

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних систем

«До захисту в ЕК»
Директор інституту(декан факультету)

(підпис) Форсюк А. В.
(прізвище та ініціали)

« ____ » _____ 2020р.

«До захисту допущено»
Завідувач кафедри

(підпис) Чумаченко С. М.
(прізвище та ініціали)

« ____ » _____ 2020р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності 122 «Комп'ютерні науки та інформаційні технології»
(код та назва спеціальності)
освітньо-професійної програми «Комп'ютерні науки»

на тему: «Автоматизація розрахунків теодолітних точок для КП
«Київський інститут земельних відносин»»

Виконав: здобувач 4 курсу, групи 4

Андрійчук Тарас Юрійович
(прізвище, ім'я, по батькові повністю) _____ (підпис)

Керівник Андріюк Олена Петрівна
(прізвище, ім'я та по батькові повністю) _____ (підпис)

Консультанти Андріюк Олена Петрівна
(прізвище та ініціали) _____ (підпис)

(прізвище та ініціали) _____ (підпис)

(прізвище та ініціали) _____ (підпис)

Рецензент Смітюх Ярослав Володимирович
(прізвище та ініціали) _____ (підпис)

Засвідчую, що в цій кваліфікаційній
роботі немає запозичень із праць
інших авторів без відповідних
посилань.

Здобувач _____
(підпис)

Київ – 2020 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем

Кафедра Інформаційних систем

Освітній ступінь бакалавр

Спеціальність _____

(код і назва)

Освітньо-професійна програма «Комп'ютерні науки»

(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

інформаційних систем

с.н.с. С.М.Чумаченко

“ ____ ” _____ 2020 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

_____ Андрійчука Тараса Юрійовича _____
(прізвище, ім'я по батькові)

1. Тема роботи «Автоматизація розрахунків теодолітних точок для КП «Київський інститут земельних відносин»»

керівник роботи Андріюк Олена Петрівна, кандидат фізико-математичних наук, доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “ 27 ” квітня 2020 року № 269-кс

2. Строк подання здобувачем роботи 07.06.2020

3. Вихідні дані до роботи 1) горизонтальне положення лінії;

2) відстань.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) 1) Вступ.

2) Розділ 1. Системний аналіз об'єкту автоматизації та постановка задачі на проектування.

3) Розділ 2. Розробка автоматизації.

4) Розділ 3. Організація праці.

5) Висновки

5. Перелік графічного матеріалу

1. Організаційна структура КП «Київський інститут земельних відносин»

2. Організаційна структура відділу автоматизації робочих процесів та програмування

3. Функціональна модель діяльності відділу автоматизації робочих процесів та програмування

4. Концептуальна модель діяльності відділу відділу автоматизації робочих процесів та програмування

5. Інтерфейс користувача

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Системний аналіз об'єкту автоматизації та постановка задачі на проектування.	к. фіз-мат н., доцент Андріюк О. П.		
Розробка автоматизації.	к. фіз-мат н., доцент Андріюк О. П.		
Організація праці.	к. фіз-мат н., доцент Андріюк О. П.		

7. Дата видачі завдання 10 травня 2020 року

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Передпроектне дослідження та системний аналіз діяльності відділу автоматизації робочих процесів та програмування	11.03.2020	
2	Розробка функціональної та концептуальної моделей діяльності відділу автоматизації робочих процесів та програмування	23.03.2020	
3	Розрахунок техніко-економічного обґрунтування доцільності розробки	01.04.2020	
4	Визначення та реалізація функцій підсистеми	06.04.2020	
5	Розробка плану організації праці	25.05.2020	
6	Оформлення роботи та розробка презентації	01.06.2020	

Здобувач

(підпис)

Андрійчук Т. Ю.

(прізвище та ініціали)

Керівник роботи

(підпис)

Андріюк О. П.

(прізвище та ініціали)

АНОТАЦІЯ

Головною метою цієї кваліфікаційної роботи є розробка та створення інформаційної підсистеми призначеної для забезпечення точних розрахунків при виконанні геодезичних робіт, з метою унеможливлення викривлення даних про земельну ділянку, підвищення ефективності роботи персоналу та автоматизації процесу розрахунків.

Основними цілями створення даної підсистеми є:

- забезпечення потреби геодезистів;
- зменшення витрат часу працівників на розрахунки «вручну»;
- збільшення обсягів виконання робіт;
- раціональне використання коштів шляхом автоматизації процесу розрахунків власними силами.

Робота включає в себе схеми декомпозиції і опис реалізації підсистеми.

Предметом дослідження є автоматизація розрахунків теодолітних точок.

Результатом має бути розроблена підсистема у вигляді калькулятора та розміщена на офіційному веб-порталі. Враховуючи площу м. Києва (83,6 тис. га) та кількість земельних ділянок очікується значний попит на користування калькулятором, що в свою чергу популяризує підприємство на ринку професійних послуг та підтверджує його передове становище на ринку геодезичних послуг.

Кваліфікаційна робота містить 82 сторінок, 4 таблиці, 27 рисунків, 4 додатки і 9 літературних джерел.

КЛЮЧОВІ СЛОВА: АВТОМАТИЗАЦІЯ, ГЕОДЕЗІЯ, ТЕОДОЛІТНИЙ ХІД, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ANNOTATION

The main purpose of this qualification work is to develop and create an information subsystem designed to ensure accurate calculations when performing geodetic works, in order to prevent distortion of land data, increase staff efficiency and automate the calculation process.

The main objectives of this subsystem are:

- meeting the needs of surveyors;
- reduction of employees' time spent on "manual" payments;
- increase in the volume of work performed;
- rational use of funds by automating the process of settlements on their own.

The work includes decomposition schemes and a description of the implementation of the subsystem.

The subject of research is the automation of calculations of theodolite courses.

The result should be a subsystem in the form of a calculator and posted on the official web portal. Given the area of Kyiv (83.6 thousand hectares) and the number of land plots, significant demand for the use of the calculator is expected, which in turn promotes the company in the market of professional services and confirms its leading position in the market of geodetic services.

Qualification work contains 82 pages, 4 tables, 27 figures, 4 appendices and 9 references.

KEY WORDS: AUTOMATION, GEODESY, THEODOLITE PROGRESS, SOFTWARE.

ВСТУП

КП «Київський інститут земельних відносин» надає широкий спектр послуг в галузі землевпорядкування, геодезії, топографії та інженерної геодезії, маючи розвинену матеріально-технічну базу для якісного виконання робіт будь-якої складності, сучасне обладнання, аналогів якому немає в Україні, ліцензійне програмне забезпечення та команду сертифікованих спеціалістів задіяних в оперативному виконанні завдань. Враховуючи вартість 1 м² землі м. Києва при виконанні цих завдань важливе значення мають точні розрахунки.

Результатом кваліфікаційної роботи має бути розроблена підсистема у вигляді калькулятора, що буде розміщена на офіційному веб-порталі та забезпечить точність розрахунків.

Враховуючи площу м. Києва (83,6 тис. га) та кількість земельних ділянок очікується, що попит на користування калькулятором буде значним, що в свою чергу буде популяризувати компанію на ринку професійних послуг та підтверджувати передове становище підприємства на ринку геодезичних послуг.

ЗМІСТ

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТУ АВТОМАТИЗАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ.....	9
1.1 Найменування підприємства	9
1.1.1 Загальна характеристика КП «Київський інститут земельних відносин»	9
1.2 Організаційна структура підприємства	16
1.2.1 Загальна схема організаційної структури підприємства	16
1.2.2 Схема та функції відділу автоматизації робочих процесів та програмування.....	17
1.2.3 Взаємодія з іншими відділами	19
1.3 Аналіз нинішнього стану автоматизації	19
1.4 Розроблення функціональної моделі	19
1.5 Аналіз систем аналогів управління проектом	20
1.5.1 Аналіз калькулятора «ДГМ»	20
1.5.2 Аналіз програми « <i>Geodetic Survey Solution</i> »	21
1.5.3 Порівняння систем-аналогів	21
1.6 Обґрунтування доцільності проектування і розроблення підсистеми.....	22
1.7 Концептуальна модель підсистеми	23
1.8 Постановка задачі на проектування.....	23
1.8.1 Призначення та цілі створення підсистеми	23
1.8.2 Вимоги до створюваної підсистеми	24
1.8.3 Вхідні та вихідні дані підсистеми.....	24
РОЗДІЛ 2. РОЗРОБКА АВТОМАТИЗАЦІЇ	25
2.1 Інформаційне забезпечення проекту	25
2.2 Програмне забезпечення розробки	25
2.3 Технічне забезпечення розробки інформаційної системи....	25
2.4 Розроблення інтерфейсу	26
2.5 Алгоритмічне забезпечення розробки системи.....	34
2.6 Інструкція користувача.....	34

2.7	Визначення витрат на розроблення системи	38
2.7.1	Визначення розміру оплати праці	38
2.7.2	Визначення розміру витрат на заробітну плату	38
2.7.3	Визначення розміру витрат на нарахування на заробітну плату	39
2.7.4	Визначення розміру амортизаційних витрат	39
2.7.5	Визначення розміру загальновиробничих витрат	40
РОЗДІЛ 3. ОРГАНІЗАЦІЯ ПРАЦІ.....		42
ВИСНОВОК		49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		50
ДОДАТКИ		51

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТУ АВТОМАТИЗАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ

1.1 Найменування підприємства

- повне – Комунальне підприємство «Київський інститут земельних відносин»
- скорочене – КП «КІЗВ»

Місце знаходження :

- юридична адреса:
01001, м. Київ вул. Хрещатик, 32А
- фізична адреса:
01001, м. Київ вул. Хрещатик, 32А

Сайт: <https://kizv.gov.ua/>

1.1.1 Загальна характеристика КП «Київський інститут земельних відносин»

Комунальне підприємство «Київський інститут земельних відносин», створено рішенням Київської міської ради від 19.12.2002р. № 166/326 та підпорядковується Департаменту земельних ресурсів виконавчого органу Київської міської ради (Київської міської державної адміністрації).

Діяльність підприємства направлена на виконання проектно-вишукувальних, науково-дослідних, дослідно-конструкторських, дослідно-технологічних, спеціальних та інших робіт в галузі ведення міського земельного кадастру, землеустрою та моніторингу земель з використанням новітніх технологій з метою отримання прибутку на основі задоволення потреб громадян, державних та інших підприємств, установ і організацій у роботах та послугах, що надаються у сферах, визначених предметом діяльності.

Послуги з геодезії потрібні:

- Замовникам будівельних робіт
- Девелоперським компаніям
- Фізичним особам, які хочуть побудувати об'єкт нерухомості
- Сільським / міським радам, які розробляють генеральні плани населених пунктів
- Архітектурним майстерням й проектним організаціям
- Будівельним компаніям

Майбутнє будівельних об'єктів безпосередньо залежить від достовірності геодезичних результатів.

Практичні задачі геодезії надзвичайно різноманітні. До їх числа відносять:

- визначення положення окремих точок земної поверхні в обраній системі координат;
- складання карт і планів місцевості;
- виконання вимірювань, необхідних для вишукування, проектування, будівництва і експлуатації будівель і споруд.

Всі завдання геодезії вирішуються за допомогою спеціальних вимірювань, які називають – геодезичними.

На виконання рішення Київської міської ради від 15 лютого 2018 року N 47/4111 в Києві почала працювати міська геодезична мережа[1], яка забезпечить вихідними даними для виконання землевпорядних, топографо-геодезичних, геологічних, маркшейдерських, картографічних робіт, кадастрової та містобудівної діяльності.

Отримати інформацію про пункти МГМ (включаючи опис їх центрів, місцезнаходження, значення координат, висот і вимірів) у місцевій системі координат м. Києва (ідентифікатор МСК-80) можуть юридичні особи та фізичні особи-підприємці, які відповідно до законодавства України мають право виконувати вищезгадані роботи.

Утримувачем міської геодезичної мережі визначено Департамент земельних ресурсів, адміністратором – комунальне підприємство «Київський інститут земельних відносин».

Міська геодезична мережа міста Київ – це єдиний інформаційний простір, призначений для формування, адміністрування та ведення єдиної бази геодезичних даних, використання і поширення інформації про пункти міської геодезичної мережі Києва.

Доступний, корисний і зручний у користування веб-портал забезпечує вихідними даними для виконання землепорядних, топографо-геодезичних, геологічних, маркшейдерських, картографічних робіт, кадастрової та містобудівної діяльності.

Отримати необхідну для виконання робіт вихідну інформацію, координати пунктів МГМ та кроки їх місця розташування, можуть юридичні особи та фізичні особи-підприємці, які відповідно до законодавства України мають право виконувати вищезгадані роботи.

В даний час є нові технології геодезичного забезпечення зйомки земної поверхні. Крім оптико-механічних приладів, використовуються нові електронні тахеометри, супутникові навігаційні системи. Для практичного використання нових технік та технологій потрібні перепідготовка фахівців, визначення часу виконання робіт і чисельності польових бригад, та ін. Незважаючи на високу продуктивність, ця техніка має деякі особливості, які представляють певні труднощі при використанні. Наприклад, прийом сигналів супутникової навігаційної системи представляється складним на щільно забудованій території, що ускладнює можливість використання цієї технології.

Збиток від можливої похибки не повинен перевищувати витрат на забезпечення точності геодезичних побудов, за допомогою яких усувається цей збиток.

Одним з методів зниження шкоди є використання супутникових навігаційних методів. Але основною проблемою даного методу геодезичних вимірювань є велика щільність міської забудови. У таких умовах визначення координат меж земельних ділянок стикається з суттєвими перешкодами і їх доводиться усувати додатковими геодезичними вимірами. Для цього точки згущення або знімальної основи необхідно підняти над перешкодами і розташувати на дахах будівель. Після чого за допомогою зворотної геодезичної зарубки визначаються координати пункту згущення «Р». Витрати на виконання додаткових геодезичних вимірювань також враховуються при побудові графіка.

Використання супутникової навігаційної системи в геодезичному забезпеченні господарських робіт залучає високою продуктивністю праці.

Тому при рекогностуванні об'єкту зйомки доцільно розглянути можливість отримання координат точок, не тільки основи, але й об'єктів зйомки. Такими точками можуть бути вершини кутів кордонів та інші об'єкти зйомки. Визначення таких точок слід підтвердити спеціальним абрисом і розрахованим допустимим кутом маски на перешкоди. Щільно забудована багатопверховими будинками ділянка і територія і навколо неї не дозволяють спостерігати без спотворень сигналів супутникової навігаційної системи. Однак рекогностування показало, що може бути можливим в оцінці координати в вершині кута.

Для рекогностування ділянки, яка відведена бригаді для топографічного знімання потрібно відшукати пункти геодезичної основи.

