**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ**

**КУРСОВА РОБОТА**

з навчальної дисципліни

**Об’єктно-орієнтоване програмування для економічних систем**

на тему: «*Реалізація об'єктно-орієнтованого підходу при розробці програми, що моделює систему отримання права власності на земельну ділянку в державі Україна»*

Виконала

Студентка групи УК-02

Музалевська Альона Дмитрівна

Керівник курсової роботи

доцент, канд. ф.-м. н. Стець О. В.

**Київ – 2022**

**Зміст**

[**Вступ** 3](#_Toc106568933)

[**1.** **Огляд предметної області моделювання системи отримання права власності на земельну ділянку в державі Україна** 4](#_Toc106568934)

[**2.** **Постановка задачі розробки моделі системи отримання права власності на земельну ділянку в державі Україна** 6](#_Toc106568935)

[**3.** **Основні теоретичні положення ООП** 7](#_Toc106568936)

[**4.** **Опис структури класів та їх взаємодія** 12](#_Toc106568937)

[**5.** **Реалізація класів в середовищі C++** 15](#_Toc106568938)

[**6.** **Інтерфейс користувача системою отримання права власності на земельну ділянку** 23](#_Toc106568939)

[**7.** **Висновки** 27](#_Toc106568940)

[**8.** **Список використаних джерел** 28](#_Toc106568941)

[**9.** **Лістинг програми** 29](#_Toc106568942)

# **Вступ**

*Об'єктно-орієнтоване програмування або ООП* (object - oriented programming) – методологія програмування, заснована на представленні програми у вигляді сукупності об'єктів кожен з яких являється реалізацією визначеного типу, що використовує механізм пересилки повідомлень і класи організовані в ієрархію наслідування. Його використання дає змогу використовувати один і той же програмний код з різними даними. Класи стаюсть основою для створення великої кількості об'єктів, у кожного з яких можуть бути власні значення полів. Спадкування дозволяє не писати новий код, а використовувати і налаштовувати вже існуючий за рахунок додавання і перевизначення атрибутів. Основу ООП складають чотири основні концепції: інкапсуляція, успадкування, поліморфізм та абстракція

В даній курсовій роботі був створений програмний продукт в середовищі об'єктно-орієнтовного програмування за допомогою мови C++, який описує та удосконалює існуючу систему оформлення права власності на земельну ділянку в Україні, яку людина може подати дистанційно.

***Метою курсової роботи*** є удосконалення своїх знань, які засвоїли під час вивчення дисципліни, закріплення навичок правильно використовувати основні інструменти об'єктно-орієнтованого програмування використовуючи мову С++.

***Об'єктом дослідження*** курсової роботи є система отримання права власності на земельну ділянку, а саме її складові елементи.

## Огляд предметної області моделювання системи отримання права власності на земельну ділянку в державі Україна

*Право власності на землю* - це право володіти, користуватися і розпоряджатися земельними ділянками. *Власність на землю*— належність фiзичній особi, юридичній особі, територіальній громаді чи державі земельної ділянки на праві [володіння](https://uk.wikipedia.org/wiki/%D0%92%D0%BE%D0%BB%D0%BE%D0%B4%D1%96%D0%BD%D0%BD%D1%8F), [користування](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%B8%D1%81%D1%82%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F) та [розпорядження](https://uk.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B7%D0%BF%D0%BE%D1%80%D1%8F%D0%B4%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%B0%D0%B2%D0%BE%D0%BC%D0%BE%D1%87%D0%BD%D1%96%D1%81%D1%82%D1%8C)) нею. При реєстрації права власності на землю, громадянин отримує [витяг про земельну ділянку](https://www.kmu.gov.ua/ua/service/vityag-pro-zemelnu-dilyanku).

Особа, при бажання оформлення права власності, має спершу перевірити наявність в нього кадастрового номеру (унікальна послідовність цифр, яка надається земельній ділянці). Якщо ви бажаєте присвоїти кадастровий номер своїй земельній ділянці, потрібно звернутися до найближчого центру Державного земельного кадастру для оформлення технічної документації.

Після того як кадастровий номер ділянці присвоєне потрібно із зібраними наступними документами звернутися у ЦНАП:

* Витяг з Державного земельного кадастру (ДЗК);
* Оригiнал паспорту заявника та його копія;
* Оригінал ідентифікаційного номеру та його копія;
* Підтвердження оплати адміністративного збору;
* Документ , який має в собі підставу для виникнення права на земельну ділянку.

