання

технічного оснащення для заданих умов сільськогосподарського підприємства (рис. 4.4). Остання сторінка «Зберегти результати планування як ...» відображає панель із збереженням у результатів планування (рис. 4.5). У цьому файлі подано результати планування використання технічного оснащення для заданих умов сільськогосподарського підприємства.

Переважно бібліотеку Tkinter використовують для створення інтерфейсу користувача за допомогою коду мови Python. Завдяки цій бібліотеці створено діалогове вікно для користувача інтелектуальної інформаційної системи планування використання технічного оснащення для заданих умов сільськогосподарського підприємства, що представлено на рис. 4.2-4.5.

Нами створено блокнот у Tkinter із використанням віджету Frame, який забезпечує групування та керування іншими наявними віджетами (рис. 4.6). Він виконує роль контейнера, відповідальний за організацію розташування решти віджетів.

Представлений вище макет сторінки «Завантажити початкові дані» забезпечує вибір файла із початковими даними на підставі програмного коду за допомогою функції `open_file`, яка подана на рис. 4.7. Результати планування у діалоговому вікні користувача інтелектуальної інформаційної системи записують у файл із кодом формату `.txt`. Для цього написано функцію під назвою `save_file` (рис. 4.8).

Створення модуля планування використання технічного оснащення та виведення результатів виконували завдяки прописуванню коду класів *ResearchOneDueDate*, *Test*, *Research*, *Schedule*, які забезпечують вибір файла із початковими даними, перевірку алгоритму із заданим регламентом на виконання робіт, планування за алгоритмами та виведення результату планування.

Розроблено заходи щодо охорони праці забезпечать покращення умов праці програміста, який працює над розробленням інтелектуальної інформаційної системи планування використання технічного оснащення сільськогосподарського підприємства.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Верников, М. Руководителю предприятия. Внедрение системы автоматизации, основные проблемы и задания [Электронный ресурс]. Режим доступа: <http://www.vernikov.ru> (дата звернення: 14.12.2021).
2. Вовк С.Г., Жубрид М.Д., Цабак Н.І. Аспекти застосування систем підтримки прийняття рішень в управлінні сільгоспідприємством. Вісник Львівського державного аграрного університету: економіка АПК. 2007. № 14. С. 198-201.
3. Єрмаков О.Ю. Організація сільськогосподарського виробництва. Навч. мет. посібник. Друге видання, доп. і переробл. К.: НАУ, 2007. 266с.
4. Ільчук М.М., Зрібняк Л.Я. та ін. Організація і планування сільськогосподарського виробництва. Підручник для студентів вищих аграрних закладів освіти III-IV рівнів акредитації. За ред. М.М. Ільчука. К.: 2008. 757 с.
5. Коптелов А., Оситнянко О. Информационные технологии в сельском хозяйстве. Агробизнес: информатика – оборудование – технологии. – 2010. № 12. С. 60-64.
6. Кузьменко Н.Г. Вычислительные системы сети и телекоммуникации. Аппаратные средства. Красноярск: ИПЦ КГТУ, 2006. 204 с.
7. Лутц М. Программирование на Python. I том. СПб.: Символ-плюс, 2015. 992 с.
8. Марк Саммерфилд. Программирование на Python 3. Подробное руководство. Пер. с англ. СПб.: Символ-Плюс, 2013. 608 с.
9. Програмування числових методів мовою Python : підруч. / А. В. Анісімов, А. Ю. Дорошенко, С. Д. Погорілий, Я. Ю. Дорогий ; за ред. А. В. Анісімова. К. : Видавничо-поліграфічний центр «Київський університет», 2014. 640 с.
10. Прохоренко Н.А. Python 3 и PyQt. Разработка приложений. СПб.: БХВ-Петербург, 2012. 704 с.

11. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. М.: Альт Линукс, 2010. 126 с.
12. Чаплінський Ю.П. Мобільні інформаційні системи підтримки прийняття рішень. Наукова-технічна інформація. № 1. 2003. С. 22-26.
13. Шаманська О.І. Застосування інформаційних систем та технологій як пріоритетного напрямку ефективного функціонування та розвитку дорадчої діяльності в Україні. Ефективна економіка. 2015. № 4.
14. Hoogeveen J.A., Van de Velde S.L. Scheduling around a small common due date. European Journal of Operational Research. 1991. № 55. pp. 237-242.
15. Matplotlib [Електронний ресурс] – Режим доступу до ресурсу: <https://matplotlib.org/>. (дата звернення: 20.01.2022)
16. Michael L. Pinedo. Scheduling: Theory, Algorithms, and Systems. New York : Springer, 2008. 662 p.
17. Peter Brucker. Scheduling Algorithms. New York : Springer, 2007. 378 p.
18. Prakash S., Kumar A., Mishra R. B. MVC Architecture Driven Design and Agile Implementation of a Web-Based Software System, International Journal of Software Engineering & Applications (IJSEA), 4 (6), 2013. pp. 13-28.
19. Python (programming language) [Електронний ресурс]. Режим доступу: <https://bitly су/paHJe>. (дата звернення: 28.12.2021)
20. Python [Електронний ресурс] – Режим доступу до ресурсу: <https://www.python.org/>. (дата звернення: 12.11.2021)
21. The IDEF Process Modeling Methodology [Електронний ресурс] – режим доступу: <https://bitly су/RzJIXk 12>
22. Tkinter [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/library/tkinter.html> (дата звернення: 18.01.2022)
23. Tutorialspoint / Python [Електронний ресурс]. Режим доступу: <http://www.tutorialspoint.com/python/>

Додатки

Додаток А

Фрагмент програмного коду інтелектуальної інформаційної системи

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 11 21:43:35 2022

@author: Admin
"""

import tkinter as tk
from tkinter.filedialog import askopenfilename, asksaveasfilename

def open_file():
    """Відкриваємо файл із початковими даними"""
    filepath = askopenfilename(
        filetypes=[("Текстові файли", "*.txt"), ("Всі файли", "*.*")]
    )
    if not filepath:
        return
    txt_edit.delete("1.0", tk.END)
    with open(filepath, "r") as input_file:
        text = input_file.read()
        txt_edit.insert(tk.END, text)
    window.title(f"Інтелектуальна інформаційна система планування використання
технічного оснащення сільськогосподарського підприємства - {filepath}")

def save_file():
    """Зберігаємо поточний файл как новий файл."""
    filepath = asksaveasfilename(
        defaultextension="txt",
        filetypes=[("Текстові файли", "*.txt"), ("Всі файли", "*.*")],
    )
    if not filepath:
        return
    with open(filepath, "w") as output_file:
        text = txt_edit.get("1.0", tk.END)
        output_file.write(text)
    window.title(f"Результати планування використання технічного оснащення
сільськогосподарського підприємства - {filepath}")

window = tk.Tk()
window.title("Інтелектуальна інформаційна система планування використання
технічного оснащення сільськогосподарського підприємства")
window.rowconfigure(0, minsize=500, weight=1)
window.columnconfigure(1, minsize=700, weight=1)

txt_edit = tk.Text(window)
fr_buttons = tk.Frame(window, relief=tk.RAISED, bd=2)
btn_open = tk.Button(fr_buttons, text="Завантажити початкові дані",
command=open_file)
btn_test = tk.Button(fr_buttons, text="Тестування", command=open_file)
btn_plan_1 = tk.Button(fr_buttons, text="Планування із 1 регламентом",
command=open_file)
btn_plan_2 = tk.Button(fr_buttons, text="Планування із 2 регламентами",
command=open_file)
btn_save = tk.Button(fr_buttons, text="Зберегти результати планування як...",
command=save_file)

btn_open.grid(row=0, column=0, sticky="ew", padx=5, pady=5)
btn_test.grid(row=1, column=0, sticky="ew", padx=5, pady=5)
btn_plan_1.grid(row=2, column=0, sticky="ew", padx=5, pady=5)
```

```

btn_plan_2.grid(row=3, column=0, sticky="ew", padx=5, pady=5)
btn_save.grid(row=4, column=0, sticky="ew", padx=5, pady=5)

fr_buttons.grid(row=0, column=0, sticky="ns")
txt_edit.grid(row=0, column=1, sticky="nsew")

window.mainloop()

from params import *
import pandas as pd
import gantt
import tkinter

class ResearchOneDueDate(tkinter.Frame):
    def __init__(self, parent, controller):
        tkinter.Frame.__init__(self, parent)
        self.controller = controller

        arr = ['S', 'M', 'L']

        self.button = tkinter.Button(self, text='Назад', font=font,
                                     bg=light_grey_color, fg=dark_grey_color,
                                     command=lambda:
self.reset_frame()).grid(row=0,
                                     columnspan=2, sticky='W')
        tkinter.Label(self, text='Параметри генератора',
                      font=font_header).grid(row=1, columnspan=2, padx=275,
pady=50)

        tkinter.Label(self, text='Кількість виконуваних робіт',
font=font).grid(row=2, column=0, padx=20, pady=10, sticky='ew')