Основними задачами рекогностування є:

- вибір кінцевого місця положення кожного пункту на місцевості, тобто уточнення схеми проекрованої мережі;

- кінцевий розрахунок висот геодезичних знаків, які встановлюються на пунктах;
- вибір типів знаків, підземних центрів і визначення глибини їх закладки;
- уточнення даних, на основі яких при проектуванні були вирішені питання організації робіт і складена кошторисна частина проекту.

Не можна встановлювати пункти поблизу інженерних споруд і жилих будинків, залізничних і автомобільних доріг, ліній високої напруги, тощо, в місцях де не може бути забезпечене довготривале їх збереження (знаків і підземних центрів).

Відрекогностовані пункти позначають на місцевості курганами, віхами або стовпами, затесаними на ближніх деревах з написами на цих затесах, тощо.

Методи рекогностування можна підрозділити на дві групи: візуальні і інструментальні.

Візуальні методи використовуються при наявності оптичної видимості між пунктами, наприклад, у відкритій і напіввідкритій місцевості, де видимість між пунктами встановлюється із землі за допомогою бінокля або теодоліта і знаходження місця пункту не викликає труднощів. Якщо немає видимості із землі, її можна добитися шляхом установки на пунктах віх, телескопічних щогл, спеціальних рекогностувальних драбин та інших пристроїв. В деяких випадках можна встановити видимість, піднімаючись на дерева або місцеві предмети.

При наявності прямої видимості між пунктами висоти геодезичних знаків визначають шляхом безпосереднього вимірювання тієї висоти, з якої відкривається видимість по всім напрямкам на навколишні пункти, які необхідно спостерігати.

Інструментальні методи рекогностування застосовують тоді, коли шляхом візуальних спостережень не вдається встановити видимість між наміченими пунктами і визначитися на місцевості. Такі умови можуть бути у лісовій і пагорбистій місцевості, відкритій рівнинній місцевості з недостатнім числом орієнтирів. При цьому місця положення запроектованих пунктів, перевищення між перешкодами і пунктами визначають шляхом прокладання висотно-теодолітних або мензульних ходів, геодезичних засічок, тригонометричного та барометричного нівелювання, тощо.

Під час рекогностування збирають дані, необхідні для успішної організації робіт по будівництву знаків і закладці центрів, високоточним астрономо-геодезичним вимірам. Вказують зручні під'їзди до пунктів, можливі місця заготовки лісу і будівельних матеріалів, джерела питної води, можливі місця вертолітних майданчиків, злітно-посадочних смуг у районі робіт, тощо.

В організаційному відношенні рекогностування може виконуватися або до початку будівництва знаків, або сумісно з будівництвом.

Перша схема організації робіт може застосовуватись у відкритій рівнинній, пагорбистій і гірській місцевості, де визначення висот знаків і встановлення видимості між пунктами виконують візуальними методами.

Друга схема організації рекогностування сумісно з побудовою є більш прогресивною, тому що побудовані знаки широко використовуються при рекогностуванні, дозволяючи оперативно усувати неточності в розрахунках висоти вже побудованих знаків, шляхом коректування висот наступних знаків; виключаються випадки появи невидимості по напрямкам, що спостерігаються, підвищується якість рекогностування і побудови знаків.

Основним документом є рекогностувальний журнал.

Теодолітний хід слугує зйомочною основою для виконання топографічних зйомок. Теодолітні ходи проектують на існуючих картах та планах крупного масштабу у вигляді замкнених полігонів і розімкнених ходів. Вибір точок теодолітного ходу виконують у польових умовах. При цьому дотримуються наступних вимог:

1. Забезпечити гарну видимість суміжних точок ходу і якомога більшої кількості об'єктів місцевості в радіусі 100 – 150 м.
2. Місце навколо станції має бути зручним для встановлення теодоліта.
3. Забезпечити довгострокове збереження точки.
4. Довжини ліній теодолітного ходу мають бути в межах 40–350м.
5. Лінії між точками мусять проходити по місцевості, найбільш зручній для виконання лінійних вимірювань.

Просторове положення (координати) станцій теодолітного ходу визначають на основі кутових і лінійних вимірювань. Горизонтальні кути вимірюють теодолітом повним прийомом, кути нахилу вимірюють при одному положенні вертикального круга. Довжини ліній вимірюють мірними стрічками в прямому і зворотному напрямках. Математична обробка результатів вимірювань у теодолітному ході полягає у визначенні допустимих нев'язок кутових і лінійних вимірювань, введенні поправок в результати вимірювань та обчисленні координат точок теодолітного хода.

1.2 Організаційна структура підприємства

1.2.1 Загальна схема організаційної структури підприємства

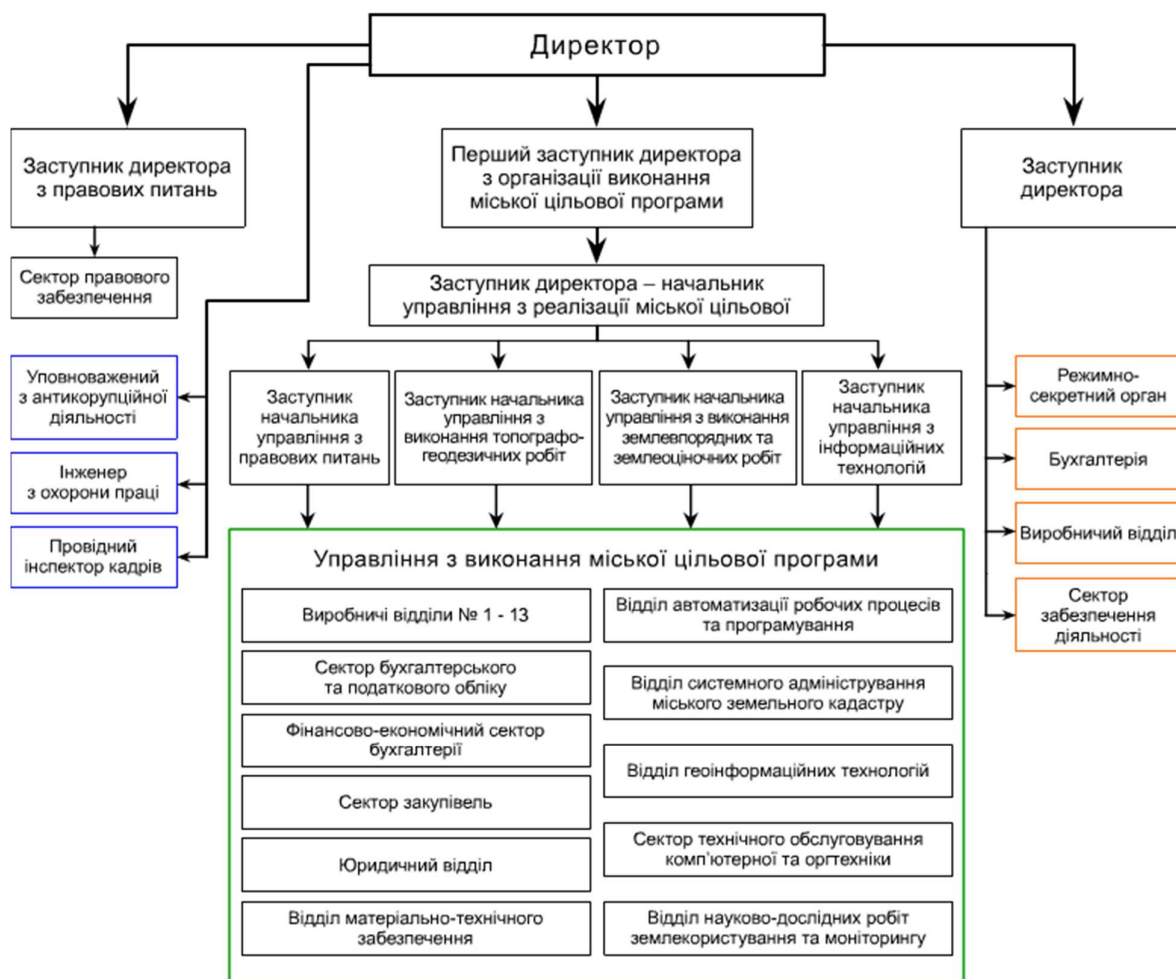


Рис. 1.1. Організаційна структура підприємства

1.2.2 Схема та функції відділу автоматизації робочих процесів та програмування

Відділ автоматизації робочих процесів та програмування має такі завдання та обов'язки:

- Розробляє програми відповідно заходів Міської цільової програми використання та охорони земель міста Києва на 2019-2021 роки, які забезпечують можливість виконання алгоритму і, відповідно, поставлену задачу засобами обчислювальної техніки, проводить їх тестування і налагодження.
- Розробляє технологію розв'язання задачі на всіх етапах.
- Визначає інформацію, яка підлягає обробленню засобами обчислювальної техніки, її обсяги, структуру, макети і схеми вводу, оброблення, зберігання і видавання інформації, методи її контролю.
- Визначає обсяги і зміст даних тестових прикладів, які забезпечують найбільш повну перевірку відповідності програм їх функціональному призначенню.
- Виконує роботи під час підготовки програм до налагодження і проводить їх налагодження.
- Розроблює інструкції на роботи з програмами, оформлює необхідну технічну документацію.
- Визначає можливість використання готових програм, розроблених іншими підприємствами (установами). Здійснює супроводження впроваджених програм і програмних засобів.
- Бере участь у проектних роботах.
- На основі логічного аналізу проводить камеральну перевірку програм.
- Визначає сукупність даних, що забезпечують урахування максимального числа умов, які включено до програми, виконує роботи з її підготовки до налагодження.

- Здійснює запуск налагоджених програм і введення вихідних даних, які визначаються умовами поставлених задач.
- Проводить коректування розробленої програми на основі аналізу вихідних даних.
- Розроблює і впроваджує методи автоматизації програмування, системи автоматичної перевірки правильності програм, типові і стандартні програмні засоби, програмуючі програми, транслятори, вхідні алгоритмічні мови, складає технологію оброблення інформації.
- Виконує роботи з уніфікації і типізації обчислювальних процесів.
- Бере участь у створенні каталогів і картотек стандартних програм, розробленні форм документів, які підлягають машинному обробленню, у проектних роботах, що стосуються розширення сфери застосування обчислювальної техніки.

Організаційна схема відділу автоматизації робочих процесів та програмування наведена на рис. 1.2.2.



Рис 1.2. Організаційна структура відділу автоматизації робочих процесів та програмування

1.2.3 Взаємодія з іншими відділами

В процесі функціонування відділ взаємодіє з усіма іншими підрозділами підприємства шляхом автоматизації робочого процесу: розробки програмного продукту та його впровадження.

1.3 Аналіз нинішнього стану автоматизації

На всіх комп'ютерах підприємства встановлено ОС Windows 10 64 bit та базовий пакет програм Microsoft Office. Відділ автоматизації робочих процесів та програмування пов'язаний з майже усією діяльністю комунального підприємства, але на нинішньому етапі підрахунки теодолітних ходів проводяться за допомогою звичайних калькуляторів, що встановлені на ПК або інших пристроях та таблиць Excel. Отже, весь процес займає велику частину робочого часу і є не точним за рахунок округлення результатів, що призводить до економічних збитків. Економічні збитки зумовлені такими чинниками:

- велика кількість часу на розрахунки;
- оплата праці окремих кадрів, що займаються розрахунками;
- неточні розрахунки та помилки в них.

1.4 Розроблення функціональної моделі

Для функціонального моделювання бізнес-процесу було застосовано програмний засіб AllFusion Process Modeler, адже ця система допомагає чітко документувати важливі аспекти будь-яких процесів. Моделювання процесів дозволяє проаналізувати як підсистема функціонує в цілому.

Для створення функціональної моделі застосовувалася методологія IDEF0 [Додаток А]. Основу методології IDEF0 складає графічна мова опису бізнес-процесів. Модель в нотації IDEF0 являє собою набір ієрархічно впорядкованих і взаємопов'язаних діаграм.

Вхідними даними є:

- тип ходу:
 - замкнений;
 - розімкнений;
- кут нахилу γ ;
- дирекційний кут;
- кут β (вимірний);
- координати пункту X та Y;
- відстань.

Вихідними даними є результат розрахунку:

- горизонтальне положення лінії d;
- відстань L.

1.5 Аналіз систем аналогів управління проектом

На сучасному ринку існує декілька програм, що використовують для розрахунків у геодезії, розглянемо декілька програм.

1.5.1 Аналіз калькулятора «ДГМ»

Геодезичний калькулятор призначений для перетворення та трансформування:

- координат точок між системами координат СК-42, СК-63, УСК-2000 та похідних від них місцевих систем координат в межах території України.
- плоских прямокутних координат за такими параметрами: зміщення по осі абсцис, зміщення по осі ординат, кут повороту, масштабний коефіцієнт.
- координат точок із зони в зону.

1.5.2 Аналіз програми «*Geodetic Survey Solution*»

Програма «*Geodetic Survey Solution*» призначена для розрахунку та урівняння нівелірних ходів, а також розрахунку пікетів. Дозволяє виконати імпорту вихідних координат з файлів, отриманих з GPS приймачів, та даних з електронних тахеометрів. На будь-якому етапі розрахунку можливе ручне введення і коригування даних вимірювань. Якщо теодолітний хід розраховується в умовній системі координат, достатньо змінити вихідні пункти для перерахунку всіх координат теодолітних ходів та пікетів. Головне вікно програми містить список всіх проектів розрахунку теодолітних ходів та карту, на якій вони відображаються. Можливий швидкий пошук теодолітних ходів та позиціонування на карті. Налаштовуваний режим переміщення по параметрах, при ручному введенні, дозволяє істотно збільшити продуктивність введення даних польових вимірювань. Передбачена можливість перетворення вихідних координат теодолітного ходу. Вікно з малюнком теодолітного ходу та пікетами відображає введені дані. Програма містить готовий набір звітних форм.

1.5.3 Порівняння систем-аналогів

У табл. 1.5.3.1 наведено результати порівняння існуючих на ринку програм, які могли б використовуватись КП «Київський інститут земельних відносин», за заданими вище критеріями. Проаналізувавши програмні продукти, можна сформулювати їх основні недоліки щодо використання :

- висока ціна;
- незрозумілий графічний інтерфейс користувача;
- недостатня звітність;
- недостатня надійність.