Якщо документи подаються представником то:

* + Довіреність на представника;
* Оригінал, копія паспорту представника;
* Ідентифікаційний номер представника[5].

Заява про реєстрацію права друкується держреєстратором або нотаріусом та підписується заявником на місці.

Останнім пунктом є отримання витягу (у паперовій або електронній формі) із Державного реєстра прав про державну реєстрацію права власності на земельну ділянку. Загальним строком розгляду поданих для державної реєстрації документів є п’ять робочих днів. Однак є можливiсть прискорити розгляд до двох, одного робочого дня, 2-х годин, при цьому розмір адміністративного збору змінюється[4].

## Постановка задачі розробки моделі системи отримання права власності на земельну ділянку в державі Україна

Розглядаючи основне завдання курсової роботи, тобто моделювання системи отримання права власності на земельну ділянку, сформулюємо основні моменти моделі:

* Ідентифікація особи через введення електронної пошти та пароля для осіб, які вже зареєстровані, чи шляхом введення своїх персональних даних;
* Перевірка наявності кадастрового номера
* Перевірка наявності документів, які дають право на отримання права власності на земельну ділянку;
* Вибір терміну та розміру адміністративного збору;
* Вибір способу оплати адміністративного збору;
* Отримання заяви, з необхідними даними для створення витягу із Державного реєстру прав;

## Основні теоретичні положення ООП

*Об’єктно-орієнтоване програмування (ООП)* - одна з парадигм програмування, яка розглядає програму як множину об’єктів, що взаємодіють мiж собою. В нiй використано декілька технологій від попередніх парадигм, зокрема успадкування, полморфiім та інкапсуляцію. [1]

*Парадигма* (дав.-гр. παράδειγμα, paradeigma - приклад, зразок) - у сучасній філософії науки - система теоретичних, методологічних і аксіологічних установок, які взяті за зразок.

*Об'єкт* — базовий елементами побудови програми, втілення абстрактної моделі окремої сутності, що має спеціально визначене призначення та дані. Властивості об'єкта визначаються його атрибутами, тобто полями даних. Поточне значення атрибутів визначає поточний стан об'єкта у множині можливих станів. Поведінка об'єкта — це методи об'єкта. Саме властивоті й поведінка об'єкта визначають його [ідентичність](https://uk.wikipedia.org/w/index.php?title=%D0%86%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%87%D0%BD%D1%96%D1%81%D1%82%D1%8C_(%D0%9E%D0%9E%D0%9F)&action=edit&redlink=1), а це, в свою чергу, вирізняє об'єкт з-поміж інших об'єктiв.

*Клас* — це, визначена розробником, конструкція, яка використовує певні об’єктів, атрибутів і методів. Створений та ініцiалізований екземпляр класу називають об'єктом класу. На основi одного класу можна створити безліч об'єктів, що відрізнятимуться один від одного своїм станом (значеннями полів). На основі класів можна створювати підкласи, якi [успадковують](https://uk.wikipedia.org/wiki/%D0%A3%D1%81%D0%BF%D0%B0%D0%B4%D0%BA%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) властивості та поведінку батьківських класів, будувати ієрархію.

[Методи](https://uk.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) — функції, які допомагають описати поведінку об'єктів. Завдяки ним об'єкт може змінювати свій стан. Також методи в змозі надавати доступ до даних, які були інкапсульовані в об'єкті. Саме це – право доступу та вид методів - задаються модифікаторами при описі методів. [Конструктором](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) класу є метод, який проводить створення та початкову ініціалізацію екземпляра класу. І навпаки метод, який проводить знищення об'єкта, називають [деструктором](https://uk.wikipedia.org/wiki/%D0%94%D0%B5%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)) класу.

Об’єктно-орієнтованих мови програмування , які можна поділити на два види:

* класичні: Simula 67, Smalltalk, AСtor, Eiffel, Objective С;
* сучасні: С++, Object Pasсal, Java, С#.

   Об’єктно-орієнтоване програмування має коріння ще в1960-тих роках, коли тільки започатковувалась маво програмування. Воно трохи вирішує проблему зберігання якості програм при збільшені складності апаратного та програмного забезпечення шляхом наголошення на модульності програми.