        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=2, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Тривалість регламентованого терміну',
font=font).grid(row=3, column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=3, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Обсяг робіт', font=font).grid(row=4, column=0,
padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=4, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Кількість технічного оснащення',
font=font).grid(row=5, column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)
        self.d_menu.grid(row=5, column=1, padx=10, pady=10, sticky='w')
        tkinter.Label(self, text='Продуктивності технічного оснащення',
font=font).grid(row=6, column=0, padx=20, pady=10, sticky='ew')
        self.d_var = tkinter.StringVar()
        self.d_var.set(arr[0])
        self.d_menu = tkinter.OptionMenu(self, self.d_var, *arr)
        self.d_menu.config(font=font)

```

```

self.d_menu.grid(row=6, column=1, padx=10, pady=10, sticky='w')

tkinter.Button(self, text='Продовжити', font=font_bold,
                command=lambda: self.kek()).grid(row=7, column=0,
pady=50)
tkinter.Button(self, text='Повне планування', font=font_bold,
                command=lambda: self.kek()).grid(row=7, column=1,
pady=50, sticky='w')

def reset_frame(self):
    from MainPage import MainPage
    self.controller.frames[MainPage].tkraise()

def kek(self):
    print('kek')

import matplotlib.pyplot as plt
import schedule

class Gantt:
    def __init__(self, path):
        self.filename = path

    with open(self.filename, 'r') as f:
        self.num_d = int(f.readline())
        self.num_jobs = []
        self.jobs = []
        self.d = []

        for _ in range(self.num_d):
            self.num_jobs.append(int(f.readline()))
            self.jobs.append([int(job) for job in f.readline().split()])
            self.d.append(int(f.readline()))

        self.num_machines = int(f.readline())
        self.machines = [int(machine) for machine in f.readline().split()]

    def get_schedule(self):
        return Schedule().build_schedule(self.num_d, self.d, self.jobs,
self.num_machines, self.machines)

    def draw_graph(self):
        schedule = self.get_schedule()

        fig, gnt = plt.subplots()
        gnt.set_title('Побудований план використання технічного оснащення')
        gnt.set_ylim(0, (self.num_machines + 1) * 10 + 10)
        gnt.set_xlim(0, max([job[0] + job[1] for machine in schedule for
due_date in machine for job in due_date]) + 5)
        gnt.set_xlabel('Час')
        gnt.set_ylabel('Технічне оснащення')
        gnt.grid(True)

        yticks = [(i + 1) * 10 + 5 for i in range(self.num_machines)]
        yticklabels = [str(i + 1) for i in range(self.num_machines)]

        gnt.set_yticks(yticks)
        gnt.set_yticklabels(yticklabels)

        # setting the schedules
        for i in range(self.num_machines):
            for j in range(2):
                gnt.broken_barh(schedule[i][j], ((i + 1) * 10, 9),
                                facecolors=('orange', 'green', 'blue') *

```

```

len(self.num_jobs), # mark each job with diff
color
    edgecolors='black') # color of borders
    for due_date in schedule[i]:
        for job in due_date:
            gnt.text(x=job[0] + job[1] / 2, # x = gorizontal position of
the label (center of the bar)
                    y=(i + 1) * 10 + 4, # y = vertical position of the
label
                    s=job[1], # text (length of the job)
                    ha='center', va='center', color='white', size=15)
# setting the due dates
for i in range(self.num_d):
    gnt.axvline(x=self.d[i], c='black', linewidth=5)
    # gnt.text(x=self.d[i] + 0.5, y=1, s=f'd = {self.d[i]}',
color='black', size=10)
    # possible way to

# set xtick for due date
gnt.set_xticks(list(gnt.get_xticks()) + self.d) # set xtick for due date
fig.show()

```