	Калькулятор ДГМ	Geodetic Survey Solution
Зручність використання	—	—
Крос-платформність	+	+
Звітність	—	+
Надійність	низька	висока
Ціна	безкоштовно	від \$3000

Таблиця 1.5.3.1. Результати порівняння існуючих систем-аналогів

Приклади калькуляторів, що були розглянуті, не здійснюють розрахунки для поставленої задачі. Аналогів калькулятору теодолітних точок не існує. Отже жодна з наведених систем аналогів не підходить для впровадження у КП «Київський інститут земельних відносин».

1.6 Обґрунтування доцільності проектування і розроблення підсистеми

Як зазначено вище, підприємство КП «КІЗВ» потребує автоматизації розрахунків теодолітних точок. З наведених у п. 1.5.3. результатів порівняння існуючих на ринку ІТ систем-аналогів видно, що розглянуті програмні продукти є занадто дорогими або мають функціональні обмеження і не практичність використання та не виконують поставлених задач. Отже, створення підсистеми, яка би повністю відповідала вимогам роботи, дійсно є актуальним і обґрунтованим завданням.

Отримати необхідну для виконання робіт вихідну інформацію, координати пунктів МГМ та кроки їх місця розташування, можуть юридичні особи та фізичні особи-підприємці, які відповідно до законодавства України мають право виконувати вищезгадані роботи.

Отже, інформація про координати пунктів МГМ доступна тільки професійним користувачам, після отримання інформації про координати пунктів МГМ вони заносяться користувачем калькулятору особисто.

1.7 Концептуальна модель підсистеми

Концептуальна модель створюваної підсистеми відображає схему та зв'язки з користувачем функціоналу, що в свою чергу популяризує на ринку професійних послуг та підтверджує передове становище підприємства на ринку геодезичних послуг. Цей функціонал буде забезпечувати точні розрахунки, які на даний момент виконуються геодезистами вручну.

1.8 Постановка задачі на проектування

З метою забезпечення точних розрахунків при виконанні геодезичних робіт, з метою унеможливлення викривлення даних про земельну ділянку, було вирішено створити підсистему у вигляді калькулятору теодолітних точок з вільним доступом користувачів.

1.8.1 Призначення та цілі створення підсистеми

Розроблювана інформаційна підсистема призначена для підвищення ефективності роботи, зменшення витрат та автоматизації розрахунків теодолітних точок.

Основними цілями створення даної підсистеми є:

- забезпечення потреби геодезистів;
- зменшення витрат часу працівників на розрахунки «вручну»;
- збільшення обсягів виконання робіт;
- раціональне використання коштів шляхом автоматизації процесу розрахунків власними силами.

1.8.2 Вимоги до створюваної підсистеми

Підсистема, що буде розміщена на офіційному веб-ресурсі підприємства[2] повинна бути у вільному доступі з будь-якого пристрою.

1.8.3 Вхідні та вихідні дані підсистеми.

Вхідні дані:

- тип ходу:
 - замкнений;
 - розімкнений;
- кут нахилу γ ;
- дирекційний кут;
- кут β (виміряний);
- координати пункту X та Y;
- відстань.

Вихідними даними є результат розрахунку:

- горизонтальне положення лінії d;
- відстань L.

РОЗДІЛ 2. РОЗРОБКА АВТОМАТИЗАЦІЇ

2.1 Інформаційне забезпечення проекту

Важливим етапом створення системи є розробка інформаційного забезпечення. Інформаційне забезпечення визначає інформацію, необхідну для управління процесами, створення умов функціонування ресурсу, забезпечення необхідною інформацією та отримання інформації.

Всю інформацію для подальшої обробки вносить користувач ресурсу:

- Координати точок;
- Кути;
- Відстань.

2.2 Програмне забезпечення розробки

Головна ціль програмного забезпечення – це виконання задач, що потребують автоматизації розрахунків та забезпечення максимально ефективного та зручного рівня взаємодії з ПК.

Для розробки інформаційної підсистеми було використано Sublime Text 3, що написаний на мові C++ та Python. Sublime Text 3 є повністю безкоштовною.

2.3 Технічне забезпечення розробки інформаційної системи

Комфортна робота калькулятора забезпечена такими інтернет-браузерами:

- Opera;
- GoogleChrome;
- Mozilla Firefox;
- Safari;

Нище наведені вимоги до технічного оснащення ПК для роботи в браузері GoogleChrome:

- Операційна система: Windows 7, Windows 8, Windows 10, Linux, IOS 10
- Оперативна пам'ять від 1 GB;
- Процесор Intel Pentium 4 та вище.
- Жорсткий диск від 512 Mb.

2.4 Розроблення інтерфейсу

При відкритті сторінки ми можемо бачити інтерфейс калькулятора (Рис. 2.1)

Назва

Система координат

УСК-2000 / XYZ

Тип ходу:

розмінутий

Клас точності:

теодолітний хід

на місцевості

збудований

у масштабі:

1:500

Розмірність кутів:

☒ в градусах
☐ в градусах та мінутах
☐ в градусах, мінутах та секундах

Міри ліній:

вимірна лінія

Напрямок руху:

праві кути

Введіть координати першої точки

X1:

Y1:

та

☒ дирекційний кут або ☐ координати наступної точки

°

Введіть координати останньої точки

X1:

Y1:

та

☒ дирекційний кут або ☐ координати наступної точки

°

Додати координати

Введіть наступну сторону ходу та кут

Відстань: м

Кут β: °

Кут нахилу γ: °

Додати

Додати останній кут β

Рис. 2.1. Головна сторінка

Для вводу Назви (Рис. 2.2) прописаний наступний код:

```
<label for=»name»>Назва </label>
```

```
<input name=»name» id=»name» required><br>
```

Назва

Рис. 2.2 Ввід назви

Випадаючі списки (Рис. 2.3) реалізовані за допомогою тегу SELECT:

```
<label for="traverseType">Тип ходу:</label>
<select id="traverseType" name="traverseType"
size="1">
    <option value="1">замкнутий</option>
    <option value="2"
selected="selected">розімкнутий</option>
    <option value="3">з вузловою точкою</option>
    <option value="4">з координатною
привязкою</option>
</select>
```

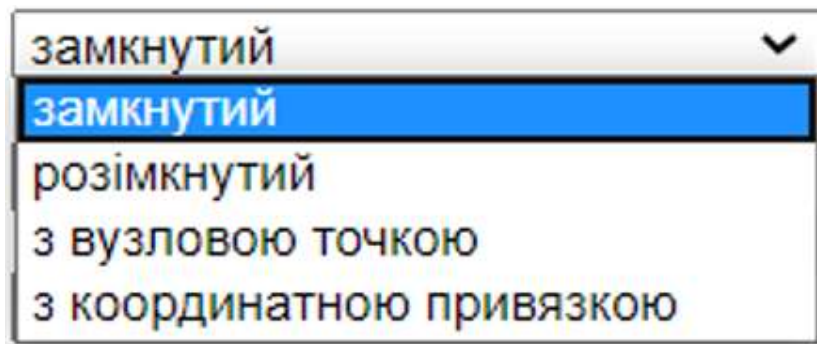


Рис. 2.3. Випадаючий список для вибору типу ходу

Код для вибору розмірності кутів (Рис. 2.4), що працює за допомогою RADIOBUTTON:

```
<p class='degrChouse'><label>Розмірність кутів:</label>
<div>
    <input class="degrBut" type="radio"
name="radioButton" id="radioDegr" value="1" checked>
    <label class="degrButLab" for="radioDegr">в
градусах</label>
</div>
```

```

<div>
    <input class="degrBut" type="radio"
name="radioButton" id="radioMin" value="2">
    <label class="degrButLab" for="radioMin">в
градусах та мінутах</label>
</div>
<div>
    <input class="degrBut" type="radio"
name="radioButton" id="radioSec" value="3">
    <label class="degrButLab" for="radioSec">в
градусах, мінутах та секундах</label>
</div>
</p>

```

Розмірність кутів:

- ☒ в градусах
- ☐ в градусах та мінутах
- ☐ в градусах, мінутах та секундах

Рис. 2.4. Вибір розмірності кутів

Код для введення координат з обмеженням до тисячних (Рис. 2.5):

```

<div id=»secondPointCoordinates» style=»display:none;»
id=»point2»>
    <p><label for=»pointX2»>X2:</label>
        <input type=»number» min=»0» name=»pointX2»
id=»pointX2» step=»0.001»>
    </p>
    <p><label for=»pointY2»>Y2:</label>
        <input type=»number» name=»pointY2» id=»pointY2»
min=»0» step=»0.001»></p>

```

</div>

Введіть координати першої точки

X1:

Y1:

Введите допустимое значение. Ближайшие допустимые значения: 5565,565 и 5565,566.

Рис. 2.5. Обмеження вводу координат

Код для введення та валідація дирекційного кута (Рис. 2.6):

<div id=»cornerBlock»>

<p id='labForCorn' style=»display:

none;»>Дирекційний кут</p>

<input type=»number» class=»cornerDegr»

name=»cornerDegrDirect» id=»cornerDegrDirect» min=»0»

max=»360»

required> <label

for=»cornerDegrDirect»>°</label>

<input type=»number» class=»cornerMin field»

name=»cornerMinDirect» id=»cornerMinDirect» step=»0.001»

style=»display: none;» min=»0» max=»60»>

<label for=»cornerMinDirect» style=»display:

none;» class=»cornerMin»>'</label>

<input type=»number» class=»cornerSec field»

name=»cornerSecDirect» id=»cornerSecDirect» step=»0.001»

style=»display: none;» min=»0» max=»60»>

<label for=»cornerSecDirect» class=»cornerSec»

style=»display: none;»>></label>

</p>

</div>

Дирекційний кут

400 °

Значение должно быть меньше или равно 360.

Рис. 2.6. Обмеження вводу дирекційного кута

Код для кнопки з додаванням координат (Рис. 2.7):

```
<div>
    <button type=»submit» id=»settingsButton»>Додати
координати</button>
</div>
```

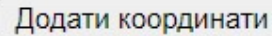


Рис. 2.7. Кнопка для додавання координат

Код для таблиці результатів (Рис. 2.8):

```
<div class=»tableBlock»>
    <table id=»table»>
        <tr id=»titleLine»>
            <th>№</th>
            <th>Кут  $\beta$ ; вим.</th>
            <th> $f_{\beta}$ </th>
            <th>Кут  $\beta$ ; випр.</th>
            <th>Дир кут  $\alpha$ </th>
            <th>d</th>
            <th>L</th>
            <th> $\pm \gamma$ </th>
            <th> $\Delta X$ </th>
            <th> $\Delta Y$ </th>
            <th> $\pm \Delta X$ </th>
            <th> $\pm \Delta Y$ </th>
            <th> $\Delta X$  випр.</th>
            <th> $\Delta Y$  випр.</th>
            <th>X</th>
            <th>Y</th>
        </tr>
```

```

<tr id=»firstLine» class=»firstValues openTrav»>
  <td> БТ1</td>
  <td colspan='13'></td>
  <td id=»firstPointXCoordinate»></td>
  <td id=»firstPointYCoordinate»></td>
</tr>

<tr class=»openTrav»>
  <td colspan='4'></td>
  <td class=»firstValues alphaCorner»
id=»firstAlpha»></td>
  <td colspan=»11»></td>
</tr>

<tr class=»addBeta betaLine openTrav»>
  <td class=»firstValues»>БТ2</td>
  <td></td>
  <td class=»fBeta»></td>
  <td class=»correctedCorner»></td>
  <td colspan=»10»></td>
  <td class=»firstValues xClass»
id=»secondPointXCoordinate»></td>
  <td class=»firstValues yClass»
id=»secondPointYCoordinate»></td>
</tr>

<tr id=»addLastBeta» class=»betaLine»>
  <td class=»firstValues»>БТ3</td>
  <td></td>
  <td class=»fBeta»></td>
  <td class=»correctedCorner»></td>
  <td colspan=»10»></td>

```

```

<td class=»firstValues»
id=»penultPointXCoordinate»></td>
<td class=»firstValues»
id=»penultPointYCoordinate»></td>
</tr>
<tr>
<td colspan='4'></td>
<td class=»firstValues» id=»lastAlpha»></td>
<td colspan='11'></td>
</tr>
<tr id=»lastLine» class=»firstValues»>
<td>Бт4</td>
<td colspan='13'></td>
<td id=»lastPointXCoordinate»></td>
<td id=»lastPointYCoordinate»></td>
</tr>
<tr>
<td>f&beta;доп</td>
<td>&sum;&beta; вим</td>
<td>F&beta;</td>
<td>&sum;&beta; теор</td>
<td></td>
<td>P</td>
<td></td>
<td></td>
<td>&sum;&Delta;X</td>
<td>&sum;&Delta;Y</td>
<td>f&Delta;X</td>
<td>f&Delta;Y</td>
<td>&Delta;X теор</td>

```

```

<td>&Delta;Y теор</td>
<td>f абс</td>
<td>f відн</td>
</tr>
<tr>
<td id=»controlBeta»></td>
<td id=»calcCornSum»></td>
<td id=»fBeta»></td>
<td id=»teorCornSum»></td>
<td></td>
<td id=»perimeter»></td>
<td></td>
<td></td>
<td id=»sumX»></td>
<td id=»sumY»></td>
<td id=»sumMinTeorX»></td>
<td id=»sumMinTeorY»></td>
<td id=»teorSumX»></td>
<td id=»teorSumY»></td>
<td id=»fAbs»></td>
<td id=»fRel»></td>
</tr>
</table>
</div>

```

№	Кут β вим.	fβ	Кут β випр.	Дир кут α	d	L	±γ	X	Y
Бт1									
Бт2									
Бт3									
Бт4									
fдоп	Σβ вим	Fβ	Σβ теор					f абс	f відн

Рис. 2.8. Таблиця результатів
Код функцій підсистеми наведено у Додатку Г.

2.5 Алгоритмічне забезпечення розробки системи

В розроблюваній системі реалізовано ведення, розрахунок та обробку даних. Результат обчислювань заноситься до таблиці результатів.

2.6 Інструкція користувача

При завантаженні підсистеми відкривається сторінка з калькулятором (рис. 2.1). Користувач має можливість ввести дані та виконати відповідні розрахунки.