Програми, створені завдяки об’єктно-орієнтованому програмуванню, навідміну від програм іншого типу, які складаються з підпрограм або як перелік інструкцій комп’ютеру, можна вважати сукупністю об’єктів. Відповідно до парадигми об’єктно-орієнтованого програмування, кожен об’єкт має здатність отримувати повідомлення, обробляти та надсилати далі повідомлення iншим об’єктам, він унікальний незалежний з відповідним призначенням та відповідальністю.

Чотири принципи об'єктно-орієнтованого програмування - це інкапсуляція, абстракція, спадкування і поліморфізм. Розкритий зміст кожного:

- *Інкапсуляція* (приховування даних). Вся важлива інформація в об'єкті знаходиться всередині, а доступна тільки вибрана інформація. В класі зберігається уся інформація про стан кожного об’єкта, при тому що інші об’єкти до цього класу не мають доступу. Ця особливість дає змогу зменшити можливість пошкодження даних

- *Абстракція*. ЇЇ можна розглядати як природне продовження інкапсуляції. При об'єктно-орієнтованому проектуванні програми часто бувають дуже великими. Підтримувати таку велику кодову базу дуже складно. Об’єкти розкривають лише внутрішні механізми, які є важливими для використання інших об’єктів, приховуючи будь-який непотрібний код реалізації.

- *Спадкування*. Класи мають право декілька разів використовувати коди інших класів. Якщо розробник бажає повторно використати загальну логіку при будівництві ієрархії, то між об’єктами він може призначити зв’язки та підкласи. Такий принцип допомагає витрачати менше часу на розробку та збільшує рівень точності аналізу даних.

- *Поліморфізм.* Це означає залежність поведінки від класу, в якому ця поведінка викликається, тобто, два або більше класiв можуть реагувати по різному на однакові повідомлення. Програма сама визначатиме необхідність якого значення потрібно для виконання об’єкта з батьківського класу, при цьому зменшуючи необхідність дублювання коду.

Все, що входить в клас, пов'язане з модифікатором доступу, який визначає сферу застосування або видимість цього об'єкту. Основними вважаються модифікатори public, protected і private:

1. *Public* (відкритий член класу) — звернення до члена допускаються з будь-якого коду. Він має спадкування, яке зовсім не змінює модифікаторів батьківського класу (С++);
2. *Protected* (член класу, знаходиться всередині, захищений)— звернення до члена мають змогу лише з методiв того класу, у якому цей член визначений, але й його класів-спадкоємцiв з будь-яких методiв. Цей тип має спадкування, яке робить всі public-члени батьківського класу protected-членами класу-спадкоємця (С++);
3. *Private* (член класу, який знаходиться всередині, закритий) — звернення до члену мають змогу лише з методiв того класу, у якому цей член визначений. Ніякі спадкоємці класу не мають доступу до цього члену. Цей тип має спадкування, яке забороняє доступ з дочірнього класу до всiх членiв батькiвського класу, включаючи навіть public-члени(С++)[3];

Визначаючи клас, ми не можемо ініціювати його поля (члени) в самому визначенні. Можна присвоїти їм значення, написавши відповідний метод класу і викликавши його, після створення об'єкта поза класом. Такий спосіб не є зовсім зручним, так як оголошуючи, припустимо, 10 об'єктiв класу нам доведеться 10 разiв викликати метод, який присвоює значення полів класу. Тому, як правило, для iніціалізацiї полів класу, а так само для виділення динамічної пам'яті, використовується конструктор.

*Конструктор* - це метод класу, який виконується автоматично в момент створення об'єкту класу і виконує його функціонал. *Ще* існує *деструктор*. Це теж метод класу, але який працює при знищенні об'єктів класу. Найчастіше його роль полягає в тому, щоб звільнити динамічну пам'ять, яку виділяв конструктор для об'єкта.

Важливi аспекти застосування конструкторів i деструкторiв в програмі:

* Конструктор i деструктор мають бути public;
* Конструктор i деструктор не мають типу значення, що повертається;
* Конструктор може приймати параметри, навдміну від деконструктора;
* Iм'я класу, конструктора і деструктора мають бути однаковими;
* Якщо визначаєте деструкцію, то перед ім'ям треба додатково написати символ ~;
* Ви можете створювати кілька конструкторів, але їх сигнатура повинна відрізнятися;
* В класі деструктор має бути визначеним тільки один[2].