Назва

Система координат

УСК-2000 / XYZ

Тип ходу:

розмінутий

Клас точності:

теодолітний хід

на місцевості

забудований

у масштабі:

1:500

Розмірність кутів:

☒ в градусах
☐ в градусах та хвилинах
☐ в градусах, хвилинах та секундах

Міри ліній:

вимірювана лінія

Напрямок руху:

праві кути

Введіть координати першої точки

X1:

Y1:

та

☒ дирекційний кут або ☐ координати наступної точки

°

Введіть координати останньої точки

X1:

Y1:

та

☒ дирекційний кут або ☐ координати наступної точки

°

Додати координати

№	Кут β вим.	Fβ	Кут β вимр.	Дир кут α	d	L	±γ	X	Y
Bt1									
Bt2									
Bt3									
Bt4									
Вдоп	Σβ вим	Fβ	Σβ теор					f abs	f відн

Введіть наступну сторону ходу та кут

Відстань: м

Кут β: °

Кут нахилу γ: °

Додати

Додати останній кут β

Рис. 2.1. Головна сторінка

Для роботи у калькуляторі потрібно ввести необхідні дані, відповідно до очікуваного результату. Спочатку треба ввести назву ходу, що розраховується (Рис. 2.2), далі вибрати систему координат, (Рис. 2.3), тип ходу (Рис. 2.4), клас точності (Рис. 2.5), тип місцевості (Рис. 2.6), масштаб (Рис. 2.7), розмірність кутів (Рис. 2.8), міру ліній (Рис. 2.9) та напрямок руху (Рис. 2.10).

Назва

Рис. 2.2. Введення назви ходу

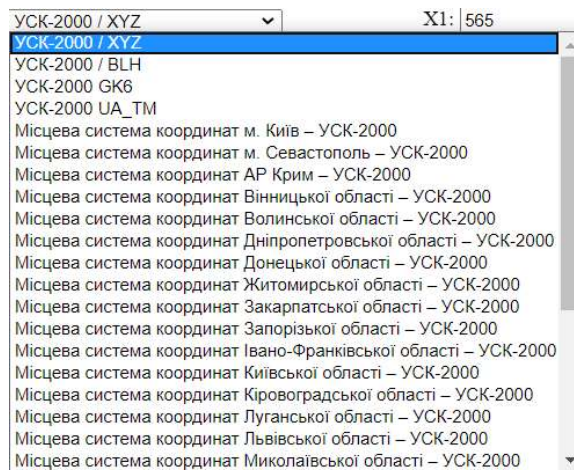


Рис. 2.3. Вибір системи координат

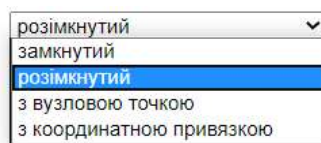


Рис. 2.4. Вибір типу ходу

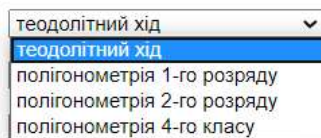


Рис. 2.5. Вибір класу точності

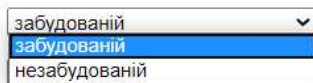


Рис. 2.6. Вибір типу місцевості

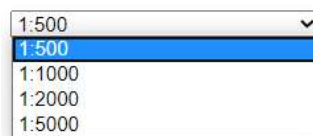


Рис. 2.7. Вибір масштабу

Розмірність кутів:

- ☒ в градусах
- ☐ в градусах та мінутах
- ☐ в градусах, мінутах та секундах

Рис. 2.8. Вибір розмірності кутів

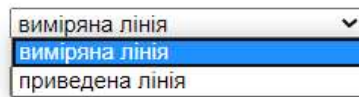


Рис. 2.9. Вибір міри ліній



Рис. 2.10. Вибір напрямку руху

Після, залежно від вибраних позицій, ввести необхідні дані для розрахунку. Якщо вибраний тип ходу «замкнений», то вводимо координати першої точки, дирекційний кут та відстань до точки (Рис. 2.11).

Введіть координати першої точки

X1:

Y1:

Дирекційний кут

°

Відстань до т1:

м

Рис. 2.11. Введення даних для замкнутого ходу

Якщо вибрано тип ходу «розімкнутий» - необхідно ввести координати першої та останньої точок і дирекційний кут або координати наступної точки (Рис. 2.12).

Введіть координати першої точки	Введіть координати останньої точки
X1:	X1:
Y1:	Y1:
та	та
☒ дирекційний кут або ☐ координати наступної точки	☐ дирекційний кут або ☒ координати наступної точки
°	X2:
	Y2:

Рис. 2.12. Введення даних для розімкнутого ходу

Після введення даних натиснути на кнопку «Додати координати» (Рис. 2.13). Дані, що були введені заносяться до таблиці результатів.

Додати координати

Рис. 2.13. Кнопка «Додати координати»

Далі вводимо наступну сторону ходу, вимірний кут та кут нахилу (Рис. 2.14), після чого, для виконання розрахунків натиснути кнопку «Додати» (Рис. 2.15).

Введіть наступну сторону ходу та кут

Відстань: м

Кут β : °

Кут нахилу γ : °

Додати Додати останній кут β

Рис. 2.14. Введення наступної сторони ходу та кутів

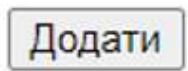


Рис. 2.15. Кнопка «Додати»

Якщо кутова нев'язка більша за допустиму, то на екрані вискочить повідомлення (Рис. 2.16) про закінчення розрахунку.

Подтвердите действие

Кутова нев'язка більша за допустиму, розрахунок зупинено

OK

Рис. 2.16. Повідомлення про закінчення розрахунку

Приклад розрахунків наведений у додатку В.

2.7 Визначення витрат на розроблення системи

2.7.1 Визначення розміру оплати праці

Витрати на розроблення підсистеми складаються з наступних елементів:

- Витрати на заробітну плату
- Витрати на нарахування на заробітну плату
- Амортизаційні витрати
- Загальновиробничі витрати

2.7.2 Визначення розміру витрат на заробітну плату

Оплата праці працівникам розраховується за такою формулою:

$$\text{ФОП} = \text{Ч} * \text{М} * \text{ЗП}, \text{ де}$$

Ч- число виконавців проекту (1 чол.)

М-кількість годин розробки (заплановано виконання завдання за 25 днів

ЗП- розмір заробітної плати за 1 годину, визначається за такою формулою:

Оклад за місяць / кількість робочих днів у місяць/тривалість робочого дня

Визначено що у розробці підсистеми буде приймати участь 1 фахівець протягом 25 днів, що в перерахунку на години складає $25 \cdot 8 = 200$.

Оклад виконавця становить 15 000 грн.

Розмір заробітної плати за 1 годину становить $15000/20/8=93,75$ грн

Враховуючи ці складові, оплата праці працівникам, що були задіяні в розробці підсистеми становить :

$ФОП = 1 \cdot 200 \cdot 93,75 = 18750$ грн.

2.7.3 Визначення розміру витрат на нарахування на заробітну плату

Відповідно до закону України розмір нарахувань Єдиного соціального внеску (ЄСВ) становить 22% від фонду оплати праці (ФОП).

Враховуючи ці складові, витрати на нарахування на заробітну плату становлять :

$ЄСВ = ФОП \cdot 22\% = 18750 \cdot 22\% = 4125$ грн.

2.7.4 Визначення розміру амортизаційних витрат

Балансова вартість ПК розраховується за такою формулою:

$Ц(РС) = Ц_p \cdot (1 + K_{ун}) = 27\,000 \cdot (1 + 0.2) = 32\,400$ грн

$Ц_p$ – ринкова ціна ПК

$K_{ун}$ – коефіцієнт витрат на установку і налагодження ПК

Амортизаційні відрахування використання ПК обчислюються за формулою:

$$Z(A) = Ц (PC) / H_a = 32\,400 / 36 = 900 \text{ грн}$$

H_a – норма амортизаційних відрахувань

2.7.5 Визначення розміру загальновиробничих витрат

Загальновиробничі витрати – це витрати, що несе підприємство для забезпечення повноцінного виробничого процесу, крім заробітної плати, нарахування ЄСВ та амортизації. Загальновиробничі витрати підприємства на місяць наведені у Таблиці 2.8.5.1.

№ п/п	Елементи витрат	Сума, грн
1	Витрати на орендну плату	7000
2	Витрати на комунальні послуги	108
3	Непрямі витрати	1555,2
	РАЗОМ	8663,2

Таблиця 2.8.5.1. Загальновиробничі витрати підприємства на місяць

Витрати на електроенергію:

$$Z \text{ €} = P(PC) * T'(PC) * C\text{€} * A = 2.2 * 40 * 1.8 * 0.68 = 108$$

$P(PC)$ – потужність ПК

$C\text{€}$ – вартість 1 кВт електроенергії – 1.8 грн/кВт

A – коефіцієнт інтенсивного використання

$$\text{Непрямі витрати } Z(M) = 38\,880 * 0.04 = 1\,555,2$$

Визначено, що на час розробки підсистеми загальновиробничі витрати дорівнюють 10829 грн (8663,2/20/8*200).

Загальні витрати на розробку підсистеми наведені у Таблиці 2.8.5.2.

№ п/п	Елементи витрат	Сума, грн
1	Витрати на заробітну плату	18750
2	Витрати на нарахування на заробітну плату	4125
3	Амортизаційні витрати	900
4	Загальновиробничі витрати	10829
	РАЗОМ	34604

Таблиця 2.8.5.2. Загальні витрати на розробку підсистеми

РОЗДІЛ 3. ОРГАНІЗАЦІЯ ПРАЦІ

Діяльність працівників Комунального підприємства «Київський інститут земельних відносин» регламентується посадовими інструкціями, що розробляються згідно з галузевими випусками Довідника кваліфікаційних характеристик професій працівників (далі – ДКХП):

- випуск 1 «Професії працівників, що є загальними для всіх видів економічної діяльності»;
- випуск 74 «Геологорозвідка та топографо-геодезичні роботи».

Головною метою організації праці є створення необхідних умов для правильного та кваліфікаційного розвитку земельних відносин, ефективного використання земельних ресурсів, збільшення їх ролі в соціально-економічному розвитку міста, в виконанні науково-дослідних, дослідно-конструкторських, дослідно-технологічних, спеціальних та інших робіт в галузі ведення державного земельного кадастру, землеустрою та моніторингу земель з використанням новітніх інформаційних технологій, виконання Міської цільової програми використання та охорони земель міста Києва на 2019-2021 роки, затвердженої рішенням Київської міської ради від 04.12.2018 № 229/6280. Нормативи чисельності орієнтовано на комплекс організаційно-технічних умов, які забезпечують нормальну роботу, і має відповідати таким умовам:

- забезпечення своєчасної сплати податків і зборів (обов'язкових платежів) до бюджетів та до державних цільових фондів згідно з чинним законодавством України;
- забезпечення цільового використання комунального майна та коштів;
- забезпечення здійснення робіт, виконання доручень Засновника та Управління, надавання послуг згідно з предметом

діяльності;

- виконання Міської цільової програми використання та охорони земель міста Києва на 2019-2021 роки, затвердженої рішенням Київської міської ради від 04.12.2018 № 229/6280;

- мінімум витрат часу в разі обміну між співробітниками необхідною інформацією;

- вільний підхід до кожного робочого місця, засобів технічного оснащення;

- можливість установаження додаткового обладнання;

- додержання санітарно-гігієнічних норм і правил;

- додержання правил охорони праці та протипожежного захисту;

- додержання режиму праці та відпочинку.

Під час організації робочих місць рекомендується дотримуватися норм щодо рівнів освітленості, температури, вологості, чистоти повітряного середовища та шуму. На робочому місці працівника повинен бути необхідний набір інструментів, пристроїв, засобів індивідуального захисту, інвентар, що відповідає умовам технічної експлуатації та вимогам техніки безпеки.

Велике значення в оснащенні робочих місць має встановлення меблів з уніфікованих елементів, які обираються з урахуванням посадових обов'язків і характеру роботи кожного працівника. Із таких елементів можливо монтувати різноманітні робочі столи, приставки до них, тумби, шафи.

Колір меблів і обладнання повинні гармонувати із загальним фоном приміщення. Кольорове оформлення службових приміщень повинно забезпечувати стабільну працездатність виконавців протягом усього дня.

Озеленення приміщень, що є декоративним елементом інтер'єру, поліпшує склад повітря, зменшує нервово-психічну

стомленість, що особливо важливо для умов розумової праці.

Ергономічні принципи організації праці в структурних підрозділах КП «КІЗВ» мають забезпечити сучасні умови праці. Умови праці – це сукупність чинників виробничого середовища, які впливають на здоров'я та працездатність людини під час роботи. Основними чинниками організації праці в структурних підрозділах КП «КІЗВ» є: планування приміщень, освітлення приміщень і робочих місць, мікроклімат, шум, колір приміщень, режим праці та відпочинку.

На сьогодні основним засобом підготовки документів у разі здійснення робіт з формування та оброблення документації у підрозділах КП «КІЗВ» є персональний комп'ютер (далі – ПК).

Робочі місця працівників КП «КІЗВ» мають бути зручними й обладнаними відповідно до гігієнічних норм, їх оснащують одно- (дво-) тумбовими столами, що забезпечує належне розміщення на них оргтехніки та предметів праці, а у шухлядах столів – зберігання відповідних документів і канцелярських приладь, підйимально-обертальними стільцями, лотками для паперу, малою оргтехнікою тощо.

Відстань між очима працюючого та екраном ПК має бути не меншою, ніж півтори довжини діагоналі використовуваного монітора; дисплей слід розташовувати нижче рівня очей на 15 градусів, а клавіатуру – на відстані близько сорока сантиметрів від тулуба (Рис. 3.1).

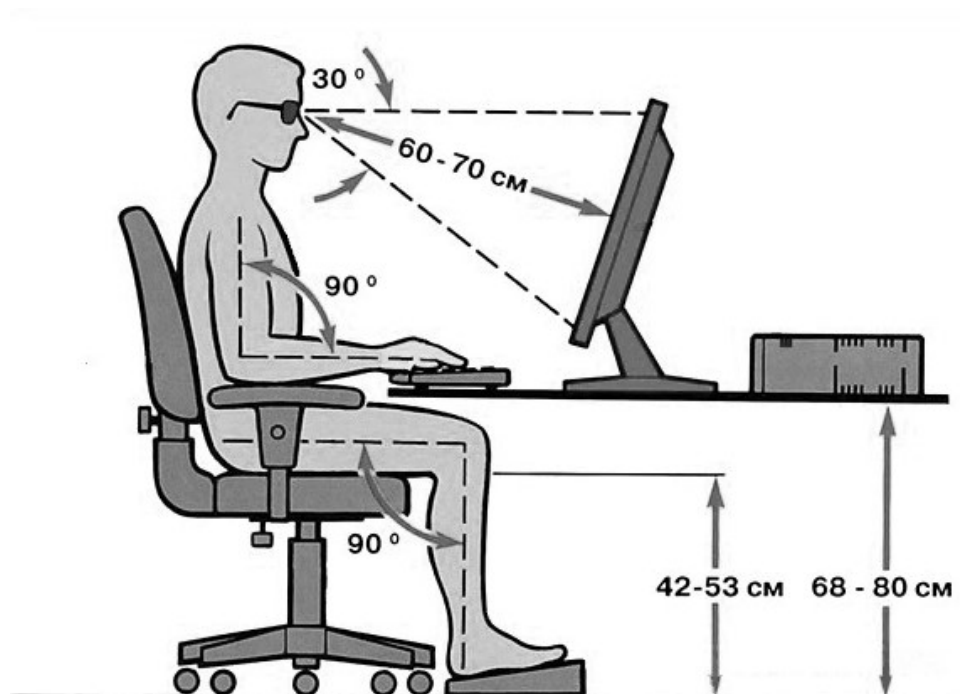


Рис. 3.1. Правильне розташування працівника за комп'ютером

Зап'ястя повинні розташовуватися на одному рівні з передпліччями. Екран дисплея повинен розташовуватися таким чином, щоб сонячне проміння і світло ламп не відбивались від нього та не осліплювали працюючого. В ідеалі рівень яскравості екрана повинен відповідати яскравості світла в приміщенні, чого можна домогтися за допомогою налагодження монітора.

Надзвичайно негативно на очі впливає мигання екрана монітора. Мінімальна частота оновлення екрана, згідно з вимогами ергономіки, має становити не менше 75 Гц.

Службові приміщення повинні мати прямокутну або квадратну форму. Найбільш раціональним є приміщення прямокутної форми із співвідношенням сторін 1:1,5 (але не більш 1:2). Ширина приміщення має бути не менше 2,5 м, а висота – 3,25 м.

Особливе значення має освітлення робочого місця. Якщо на робочому місці є місцеве освітлення, то рекомендується застосовувати поворотні, пересувні лампи. М'яке світло, що не сліпить очі, повинно

надходити до поверхні робочого стола ліворуч. Недопустиме застосування лише місцевого освітлення, оскільки це викликає неминучу адаптацію зору. Рекомендовано такі норми освітлення службових приміщень і робочих місць:

- у разі люмінесцентного освітлення – 300 люкс;
- у разі освітлення лампами розжарювання – 150 люкс.

У разі одностороннього освітлення рекомендована ширина приміщення не повинна перевищувати 7 м, двостороннього – 15 м.

Нормальний рівень насичення приміщень світлом при коефіцієнтах відблиску поверхонь: підлоги – 25%, стелі – 70%, меблі та устаткування – не менше 35%. Захистом від прямого сонячного проміння може бути сонцезахисний козирок, жалюзі, штори та подібне.

Ефективність освітлення в службових приміщеннях підвищується в разі раціонального фарбування стін, стель. На ефективність праці суттєво впливає колір приміщення. Найсприятливішими кольорами є білий, зелено-блакитний, жовто-зелений, світло-зелений, бежевий, світло-бежевий світло-жовтий, світло-блакитний. Ці кольори не тільки заспокійливо діють на нервову систему людини, але і значно поліпшують освітленість у приміщеннях.

Для кращого розсіювання та відбиття світла рекомендується фарбувати стелю в білий колір, він відбиває до 90% світлового потоку. Стіни повинні менше відбивати світло та мають бути темнішого кольору, ніж стеля. Якщо вікна в приміщенні розташовано на сонячний бік, то стіни краще фарбувати в прохолодні тони: блакитний, зеленуватий, салатний. Для приміщень з вікнами, орієнтованими на північ, північний захід, північний схід більше підходять кольори теплої гами: жовтуватий, світло-оранжевий, піщаний.

Підлога, незалежно від матеріалу, з якого її зроблено, повинна мати темніший колір, ніж стіни.

Здоров'я та працездатність робітника залежить також від чинників навколишнього середовища, у першу чергу – від підтримання рекомендованої температури та вологості повітря. Слід передбачити в КП КІЗВ такі норми температури, відносної вологості та швидкості руху повітря в робочих приміщеннях:

➤ температура повітря:

- у холодний період року (нижче $+10^{\circ}\text{C}$), температура в приміщенні в межах від 20 до 22°C ;

- у теплий період року (вище $+10^{\circ}\text{C}$) – $20-25^{\circ}\text{C}$;

➤ відносна вологість повітря – 30-60%;

➤ швидкість руху повітря:

- у холодний період року – не більше – 0,2 м/с;

- у теплий період року – 0,2-0,5 м/с;

➤ атмосферний тиск – від 84 до 107 Кпа (630-800 мм. Рт.ст.);

➤ максимальна запиленість повітря в приміщеннях при розмірі часток не більше 3 мкм – 0,75 мг/куб. м.

Озеленення приміщень є декоративним елементом інтер'єру, воно поліпшує склад повітря, зменшує нервово-психічне та зорове стомлення, що особливо важливо для умов розумової праці.

Під час проектування робочих місць має бути передбачено комплексне оснащення необхідними засобами праці. Оснащення робочих місць необхідними технічними засобами, як правило, передбачається під час проектування адміністративних будівель та їх реконструкції.

Для підвищення працездатності і зменшення стомлення рекомендується встановити регламентовані перерви, які відновлюють працездатність.

Для працівників, які вводять інформацію в комп'ютер, передбачається перерва через 2 години після початку роботи і через 1,5-2 години після

перерви на обід тривалістю по 15 хвилин кожна або через кожную годину роботи за комп'ютером по 10 хвилин протягом робочого дня.

Шухляди столу слід обладнати пристосуваннями для зручного розташування папок. Бажано, щоб на кожній полиці містилася картка із переліком вмісту папок.

Для зберігання спеціальної, довідкової літератури та документів має бути шафа.

Працівники КП КІЗВ повинні забезпечуватися необхідними довідковими матеріалами, у тому числі нормативно-методичними документами, списками працівників організації та її структурних підрозділів з номерами телефонів і адрес, абетковими книжками адрес і телефонів організацій, пов'язаних із діяльністю організації, картотеками тощо.

ВИСНОВОК

Виконанням цієї кваліфікаційної роботи є розробка та створення інформаційної підсистеми призначеної для забезпечення точних розрахунків при виконанні геодезичних робіт з метою унеможливлення викривлення даних про земельну ділянку, підвищення ефективності роботи персоналу та автоматизації процесу розрахунків теодолітних точок.

При моделюванні предметної області на основі методології системного аналізу та проектування за допомогою CASE-засобу AllFusion Process Modeler було створено функціональну та концептуальну моделі процесу. В результаті аналізу прийнято рішення про доцільність розроблення цієї підсистеми власними силами, задачею якої є забезпечення точності розрахунків та єдиної методології.

Розроблено підсистему у вигляді калькулятора, що буде розміщена на офіційному веб-порталі та забезпечить точність розрахунків. Враховуючи площу м. Києва (83,6 тис. га) та кількість земельних ділянок очікується, що попит на користування калькулятором буде значним, і в свою чергу популяризує на ринку професійних послуг та підтверджує передове становище підприємства на ринку геодезичних послуг.

Питання охорони праці при експлуатації комп'ютерної техніки також були розглянуті у кваліфікаційній роботі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://mgm.kyivland.gov.ua/>
2. <https://kizv.gov.ua/>
3. Войтенко С.П. Інженерна геодезія: підручник // С.П. Войтенко. – К: Знання, 2009.
4. Російсько-український тлумачний словник основних термінів та понять з геодезії / Укл. Новицький В.В.- Харків : ХІІМГ, 1993.
5. ДСТУ 2756-94. Геодезія. Терміни та визначення. К.: Держстандарт України, 1994.
6. ДСТУ 2757-94. Картографія. Терміни та визначення. К.: Держстандарт України, 1994.
7. Підручник. Частина 2 / А. Л. Островський, О. І. Мороз, В. Л. Тарнавський. Друге вид., виправлене. Львів: Видавництво Львівської політехніки, 2012.
8. Романчук С.В. Геодезія. Навчальний посібник / С.В. Романчук, В.П. Кирилюк, М.В. Шемякін. – Умань: Уманський ДАУ, 2008.
9. Порицький Г.О. Геодезія: Підручник / Г.О. Порицький, Б.І. Новак, Л.П. Рафальська. – К. : „Арістей”, 2007.

ДОДАТКИ

Додаток А «Функціональна модель (AS-IS)»

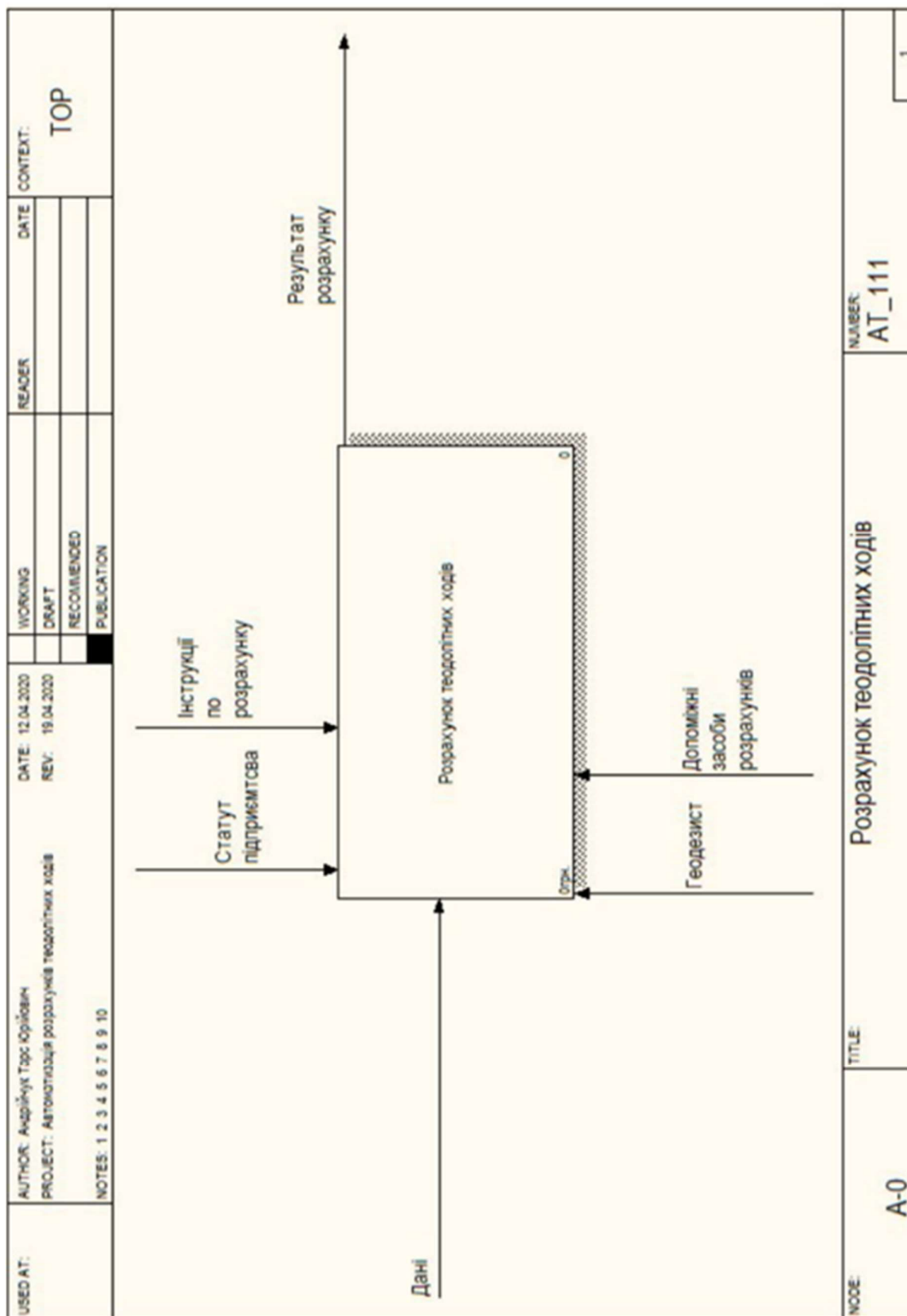


Рис. А. Функціональна схема моделі AS-IS

Додаток Б «Концептуальна модель (ТО-ВЕ)»

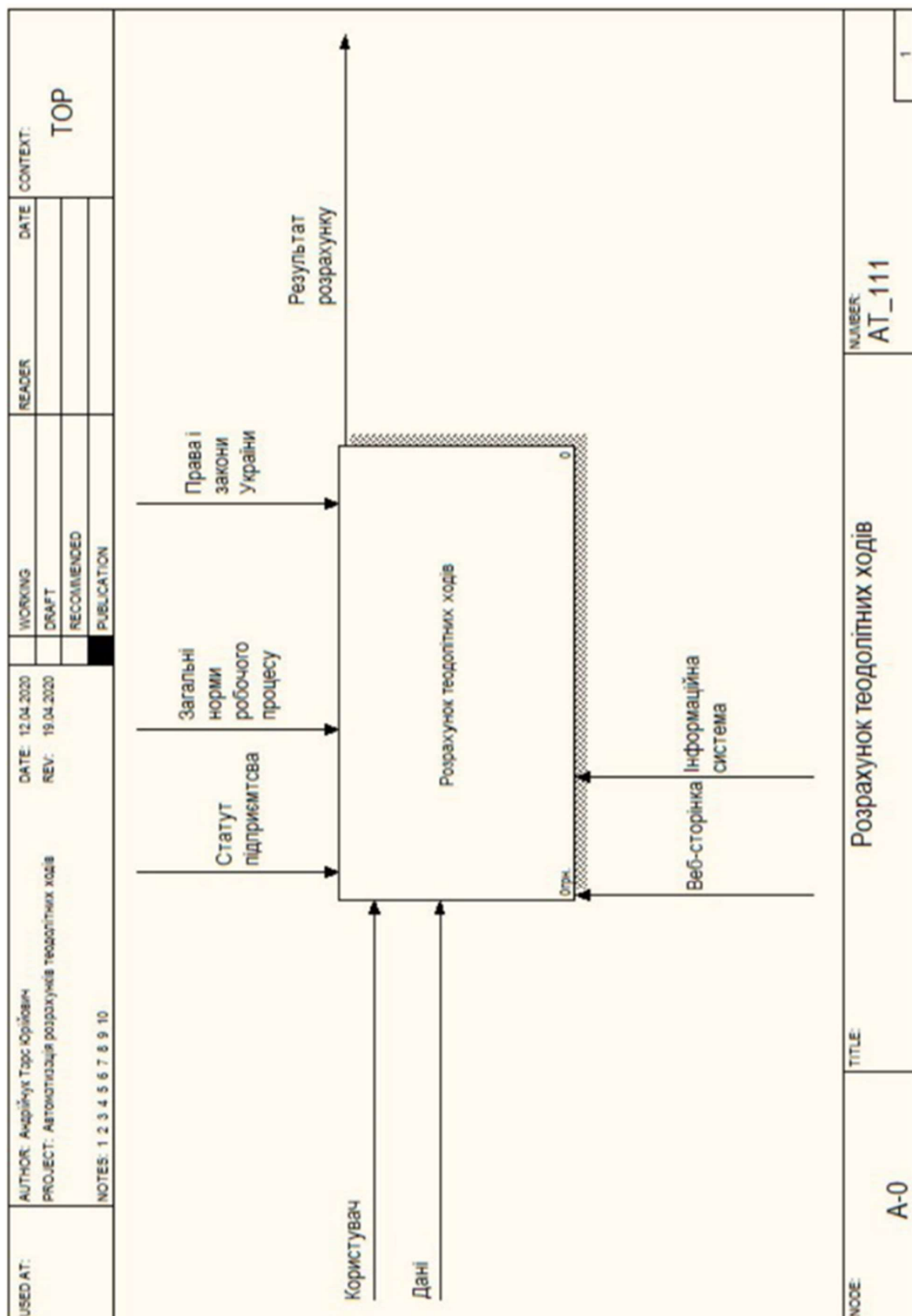


Рис. Б. Концептуальна схема моделі ТО-ВЕ

Додаток В «Приклад розрахунків»