## Опис структури класів та їх взаємодія

Програма була створена за допомогою середовища програмування CodeBlocks на мові С++. При розробці моделі програми були реалізовані такі класи:

1. class Person;
2. class Authorisation;
3. class Cadastral\_number;
4. class Permission;
5. class Payment;
6. class Statement;

Зв’язок між класами зображено на Рис.1 у вигляді діаграми Буча.

З діаграми видно, що ми маємо один абстрактний клас Person, який в свою чергу містить protected змінну findcad та віртуальний метод Choice(). Абстрактний клас має два наслідники, які успадковують його поле та метод (класи Authorisation та Cadastral\_number). Вони унаслідують і protected змінну, і public virtual void.

Клас Permission асоційований з класом Cadastral\_number у вигляді композиції. Клас Permission ніяк не впливає на інші класи.

Клас Payment та клас Statement наслідники класу Authorisation, а клас Statement є наслідником класу Payment. Тобто клас Statement має множинне наслідування водночас трьох класів: Authorisation, Cadаstral\_number і Payment. При чому успадковав певні методи та поля цих трьох класів.

class Person

int findcad;

virtual void Choice();

class Cadastral\_number

int amount, fee, cad\_number;

void Choice();

void Response();

void Admin\_Fee();

void Response\_cad();

class Authorisation

string email, password, name, surname, middle;

int option, ident, passport;

void For\_old\_ac();

void New\_ac();

void Account();

void Old\_ac()

class Permission

int Doc\_property\_rights;

void Rights();

class Statement

void Statement\_method(int option, string surname, string name, string middle, int ident, int passport, int cad\_number, int amount, int p\_method, int bank\_card);

class Payment

int p\_method, bank\_card,

void Pay\_method();

**А**

Рис. 1 Діаграма класів

Розглядаємо класи детальніше. Абстрактному клас Person має одну protected змінну findcad, яка використовується public virtual void Choice(). Віртуальний метод потрібен для здійснення опитування користувача щодо наявності у нього потрібних документів для реалізації послуги.

Клас Authorisation працює на авторизації чи ідентифікації особи в системі в залежності від обраного варіанту особою:

1. Через авторизацію: користувач вводить свій email та password для входу в систему. При успішній перевірці даних видається всі інформація про користувача, інакше система просить ввести email/password ще раз.
2. Ідентифікація особи через те, що користувач сама вводить всі свої особисті дані, які потрібні для оформлення права власності. Користувач має ввести: surname – прізвище, name – ім’я, middle – по-батькові, ident – ідентифікаційний код, passport – номер паспорта. Ці дані система запам’ятовує.

Перевірка відбувається наявності кадастрового номера та документів у класі Cadаstral\_number, що підтверджують право користувача отримати право власності на конкретну земельну ділянку. Якщо б хоча б одне буде відсутнє, то програма виконуватися не буде та попередить користувача о необхідності цих документів. Композиція класу Permission застосована для перевірки документів (doc), наявність кадастрового номера (cad\_number) перевіряється за допомогою віртуального методу Choice(), який був успадкований з абстрактного класу. Ще людина має обирати термін та вартість отримуваної послуги за допомогою методів Admin\_fee() та Response().

Клас Payment, який успадковує поля та методи класу Authorisation, є класом де людина має вибирати спосіб оплати адміністративного збору. Першим варіантом є сплата готівкою і одному з центрів ЦНАП його (користувача) міста. Другий варіант для оплати карткою на офіційному сайті ЦНАП. Якщо обран другий варіант, тоді додатково людина маєте ввести номер банківської картки з якою відбудеться сплата.

У класі Statement ми бачимо наслідування одразу 3 класів. Саме в цьому класі збирається до купи вся зібрана інформація та групується. Першим ми бачимо особисту інформацію про людину-користувача, далі кадастровий номер земельної ділянки, на яку оформлюється право власності, вибраний термін оформлення та сума адміністративного збору, яка залежить від обраного терміну, спосіб оплати (якщо вибрана оплата банківською картою, то ще ‘ номер банківської карти).