Назва

Система координат

Тип ходу:

Клас точності:

на місцевості

у масштабі :

Розмірність кутів:

Міри ліній:

Напрямок руху:

Назва

УСК-2000 / XYZ

розиментий

теодолітний хід

забудованій

1:500

в градусах

в градусах та минутах

в градусах, минутах та секундах

вирішена ліній

праві кути

Введіть координати першої точки

Введіть координати останньої точки

X1: 8569

Y1: 8639

та

дирекційний кут або координати наступної точки

X1: 5699

Y1: 5328

та

дирекційний кут або координати наступної точки

125°

52°

Додати координати

№	Кут в вим.	φ	Кут в вимр.	Дир кут α	d	L	Δγ	X	Y
Bг2	52°			125°				8569	8639
τ1	123°				414.73	424.00	12°		
τ2	98°				787.03	855.00	23°		
Bг3	87°				629.25	742.00	32°	5699	5328
φвон	Σβ вим	φβ	Σβ теор	52°				f абе	f вян
0°2'	360°	-433°	793°						

Рис. В.1 Приклад розрахунків

Name

УСК-2000 / XYZ

розмінутий

теодолітний Хід

збудований

1:500

Введіть координати першої точки

X1: 135

Y1: 513

та

Введіть координати останньої точки

X1: 546

Y1: 846

Клас точності:

на місцевості

у масштабі :

1:500

Введіть координати першої точки

X1: 135

Y1: 513

та

Введіть координати останньої точки

X1: 546

Y1: 846

Розмірність кутів:

в градусах

в градусах та хвилинах

в градусах, хвилинах та секундах

Міри ліній:

виміряна лінія

прямі кути

Додати координати

№	Кут в вим.	Ф	Кут в випр.	Дир кут α	d	L	≠γ	X	Y
Бг2	135°5'16"			35°33'51"				135	513
t1	96°56'18"				839.56	863.00	13°23'22"		
t2	96°6'26"				312.35	336.00	21°37'34"		
Бг3	30°56'20"				894.85	896.00	2°53'52"	548	846
Фдоп	Σβ вим	Фβ	Σβ теор	63°21'33"				f аос	f відлн
0°2'	360°	-332°12'18"	692°12'18"						

Рис. В.2 Приклад розрахунків

54

ДОДАТОК Г «КОД ФУНКЦІОНАЛУ ПІДСИСТЕМИ (JAVASCRIPT)»

```
let pointX2 = document.getElementById('pointX2');
let pointY2 = document.getElementById('pointY2');
let lastPointX2 = document.getElementById('lastPointX2');
let lastPointY2 = document.getElementById('lastPointY2');
let pointPenultX1 = document.getElementById('pointPenultX1');
let pointPenultY1 = document.getElementById('pointPenultY1');
let cornerDegrDirect = document.getElementById('cornerDegrDirect');
let cornerMinDirect = document.getElementById('cornerMinDirect');
let cornerSecDirect = document.getElementById('cornerSecDirect');
let lastCornerDegrDirect =
document.getElementById('lastCornerDegrDirect');
let lastCornerMinDirect =
document.getElementById('lastCornerMinDirect');
let lastCornerSecDirect =
document.getElementById('lastCornerSecDirect');

let corner = document.getElementById('cornerBlock');
let lastCorner = document.getElementById('lastCornerBlock');
let cornerGamma = document.getElementById('cornerGamma');
let gammaCornerDegr = document.getElementById('gammaCornerDegr');
let secondPointCoordinates =
document.getElementById('secondPointCoordinates');
let lastPointCoordinates =
document.getElementById('lastPointCoordinates');

let form = document.getElementById('settingsBlock');
let secondForm = document.getElementById('sideOfTraverse');
let firstLengthVal = document.getElementById('firstLengthVal');

let fields, secondFormFields;
let minFields = document.querySelectorAll('.cornerMin');
let secFields = document.querySelectorAll('.cornerSec');
let table = document.getElementById('table');
let rows = table.getElementsByTagName('tr');

let alpha = document.getElementById('firstAlpha');
let lastAlpha = document.getElementById('lastAlpha');
let fBetaSum = document.getElementById('fBeta');
document.addEventListener('click', cornerMin);
```

```

if (form) {
    form.addEventListener('click', openOrCloseTraverse);
    form.addEventListener('click', selectTeoStep);
    form.addEventListener('click', selectCornerOrPoints);
    form.addEventListener('click', selectLastCornerOrPoints);
    form.addEventListener('click', lengthOrDistance);
}

function cornerMin() {
    if (document.getElementById('radioMin').checked) {
        minFields.forEach(element => {
            element.style.display = 'inline-block'
        })
        secFields.forEach(element => {
            element.style.display = "none"
        })
    } else if (document.getElementById('radioSec').checked) {
        minFields.forEach(element => {
            element.style.display = 'inline-block'
        })
        secFields.forEach(element => {
            element.style.display = "inline-block"
        })
    } else {
        minFields.forEach(element => {
            element.style.display = 'none'
        })
        secFields.forEach(element => {
            element.style.display = "none"
        })
    }
}

function openOrCloseTraverse() {
    let traverseType = document.getElementById('traverseType').value;
    if (traverseType == '1') {
        document.getElementById('lastCoordinates').style.display =
'none';
        document.getElementById('selectPointOrCorner').style.display =
'none';
        document.getElementById('labForCorn').style.display = 'block';
    }
}

```

```

        document.getElementById('firstLength').style.display = 'block';
        pointPenultX1.required = false;
        pointPenultY1.required = false;
        lastCornerDegrDirect.required = false;
        firstLengthVal.required = true;
    } else {
        document.getElementById('lastCoordinates').style.display =
'block';
        document.getElementById('selectPointOrCorner').style.display =
'block';
        document.getElementById('labForCorn').style.display = 'none';
        document.getElementById('firstLength').style.display = 'none';
        pointPenultX1.required = true;
        pointPenultY1.required = true;
        lastCornerDegrDirect.required = true;
        firstLengthVal.required = false;
    }
}

```

```

function selectCornerOrPoints() {
    let traverseType = document.getElementById('traverseType').value;
    if (document.getElementById("corner").checked) {
        cornerBlock.style.display = 'block';
        secondPointCoordinates.style.display = 'none';
        pointX2.required = false;
        pointY2.required = false;
        pointX2.value = '';
        pointY2.value = '';
        pointX2.classList.remove("invalid");
        pointY2.classList.remove("invalid");
        cornerDegrDirect.required = true;
    } else {
        if (traverseType !== '1') {
            cornerBlock.style.display = 'none';
            secondPointCoordinates.style.display = 'block';
            cornerDegrDirect.required = false;
            pointX2.required = true;
            pointY2.required = true;
            cornerDegrDirect.classList.remove("invalid");
            cornerDegrDirect.value = '';
            if (cornerMinDirect.value) {
                cornerMinDirect.value = ''
            }
        }
    }
}

```

```

    };
    if (cornerSecDirect.value) {
        cornerSecDirect.value = '';
    };
} else {
    cornerBlock.style.display = 'block';
    secondPointCoordinates.style.display = 'none';
    cornerDegrDirect.required = true;
    pointX2.required = false;
    pointY2.required = false;
}
}
}

function selectLastCornerOrPoints() {
    let traverseType = document.getElementById('traverseType').value;
    if (document.getElementById("lastCorner").checked) {
        lastCornerBlock.style.display = 'block';
        lastPointCoordinates.style.display = 'none';
        lastPointX2.required = false;
        lastPointY2.required = false;
        if (traverseType !== "1") {
            lastCornerDegrDirect.required = true;
        } else {
            lastCornerDegrDirect.required = false;
        }
        lastPointX2.value = '';
        lastPointY2.value = '';
        lastPointX2.classList.remove("invalid");
        lastPointY2.classList.remove("invalid");
    } else {
        lastCornerBlock.style.display = 'none';
        lastPointCoordinates.style.display = 'block';
        lastCornerDegrDirect.required = false;
        lastCornerDegrDirect.classList.remove("invalid");
        lastPointX2.required = true;
        lastPointY2.required = true;
        lastCornerDegrDirect.value = '';
        if (lastCornerMinDirect.value) {
            lastCornerMinDirect.value = '';
        };
        if (lastCornerSecDirect.value) {
            lastCornerSecDirect.value = ''

```

```

    };
  }
}

function selectTeoStep() {
  if (document.getElementById("teoStep").selected) {
    document.getElementById('builtUpBlock').style.display = 'block';
  } else {
    document.getElementById('builtUpBlock').style.display = 'none';
  }
}

function lengthOrDistance() {
  if (document.getElementById("measureDistance").selected) {
    cornerGamma.style.display = 'block';
    gammaCornerDegr.required = true;
  } else {
    cornerGamma.style.display = 'none';
    gammaCornerDegr.required = false;
  }
}

fields = form.querySelectorAll('input');

secondFormFields = secondForm.querySelectorAll('input')

form.addEventListener('submit', (e) => {
  e.preventDefault();
  addSettings();
});

const formToJSON = elements => [].reduce.call(elements, (data,
element) => {
  data[element.name] = element.value;
  return data;
}, {});

function addSettings() {
  validationFunc(fields);
  if (form.querySelectorAll('.invalid').length == 0) {
    addToTable();
  }
}

```

```

document.getElementById('sideOfTraverseButton').removeAttribute("disabled");
    document.getElementById('settingsButton').disabled = true;
}
}

function addToTable() {
    let degr = form.cornerDegrDirect.value;
    let min = form.cornerMinDirect.value;
    let sec = form.cornerSecDirect.value;

    let lastDegr = form.lastCornerDegrDirect.value;
    let lastMin = form.lastCornerMinDirect.value;
    let lastSec = form.lastCornerSecDirect.value;

    let X1 = form.pointX1.value;
    let Y1 = form.pointY1.value;
    let X2 = form.pointX2.value;
    let Y2 = form.pointY2.value;

    let lastX1 = form.pointPenultX1.value;
    let lastY1 = form.pointPenultY1.value;
    let lastX2 = form.lastPointX2.value;
    let lastY2 = form.lastPointY2.value;
    let length = form.firstLengthVal.value;
    let traverseType = document.getElementById('traverseType').value;
    let title = document.getElementById('titleLine');
    if (traverseType !== '1') {

        if (document.getElementById("corner").checked) {
            document.getElementById('firstLine').style.display = 'none';
            document.getElementById('secondPointXCoordinate').innerHTML =
X1;
            document.getElementById('secondPointYCoordinate').innerHTML =
Y1;

            writeDegr(degr, min, sec, alpha)
        } else {
            document.getElementById('firstLine').style.display = '';
            document.getElementById('firstPointXCoordinate').innerHTML =
X1;

```

```

        document.getElementById('firstPointYCoordinate').innerHTML =
Y1;
        document.getElementById('secondPointXCoordinate').innerHTML =
X2;
        document.getElementById('secondPointYCoordinate').innerHTML =
Y2;

        alpha.innerHTML = countDegr(X1, Y1, X2, Y2);
    }

    document.getElementById('penultPointXCoordinate').innerHTML =
lastX1;
    document.getElementById('penultPointYCoordinate').innerHTML =
lastY1;

    if (document.getElementById("lastCorner").checked) {
        document.getElementById('lastLine').style.display = 'none';
        writeDegr(lastDegr, lastMin, lastSec, lastAlpha)
    } else {
        document.getElementById('lastLine').style.display = '';
        document.getElementById('lastPointXCoordinate').innerHTML =
lastX2;
        document.getElementById('lastPointYCoordinate').innerHTML =
lastY2;

        lastAlpha.innerHTML = countDegr(lastX1, lastY1, lastX2,
lastY2);
    }
    } else {
        let openTravs = document.querySelectorAll('.openTrav');
        openTravs.forEach(elem => {

            if (elem.parentNode) {
                elem.parentNode.removeChild(elem);
            }
        })
        document.getElementById('lastLine').style.display = 'none';
        title.insertAdjacentHTML('afterend', '<tr class="addBeta"><td
class="firstValues">Bt2</td><td colspan="13"></td><td
class="firstValues xClass" id="secondPointXCoordinate"></td><td
class="firstValues yClass" id="secondPointYCoordinate"></td></tr>');

        document.getElementById('secondPointXCoordinate').innerHTML = X1;

```

```

        document.getElementById('secondPointYCoordinate').innerHTML = Y1;
        let bet = document.querySelector('.addBeta');
        bet.insertAdjacentHTML('afterend', '<tr class="afterBeta" ><td
colspan="4"></td><td class="firstValues alphaCorner"
id="firstAlpha"></td><td class="measureLength"><td
class="measureDistance"></td><td class="gammaCorner"></td><td
></td><td></td><td class="fX"></td><td
class="fY"></td><td></td><td></td><td colspan="2"></td></tr>');
        let alpha = document.getElementById('firstAlpha');
        let leng = document.querySelector('.measureLength');
        leng.innerHTML = length;
        writeDegr(degr, min, sec, alpha);
        writeDegr(degr, min, sec, lastAlpha);
        document.getElementById('penultPointXCoordinate').innerHTML = X1;
        document.getElementById('penultPointYCoordinate').innerHTML = Y1;
        let lastBetaName =
document.getElementById('addLastBeta').cells[0];
        lastBetaName.innerHTML = 'Bn2';
        let afterBet = document.querySelector('.afterBeta');
        afterBet.insertAdjacentHTML('afterend', '<tr class="addBeta
betaLine"><td>121</td><td></td><td class="fBeta"></td><td
class="correctedCorner"></td><td colspan="10"></td><td
class="xClass"></td><td class="yClass"></td></tr>')
    }
}

function writeDegr(deg, min, sec, value) {
    if (deg !== '' && min !== '' && sec !== '') {
        value.innerHTML = deg + '&deg;' + min + "'" + sec + "'";
    };
    if (deg !== '' && min !== '' && sec == '') {
        value.innerHTML = deg + '&deg;' + min + "'";
    };
    if (deg !== '' && min == '' && sec == '') {
        value.innerHTML = deg + '&deg;';
    };
}

function countDegr(x, y, x2, y2) {

    deltx = Number(x2) - Number(x);
    deltY = Number(y2) - Number(y);

```

```

radians = Math.atan(deltY / deltax);
degrees = Math.abs(radians * 180 / Math.PI);

if (deltX.toString()[0] !== '-' && deltY.toString()[0] !== '-') {
  cornerDegr = degrees;
} else if (deltX.toString()[0] == '-' && deltY.toString()[0] !== '-') {
  cornerDegr = 180 - degrees;
} else if (deltX.toString()[0] == '-' && deltY.toString()[0] == '-') {
  cornerDegr = 180 + degrees;
} else if (deltX.toString()[0] !== '-' && deltY.toString()[0] == '-') {
  cornerDegr = 360 - degrees;
}

if (cornerDegr.toString().split('.')[1]) {
  let afterPoint = Number('0' + '.' +
(cornerDegr.toString().split('.')[1])) * 60;
  cornerMin = afterPoint.toString().split('.')[0];
  let afterMinPoint = Number('0' + '.' +
afterPoint.toString().split('.')[1]) * 60;

  cornerSec = Math.round(afterMinPoint)
  cornerDegr = cornerDegr.toString().split('.')[0] + '&deg;' +
cornerMin + "" + cornerSec + ""
} else {
  cornerDegr = cornerDegr + '&deg;';
}

length = (Math.sqrt(Math.pow(deltX, 2) + Math.pow(deltY, 2)));
return cornerDegr;
}

function validationFunc(inp) {
  inp.forEach(element => {

    if (element.required == true && element.id !== 'length') {
      if (element.value.length == 0 || element.value < 0) {
        element.classList.add('invalid');
      }
      element.addEventListener('keyup', function () {
        if (element.value.length == 0 || element.value < 0) {
          element.classList.add('invalid');
        }
      });
    }
  });
}

```

```

        } else if (element.value >= 0) {
            element.classList.remove('invalid');
        }
    })
}

}))

document.getElementById('sideOfTraverseButton').addEventListener('click', function () {
    document.getElementById('length').required = true;
    if (document.getElementById("measureDistance").selected) {
        gammaCornerDegr.required = true;
    } else {
        gammaCornerDegr.required = false;
    }
})

document.getElementById('lastBetaCorner').addEventListener('click', function () {
    document.getElementById('length').required = false;
    gammaCornerDegr.required = false;
})

secondForm.addEventListener('keypress', (e) => {
    if (e.keyCode == 13) {
        e.preventDefault();
    }
})

let btn1 = document.getElementById('sideOfTraverseButton');

if (btn1 != undefined) {
    btn1.addEventListener('click', (e) => {
        e.preventDefault();
        add();
    });
}

let btn2 = document.getElementById('lastBetaCorner');

```

```

if (btn2 != undefined) {
    btn2.addEventListener('click', (e) => {
        e.preventDefault();
        document.getElementById('length').classList.remove("invalid");

document.getElementById('gammaCornerDegr').classList.remove("invalid"
);
        addLastCorner();
    });
}

function addLastCorner() {
    validationFunc(secondFormFields);
    if (secondForm.querySelectorAll('.invalid').length == 0) {
        //secondForm.action = '#';
        // secondForm.method = 'GET'
        let degr = secondForm.betaCornerDegr.value;
        let min = secondForm.betaCornerMin.value;
        let sec = secondForm.betaCornerSec.value;

        let lastBeta = document.getElementById('addLastBeta').cells[1];
        writeDegr(degr, min, sec, lastBeta)
        document.getElementById('sideOfTraverseButton').disabled = true;
        document.getElementById('lastBetaCorner').disabled = true;
        secondForm.reset();
        secondForm.style.display = "none";
        calcMeasureCorners();
        calcTeorCorners();
    }
}

function add() {
    validationFunc(secondFormFields);
    lengthLimitation();
    let degr = secondForm.betaCornerDegr.value;
    let min = secondForm.betaCornerMin.value;
    let sec = secondForm.betaCornerSec.value;

    if (secondForm.querySelectorAll('.invalid').length == 0) {
        //secondForm.action = '#';
        // secondForm.method = 'GET'
        let betaCells = document.querySelectorAll('.addBeta');
        console.log(betaCells)
    }
}

```

```

        let lastBeta = betaCells[betaCells.length - 1];

        if (lastBeta.cells[1].firstChild == null) {
            writeDegr(degr, min, sec, lastBeta.cells[1]);

document.getElementById('lastBetaCorner').removeAttribute("disabled")
;
        } else {
            let afterBetaLines = document.querySelectorAll('.afterBeta');
            let lastAfterBeta = afterBetaLines[afterBetaLines.length - 1];

            lastAfterBeta.insertAdjacentHTML('afterend', '<tr
class="addBeta betaLine"><td></td><td></td><td class="fBeta"></td><td
class="correctedCorner"></td><td colspan="4"></td><td
class="xClass"></td><td class="yClass"></td></tr>');
            betaCells = document.querySelectorAll('.addBeta');
            lastBeta = betaCells[betaCells.length - 1];
            writeDegr(degr, min, sec, lastBeta.cells[1])
            lastBeta.cells[0].innerHTML = 'T' + afterBetaLines.length;

        }
        afterBeta();
        secondForm.reset();
        calcMeasureCorners();
        calculateAmountOfSides();
    }
};

function afterBeta() {
    let degrees;
    let length = secondForm.length.value;
    let betaCells = document.querySelectorAll('.addBeta');
    let lastBeta = betaCells[betaCells.length - 1];

    lastBeta.insertAdjacentHTML('afterend', '<tr class="afterBeta"><td
colspan="4"></td><td class="alphaCorner"></td><td
class="measureLength"></td><td class="measureDistance"></td><td
class="gammaCorner"></td><td colspan="2"></td></tr>');
    let measureLengths = document.querySelectorAll('.measureLength');
    let lastmeasureLength = measureLengths[measureLengths.length - 1];
    let measureDistance =
document.querySelectorAll('.measureDistance');

```

```

    let lastMeasureDistance = measureDistance[measureDistance.length -
1];
    let gammaCorners = document.querySelectorAll('.gammaCorner');
    let lastGammaCorner = gammaCorners[gammaCorners.length - 1];

    if (document.getElementById("measureDistance").selected) {

        lastMeasureDistance.innerHTML = Number(length).toFixed(2);

        let gammaDegr = secondForm.gammaCornerDegr.value;
        let gammaMin = secondForm.gammaCornerMin.value;
        let gammaSec = secondForm.gammaCornerSec.value;

        writeDegr(gammaDegr, gammaMin, gammaSec, lastGammaCorner);

        if (gammaDegr.length > 0 && gammaMin.length == 0 &&
gammaSec.length == 0) {
            degrees = gammaDegr;
        } else if (gammaDegr.length > 0 && gammaMin.length > 0 &&
gammaSec.length == 0) {
            degrees = Number(gammaDegr) + (gammaMin / 60);
        } else if (gammaDegr.length > 0 && gammaMin.length > 0 &&
gammaSec.length > 0) {
            degrees = Number(gammaDegr) + ((Number(gammaMin) + gammaSec /
60) / 60)
        }

        degrees = (degrees == undefined) ? 0 : degrees;

        radians = (degrees * Math.PI) / 180;
        length = Math.cos(radians) * length;
        lastmeasureLength.innerHTML = degrees != 0 ? length.toFixed(2) :
length;

    } else {
        lastmeasureLength.innerHTML = Number(length).toFixed(2);
    }
}

function prevSiblingFunc(elem, selector) {

    var sibling = elem.previousElementSibling;
    if (!selector) return sibling;

```

```

while (sibling) {
    if (sibling.matches(selector)) return sibling;
    sibling = sibling.previousElementSibling;
}
};

function nextSiblingFunc(elem, selector) {

    var sibling = elem.nextElementSibling;
    if (!selector) return sibling;

    while (sibling) {
        if (sibling.matches(selector)) return sibling;
        sibling = sibling.nextElementSibling;
    }
};

minFields.forEach(element => {
    element.addEventListener('keyup', function () {
        if (element.value > 0) {
            if (element.id == 'gammaCornerMin') {
                prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
89);
                prevSiblingFunc(element, '.cornerDegr').setAttribute('min', -
89);
            } else {
                prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
359);
            }
        } else {
            if (element.id == 'gammaCornerMin') {
                prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
90);
                prevSiblingFunc(element, '.cornerDegr').setAttribute('min', -
90);
            } else {
                prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
360);
            }
        }

        if (prevSiblingFunc(element, '.cornerDegr').value == '') {
            prevSiblingFunc(element, '.cornerDegr').value = 0

```

```

    }
  })

  element.addEventListener('change', function () {
    if (element.value.length == '' && nextSiblingFunc(element,
'.cornerSec.field').value >= 0) {
      element.value = 0;
    }
  })
})

secFields.forEach(element => {
  element.addEventListener('keyup', function () {
    if (element.value > 0) {

      if (element.id == 'gammaCornerSec') {
        prevSiblingFunc(element,
'.cornerMin.field').setAttribute('max', 59);
        prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
89);
        prevSiblingFunc(element, '.cornerDegr').setAttribute('min', -
89);
      } else {
        prevSiblingFunc(element,
'.cornerMin.field').setAttribute('max', 59);
        prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
359);
      }
    } else {
      if (element.id == 'gammaCornerSec') {
        prevSiblingFunc(element,
'.cornerMin.field').setAttribute('max', 60);
        if (prevSiblingFunc(element, '.cornerMin.field').value <= 0)
{
          prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
90);
          prevSiblingFunc(element, '.cornerDegr').setAttribute('min',
-90);
        }
      } else {
        prevSiblingFunc(element,
'.cornerMin.field').setAttribute('max', 60);

```

```

        if (prevSiblingFunc(element, '.cornerMin.field').value <= 0)
    {
        prevSiblingFunc(element, '.cornerDegr').setAttribute('max',
360);
    }
}
}
if (prevSiblingFunc(element, '.cornerMin.field').value == '') {
    prevSiblingFunc(element, '.cornerMin.field').value = 0
}
if (prevSiblingFunc(element, '.cornerDegr').value == '') {
    prevSiblingFunc(element, '.cornerDegr').value = 0
}
})
})

```

```

function calcMeasureCorners() {
    let sum = 0,
        sec;
    let betas = document.querySelectorAll('.addBeta');
    let lastBeta =
document.getElementById('addLastBeta').cells[1].textContent;

    betas.forEach(elem => {
        let corner = transDegr(elem.cells[1].textContent);
        sec = fromSecMinDegToSec(corner.deg, corner.min, corner.sec);
        sum += sec;
    })

    if (lastBeta) {
        let res = transDegr(lastBeta);
        sec = fromSecMinDegToSec(res.deg, res.min, res.sec);
        sum += sec;
    }
    let resCorn = fromSecToSecMinDeg(sum);
    writeDegr(resCorn.deg, resCorn.min, resCorn.sec,
document.getElementById('calcCornSum'));
}

```

```

function transDegr(string) {
    let deg, min, sec;
    if (string.includes('°')) {

```

```

        string = string.split('°');
        deg = Number(string.shift());
    } else {
        deg = 0;
    }
    if (string.toString().includes('\')) {
        string = string.toString().split("'");
        min = Number(string.shift());
    } else {
        min = 0;
    }
    if (string.toString().includes('"')) {
        string = string.toString().split('"');
        sec = Number(string.shift());
    } else {
        sec = 0;
    }
    return {
        deg,
        min,
        sec
    };
}

function calcTeorCorners() {
    let sumTeorIn, sumTeorOut, sum;
    console.log(document.querySelectorAll('.betaLine'));
    let countBetas = document.querySelectorAll('.betaLine').length;
    let resCorn = document.getElementById('calcCornSum');
    let teorCornSum = document.getElementById('teorCornSum');
    let res = transDegr(resCorn.textContent);
    res = fromSecMinDegToSec(res.deg, res.min, res.sec);
    let fBeta;
    let traverseType = document.getElementById('traverseType').value;
    let cornersDirection = document.getElementById('direction').value;
    if (traverseType == '1') {
        sumTeorIn = 648000 * (countBetas - 2);
        sumTeorOut = 648000 * (countBetas + 2);

        if ((sumTeorIn - 3600) < res && res < (sumTeorIn + 3600)) {
            sum = fromSecToSecMinDeg(sumTeorIn);
        } else if ((sumTeorOut - 3600) < res && res < (sumTeorOut +
3600)) {

```

```

        sum = fromSecToSecMinDeg(sumTeorOut);
    } else {
        alert('Невірно введені кути, розрахунок неможливий');
    }

} else if (traverseType == '2') {
    let alphaCorn = transDegr(alpha.textContent);
    alphaCorn = fromSecMinDegToSec(alphaCorn.deg, alphaCorn.min,
alphaCorn.sec);
    let lastCorn = transDegr(lastAlpha.textContent);
    lastCorn = fromSecMinDegToSec(lastCorn.deg, lastCorn.min,
lastCorn.sec);
    if (cornersDirection == '1') {
        sum = fromSecToSecMinDeg((alphaCorn - lastCorn) + (648000 *
countBetas));
    } else if (cornersDirection == '2') {
        sum = fromSecToSecMinDeg((lastCorn - alphaCorn) + (648000 *
countBetas));
    }
}
writeDegr(sum.deg, sum.min, sum.sec, teorCornSum);

sum = fromSecMinDegToSec(sum.deg, sum.min, sum.sec);

fBeta = (res - sum);
let sign = fBeta.toString().substr(0, 1);

if (sign == '-') {
    fBeta = fromSecToSecMinDeg(Number(fBeta.toString().slice(1)));
} else {
    sign = '+';
    fBeta = fromSecToSecMinDeg(fBeta);
}
writeDegr(fBeta.deg, fBeta.min, fBeta.sec, fBetaSum);
fBetaSum.innerHTML = sign + fBetaSum.textContent;

if (fBetaValid(fBeta, countBetas)) {
    shareFBeta(fBeta, countBetas, sign);
}
//shareFBeta(fBeta, countBetas, sign);
}