## Реалізація класів в середовищі C++

Програмний код , який був реалізований для виконання поставленої задачі курсової роботи, а саме моделювання системи отримання права власності на земельну ділянку в державі Україна.

1. Абстрактний клас Person:

class Person

{

protected:

int findcad;

public:

virtual void Choice()

{

cout << "Виберiть варiант" << endl;

cin >> findcad;

}

};

2. Створюємо клас-наслідник Authorisation:

class Authorisation: public Person

{

public:

int option;

string email;

string password;

string name;

string surname;

string middle;

int passport;

int ident;

void Choice()

{

cout << "1. В мене вже є аккаунт" << endl;

cout << "2. В мене немає аккаунту" << endl;

cin >> option;

}

void For\_old\_ac()

{

cout << "Введiть свою назву своєї електронної пошти: " << endl;

while (cin >> email)

{

if (email != "kur@gmail.com")

{

cout << "Пошта не вiрна, перевiрте правильнiсть" << endl;

}

else

{

break;

}

}

cout << "Введiть пароль: " << endl;

while (cin >> password)

{

if (password != "19112003")

{

cout << "Пароль не вiрний, перевiрте правильнiсть" << endl;

}

else

{

break;

}

}

}

void Old\_ac()

{

cout << "Прiзвище: Muzalevska" << endl;

cout << "Iм'я: Aliona" << endl;

cout << "По-батьковi: Dmitrivna" << endl;

cout << "Номер паспорту: 000437589" << endl;

cout << "Iдентифiкацiйний код: 1234567891" << endl;

}

void New\_ac()

{

cout << "Ваше прiзвище: " << endl;

cin >> surname;

cout << "Ваше iм'я: " << endl;

cin >> name;

cout << "Ваше iм'я по-батьковi: " << endl;

cin >> middle;

cout << "Ваш номер паспорту: " << endl;

cin >> passport;

cout << "Ваш iдентифiкацiйний код: " << endl;

cin >> ident;

}

void Account()

{

if (option == 1)

{

For\_old\_ac();

Old\_ac();

}

else if (option == 2)

{

New\_ac();

}

}

};

3. Клас-наслідник Cadastral\_number та клас Permission, ( він зв’язується завдяки використання композиції):

class Cadastral\_number: public Person

{

private:

class Permission

{

public:

int doc;

void Rights()

{

cout << "Чи ви володiєте документами, якi пiдтверджують ваше право власностi?" << endl;

cout << "1. Так, володiю." << endl;

cout << "2. Нi, не володiю." << endl;

cin >> doc;

if (doc == 1)

{

cout << "Ви маєте сплатити адмiнiстративний збiр." << endl;

}

else

{

cout << "Звернiть увагу, без вiдповiдних документiв неможливо отримати право на власнiсть!" << endl;

abort();

}

}

};

public:

int amount;

int fee;

int cad\_number;

void Admin\_fee()

{

cout << "Виберiть варiант, який найбiльше вас влаштовує (розмiр збору та час, за який державна реєстрацiя права власностi буде зроблена:" << endl;

cout << "1) 230 UAH - у строк, що не перевищує 5 робочих днiв з дня реєстрацiї" << endl;

cout << "2) 2270 UAH - у строк два робочi днi" << endl;

cout << "3) 4540 UAH - у строк один робочий день" << endl;

cout << "4) 11350 UAH - у строк двi години" << endl;

cin >> amount;

}

void Response()

{

if (amount == 1)

{

cout << "Ви вибрали перший варiант, що передбачає сплату адмiнiстративного збору в розмiрi 230 UAH" << endl;

fee = 230;

}

else if (amount == 2)

{

cout << "Ви вибрали другий варiант, що передбачає сплату адмiнiстративного збору в розмiрi 2270 UAH" << endl;

fee = 2270;

}

else if (amount == 3)

{

cout << "Ви вибрали третiй варiант, що передбачає сплату адмiнiстративного збору в розмiрi 4540 UAH" << endl;

fee = 4540;

}

else if (amount == 4)

{

cout << "Ви вибрали четвертий варiант, що передбачає сплату адмiнiстративного збору в розмiрi 11350 UAH" << endl;

fee = 11350;

}

}

void Choice()

{

cout << "Чи ви маєте кадастровий номер земельної дiлянки?" << endl;

cout << "1. Так, маю." << endl;

cout << "2. Нi, не маю." << endl;

cin >> findcad;

}

void Response\_cad()

{

if (findcad == 1)

{

cout << "Вкажiть свiй кадастровий номер: " << endl;

cin >> cad\_number;

}

else

{

cout << "Вам потрiбно отримати кадастровий номер. Бiльше детальної iнформацiю ви зможете знайти на офiцiйному сайтi Державного земельного кадастру." << endl;

abort();

}

}

Permission proper;

void Rights()

{

proper.Rights();

}

};

4. Клас Payment (наслідує клас Authorisation):

class Payment: public Authorisation

{

public:

int pay;

int card;

void Pay\_method()

{

cout << "1. Сплата готiвкою в одному з центрiв ЦНАП вашого мiста" << endl;

cout << "2. Банкiвською картою на офiцiйному сайтi ЦНАП" << endl;

cin >> p\_method;

if (pay == 1)

{

cout << "Звернiться, до одного з центрiв ЦНАП вашого мiста з заявою, яку ви отримаєте в кiнцi" << endl;

}

else if (pay == 2)

{

cout << "Будь ласка, введiть номер картки:" << endl;

cin >> card;

}

}

};

5. Клас Statement (він наслідує одразу 3 класи: Authorisation, Cadastral\_number, Payment):

class Statement: public Authorisation, public Cadastral\_number, public Payment

{

public:

void Statement\_method(int option, string surname, string name, string middle, int ident, int passport, int cad\_number, int amount, int pay, int card)

{

cout << endl;

cout << endl;

cout << "^------------------- Заява про реєстрацiю права власностi на земельну дiлянку --------------------^" << endl;

cout << endl;

if (option == 1)

{

cout << "Прiзвище: Muzalevska" << endl;

cout << "Iм'я: Aliona" << endl;

cout << "По-батьковi: Dmitrivna" << endl;

cout << "Номер паспорту: 000437589"<< endl;

cout << "Iдентифiкацiйний код: 123456789123" << endl;

}

else if (option == 2)

{

cout << "Прiзвище: " << name << endl;

cout << "Iм'я: " << surname << endl;

cout << "По-батьковi: " << middle << endl;

cout << "Номер паспорту: " << passport << endl;

cout << "Iдентифiкацiйний код: " << ident << endl;

}

cout << endl;

cout << "1) Кадастровий номер земельної дiлянки, право власностi на яку ви реєструєте: " << endl;

cout << endl;

cout << cad\_number << endl;

cout << endl;

cout << "2) Строк реєстрацiї права власностi та розмiр адмiнiстративного збору:" << endl;

cout << endl;

if (amount == 1)

{

cout << "Строк, що не перевищує 5 робочих днiв - 230 UAH" << endl;

}

else if (amount == 2)

{

cout << "Строк два робочi днi - 2270 UAH" << endl;

}

else if (amount == 3)

{

cout << "Строк один робочий день - 4540 UAH" << endl;

}

else if (amount == 4)

{

cout << "Строк 2 години - 11350 UAH" << endl;

}

cout << endl;

cout << "3) Вид сплати: " << endl;

cout << endl;

if (pay == 1)

{

cout << "Сплата готiвкою в одному з центрiв ЦНАП вашого мiста." << endl;

}

else if (pay == 2)

{

cout << "Сплата банкiвською картою на офiцiйному сайтi ЦНАП." << endl;

cout << "Номер карти: " << card << endl;

}

cout << endl;