```

```

function fBetaValid(fBeta, countBetas) {
    fBeta = fromSecMinDegToSec(fBeta.deg, fBeta.min, fBeta.sec);
    let classValue = document.getElementById('classAccuracy').value;
    let controlBeta = document.getElementById('controlBeta');
    let control, res;
    if (classValue == '1') {
        control = (1 * Math.sqrt(countBetas)) * 60;
    } else if (classValue == '2') {
        control = 10 * Math.sqrt(countBetas);
    } else if (classValue == '3') {
        control = 20 * Math.sqrt(countBetas);
    } else {
        control = 5 * Math.sqrt(countBetas);
    }
    let con = fromSecToSecMinDeg(control);
    writeDegr(con.deg, con.min, con.sec, controlBeta)
    if (fBeta > control) {
        alert("Кутова нев'язка більша за допустиму, розрахунок зупинено");
        res = false;
    } else {
        res = true;
    }
    return res;
}

```

```

function shareFBeta(fBeta, countBetas, sign) {
    let distances = document.querySelectorAll('.measureLength');
    let distList = [];
    distances.forEach(elem => {
        distList.push(elem.textContent);
    })
    let minDistance = Math.min(...distList);
    let maxDistance = Math.max(...distList);
    distances.forEach(elem => {
        if (elem.textContent == minDistance) {
            elem.classList.add('minLength');
        }
        if (elem.textContent == maxDistance) {
            elem.classList.add('maxLength');
        }
    })
    fBeta = fromSecMinDegToSec(fBeta.deg, fBeta.min, fBeta.sec);
}

```

```

let d = fBeta / countBetas;
let fBetas = document.querySelectorAll('.fBeta');
if ((d - Math.trunc(d)) == 0) {
  fBetas.forEach(element => {
    element.innerHTML = d + ''
  })
} else {
  fBetas.forEach(element => {
    element.innerHTML = Math.trunc(d) + ''
  })
  let elements = document.querySelectorAll('.minLength');
  elements.forEach(elem => {
    let value =
elem.parentElement.nextElementSibling.children[2].textContent;
    let res = Number(value.split('').shift()) + ((fBeta -
(Math.trunc(d) * countBetas)) / elements.length);
    elem.parentElement.nextElementSibling.children[2].innerHTML =
res + '';
  })
}
fBetas.forEach(elem => {
  if (sign == '-') {
    elem.innerHTML = '+' + elem.textContent;
  } else if (sign == '+') {
    elem.innerHTML = '-' + elem.textContent;
  }
})
correctedCorners();
}

function correctedCorners() {
  let lines = document.querySelectorAll('.betaLine');
  lines.forEach(elem => {
    let corn = transDegr(elem.children[1].textContent);
    let sec = fromSecMinDegToSec(corn.deg, corn.min, corn.sec);
    let res = (sec +
Number(transDegr(elem.children[2].textContent).sec));
    let sign = res.toString().substr(0, 1);
    if (sign == '-') {
      res = fromSecToSecMinDeg(Number(res.toString().slice(1)));
    } else {
      sign = '';
      res = fromSecToSecMinDeg(res);
    }
  })
}

```

```

    }
    writeDegr(res.deg, res.min, res.sec, elem.children[3]);
    elem.children[3].innerHTML = sign + elem.children[3].textContent;
  })
  calcAlphaCorners();
}

function fromSecToSecMinDeg(res) {
  let sec = Math.round((res / 60 - Math.trunc(res / 60)) * 60);
  let min = Math.round((Math.trunc(res / 60) / 60 -
Math.trunc(Math.trunc(res / 60) / 60)) * 60);
  let deg = Math.trunc(Math.trunc(res / 60) / 60);
  return {
    deg,
    min,
    sec
  };
}

function fromSecMinDegToSec(deg, min, sec) {
  if (deg.toString().substr(0, 1) == '-') {
    deg = Number(deg.toString().slice(1));
    sec = -((Number(deg) * 60 + Number(min)) * 60 + Number(sec));
  } else {
    sec = ((Number(deg) * 60 + Number(min)) * 60 + Number(sec));
  }
  return sec;
}

function calcAlphaCorners() {
  let res;
  let corners = document.querySelectorAll('.alphaCorner');
  let cornersDirection = document.getElementById('direction').value;
  let alpha = document.getElementById('lastAlpha');

  corners.forEach(elem => {
    let elemIndex =
Array.prototype.slice.call(corners).indexOf(elem);
    let betaCorner =
transDegr(elem.parentElement.nextElementSibling.children[3].textConte
nt);

    elem = transDegr(elem.textContent);

```

```

    elem = fromSecMinDegToSec(elem.deg, elem.min, elem.sec);

    beta = fromSecMinDegToSec(betaCorner.deg, betaCorner.min,
betaCorner.sec);

    if (cornersDirection == '1') {
        res = elem - beta + (648000);
    } else if (cornersDirection == '2') {
        res = elem + beta - (648000);
    }

    if (res > 1296000) {
        res = res - 1296000;
    } else if (res < 0) {
        res = res + 1296000;
    }

    if (corners[elemIndex + 1]) {
        res = fromSecToSecMinDeg(res);
        writeDegr(res.deg, res.min, res.sec, corners[elemIndex + 1]);
    } else {
        alpha = transDegr(alpha.textContent);
        if (res == fromSecMinDegToSec(alpha.deg, alpha.min, alpha.sec))
{
            console.log(true)
        } else {
            console.log(false);
        }
    }
})
calculatePerimetr();
}

function calculatePerimetr() {
    let lengths = document.querySelectorAll('.measureLength');
    let sum = 0;
    lengths.forEach(elem => {
        sum += Number(elem.textContent);
    })
    document.getElementById('perimeter').innerHTML = sum.toFixed(2);
    countDeltX();
}

```

```

function countDeltX() {
  let lines = document.querySelectorAll('.afterBeta');
  let sumX = 0;
  let sumY = 0;
  lines.forEach(elem => {
    let corn = transDegr(elem.children[1].textContent);
    let degrees = Number(corn.deg) + ((Number(corn.min) +
Number(corn.sec) / 60) / 60);
    let radians = (degrees * Math.PI) / 180;
    let length = Number(elem.children[2].textContent)
    deltx = Number((Math.cos(radians) * length).toFixed(2));
    deltY = Number((Math.sin(radians) * length).toFixed(2));
    sumX += deltx;
    sumY += deltY;
    elem.children[5].innerHTML = deltx;
    elem.children[6].innerHTML = deltY;
  })
  let teorX =
Number(document.getElementById('penultPointXCoordinate').textContent)
-
Number(document.getElementById('secondPointXCoordinate').textContent)
;
  let teorY =
Number(document.getElementById('penultPointYCoordinate').textContent)
-
Number(document.getElementById('secondPointYCoordinate').textContent)
;

  document.getElementById('sumX').innerHTML = sumX.toFixed(2);
  document.getElementById('sumY').innerHTML = sumY.toFixed(2);
  document.getElementById('teorSumX').innerHTML = teorX.toFixed(2);
  document.getElementById('teorSumY').innerHTML = teorY.toFixed(2);
  let xRes = Number((sumX - teorX).toFixed(2));
  let yRes = Number((sumY - teorY).toFixed(2));
  document.getElementById('sumMinTeorX').innerHTML = xRes;
  document.getElementById('sumMinTeorY').innerHTML = yRes;

  document.getElementById('fAbs').innerHTML = fAbs().toFixed(2);

  let per = Number(document.getElementById('perimeter').textContent);
  lines.forEach(elem => {

```

```

        elem.children[7].innerHTML = -(((xRes / per) *
Number(elem.children[2].textContent)).toFixed(2));
        elem.children[8].innerHTML = -(((yRes / per) *
Number(elem.children[2].textContent)).toFixed(2));
    })

    let xPractSum = 0;
    let yPractSum = 0;
    let fXs = document.querySelectorAll('.fX');
    let fYs = document.querySelectorAll('.fY');
    fXs.forEach(elem => {
        xPractSum += Number(elem.textContent);
    })
    fYs.forEach(elem => {
        yPractSum += Number(elem.textContent);
    })

    let maxLengths = document.querySelector('.maxLength');
    let minLengths = document.querySelector('.minLength');
    let valueX = maxLengths.parentElement.children[7];
    let valueY = maxLengths.parentElement.children[8];
    let minX =
Number(minLengths.parentElement.children[7].textContent);
    let minY =
Number(minLengths.parentElement.children[8].textContent);

    if ((xRes + Number(xPractSum.toFixed(2))) > 0) {
        if (Math.abs(minX) > 0) {
            minLengths.parentElement.children[7].innerHTML = (-0.01 +
minX).toFixed(2);
        } else {
            maxLengths.parentElement.children[7].innerHTML = (-0.01 +
Number(valueX.textContent)).toFixed(2);
        }
    }
    if ((xRes + Number(xPractSum.toFixed(2))) < 0) {
        if (Math.abs(minX) > 0) {
            minLengths.parentElement.children[7].innerHTML = (+0.01 +
minX).toFixed(2);
        } else {
            maxLengths.parentElement.children[7].innerHTML = (+0.01 +
Number(valueX.textContent)).toFixed(2);
        }
    }

```

```

    }
    if ((yRes + Number(yPractSum.toFixed(2))) > 0) {
        if (Math.abs(minY) > 0) {
            minLengths.parentElement.children[8].innerHTML = (-0.01 +
minY).toFixed(2);
        } else {
            maxLengths.parentElement.children[8].innerHTML = (-0.01 +
Number(valueY.textContent)).toFixed(2);
        }
    }
    if ((yRes + Number(yPractSum.toFixed(2))) < 0) {
        if (Math.abs(minY) > 0) {
            minLengths.parentElement.children[8].innerHTML = (+0.01 +
minY).toFixed(2);
        } else {
            maxLengths.parentElement.children[8].innerHTML = (+0.01 +
Number(valueY.textContent)).toFixed(2);
        }
    }
}

lines.forEach(elem => {
    elem.children[9].innerHTML =
(Number(elem.children[5].textContent) +
Number(elem.children[7].textContent)).toFixed(2);
    elem.children[10].innerHTML =
(Number(elem.children[6].textContent) +
Number(elem.children[8].textContent)).toFixed(2);
})

let xPoints = document.querySelectorAll('.xClass');
let yPoints = document.querySelectorAll('.yClass');
calcPoints(xPoints, 9);
calcPoints(yPoints, 10);
fRel();
}

function fAbs() {
    let x = Number(document.getElementById('sumMinTeorX').textContent);
    let y = Number(document.getElementById('sumMinTeorY').textContent);
    let res = Math.sqrt(Math.pow(x, 2) + Math.pow(y, 2));
    return res;
}

```

```

function fRel() {
    let p = Number(document.getElementById('perimeter').textContent);
    let f = fAbs();
    let res = p / f;
    document.getElementById('fRel').innerHTML = '1' + ':' +
Math.round(res);

    let classValue = document.getElementById('classAccuracy').value;
    if (classValue == '2') {
        if ((1 / res) > (1 / 10000)) {
            alert('Відносна помилка ходу перевищує допустиме значення,
перевірте правильність введених даниих та повторіть розрахунок');
        }
    } else if (classValue == '3') {
        if ((1 / res) > (1 / 5000)) {
            alert('Відносна помилка ходу перевищує допустиме значення,
перевірте правильність введених даниих та повторіть розрахунок');
        }
    } else if (classValue == '4') {
        if ((1 / res) > (1 / 25000)) {
            alert('Відносна помилка ходу перевищує допустиме значення,
перевірте правильність введених даниих та повторіть розрахунок');
        }
    } else if (classValue == '1') {
        if ((1 / res) > (1 / 2000)) {
            alert('Відносна помилка ходу перевищує допустиме значення,
перевірте правильність введених даниих та повторіть розрахунок');
        }
    }
}

function calcPoints(el, index) {
    let res;
    el.forEach(elem => {
        let elemIndex = Array.prototype.slice.call(el).indexOf(elem);
        let deltpoint =
Number(elem.parentElement.nextElementSibling.children[index].textCont
ent);
        res = (Number(elem.textContent) + deltpoint).toFixed(2)
        if (el[elemIndex + 1]) {
            el[elemIndex + 1].innerHTML = res;
        } else {
            console.log(res);
        }
    }
}

```

```

    })
}

document.getElementById('length').addEventListener('keyup',
lengthLimitation)
function lengthLimitation() {
    let x, y, l;
    let builtUp = document.getElementById('builtUp').value;
    let classValue = document.getElementById('classAccuracy').value;
    if (classValue == '1') {
        if (builtUp == '1') {
            x = 1000;
            y = 200;
            l = 'Довжина сторони ходу має бути не менше 200 та не більше
1000 метрів';
        } else if (builtUp == '2') {
            x = 1500;
            y = 400;
            l = 'Довжина сторони ходу має бути не менше 400 та не більше
1500 метрів';
        }
    } else if (classValue == '2') {
        x = 800;
        y = 120;
        l = 'Довжина сторони ходу має бути не менше 120 та не більше 800
метрів';
    } else if (classValue == '3') {
        x = 500;
        y = 80;
        l = 'Довжина сторони ходу має бути не менше 80 та не більше 500
метрів';
    } else if (classValue == '4') {
        x = 3000;
        y = 250;
        l = 'Довжина сторони ходу має бути не менше 250 та не більше 3000
метрів';
    }
    limitLength(x, y, l);
}

function limitLength(x, y, l) {
    let length = document.getElementById('length');
    let err = document.getElementById('errorSpan');

```

```

    if (x >= length.value && length.value >= y) {
        length.classList.remove('invalid');
        err.style.display = 'none';
    } else {
        length.classList.add('invalid');
        err.style.display = 'block';
        err.innerHTML = 1;
    }
}

function calculateAmountOfSides() {
    let builtUp = document.getElementById('builtUp').value;
    let classValue = document.getElementById('classAccuracy').value;
    let scale = document.getElementById('scale').value;
    let sides = document.querySelectorAll('.measureLength').length;
    console.log(sides);
    if (classValue !== '1' && sides == 15) {
        res();
    } else {
        if (scale == '500') {
            if (builtUp == '1' && sides == 20) {
                res();
            }
        } else if (scale == '1000' && sides == 20) {
            res();
        } else if (scale == '2000'){
            if((builtUp == '1' && sides == 20) || (builtUp == '2' && sides
== 30)){
                res();
            }
        } else if (scale == '5000'){
            if((builtUp == '1' && sides == 30) || (builtUp == '2' && sides
== 40)){
                res();
            }
        }
    }
}

function res() {
    document.getElementById('sideOfTraverseButton').disabled = true;
    alert("Ви ввели максимально допустиму кількість сторін у ході, для
завершення розрахунку додайте значення останнього кута");
}

```