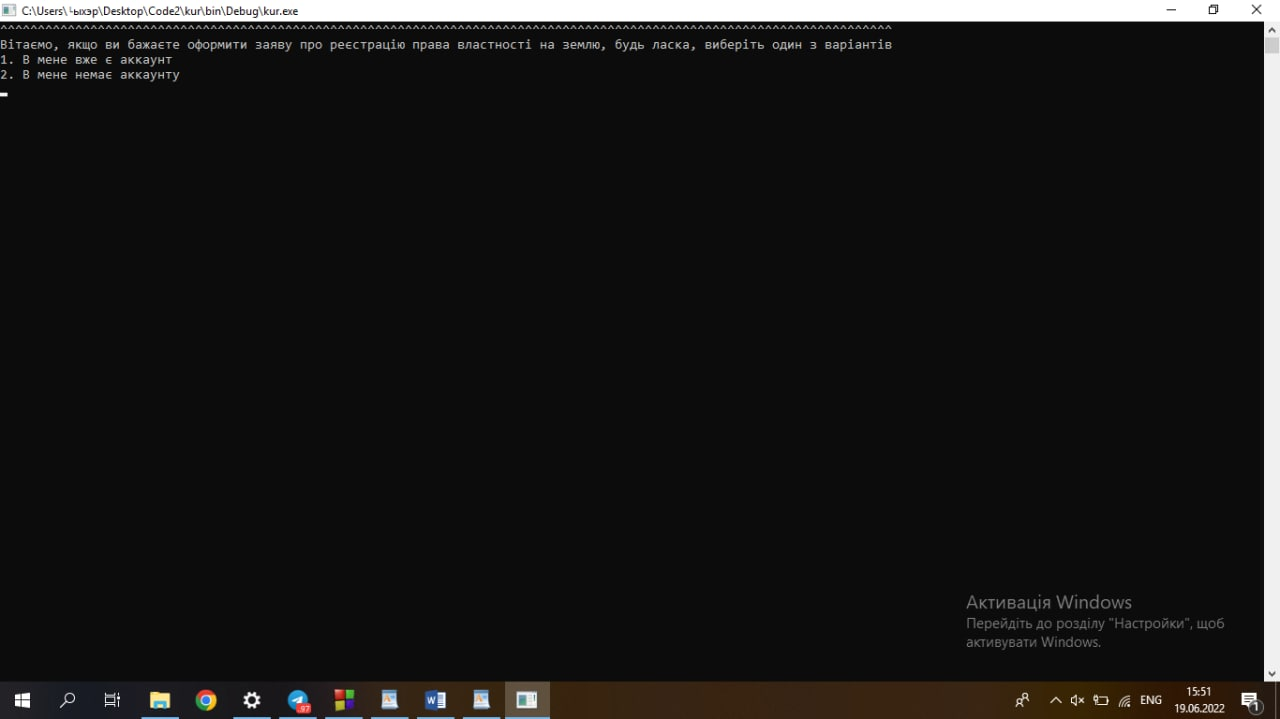
cout << endl;

}

};

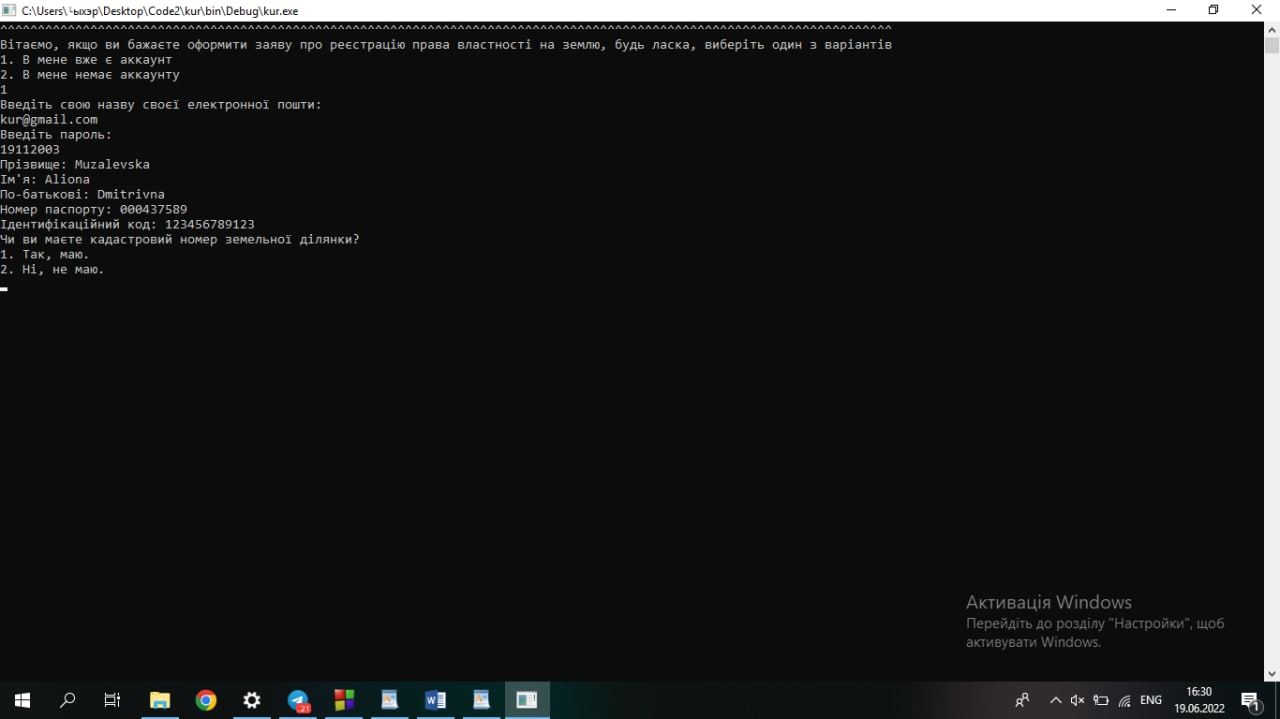
# **Інтерфейс користувача системою отримання права власності на земельну ділянку**

Початковий екран. Програма реалізована так, ніби користувач вперший раз користується нею:



*Додаток 1*

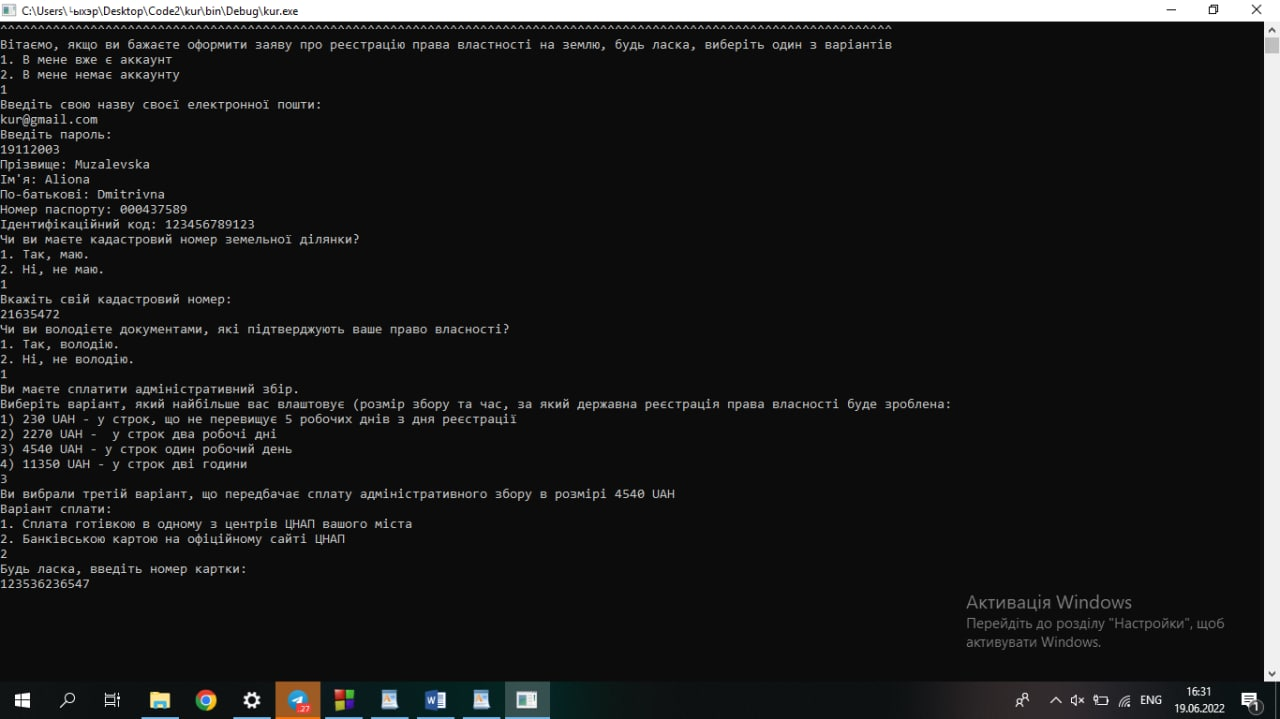
Користувач має два варіанти вибору: 1 – якщо він має аккаунт, увійти за допомогою електронної пошти та пароля, 2 – якщо він новий користувач, ввести свої дані вручну. Розглянемо перший випадок:



*Додаток 2*

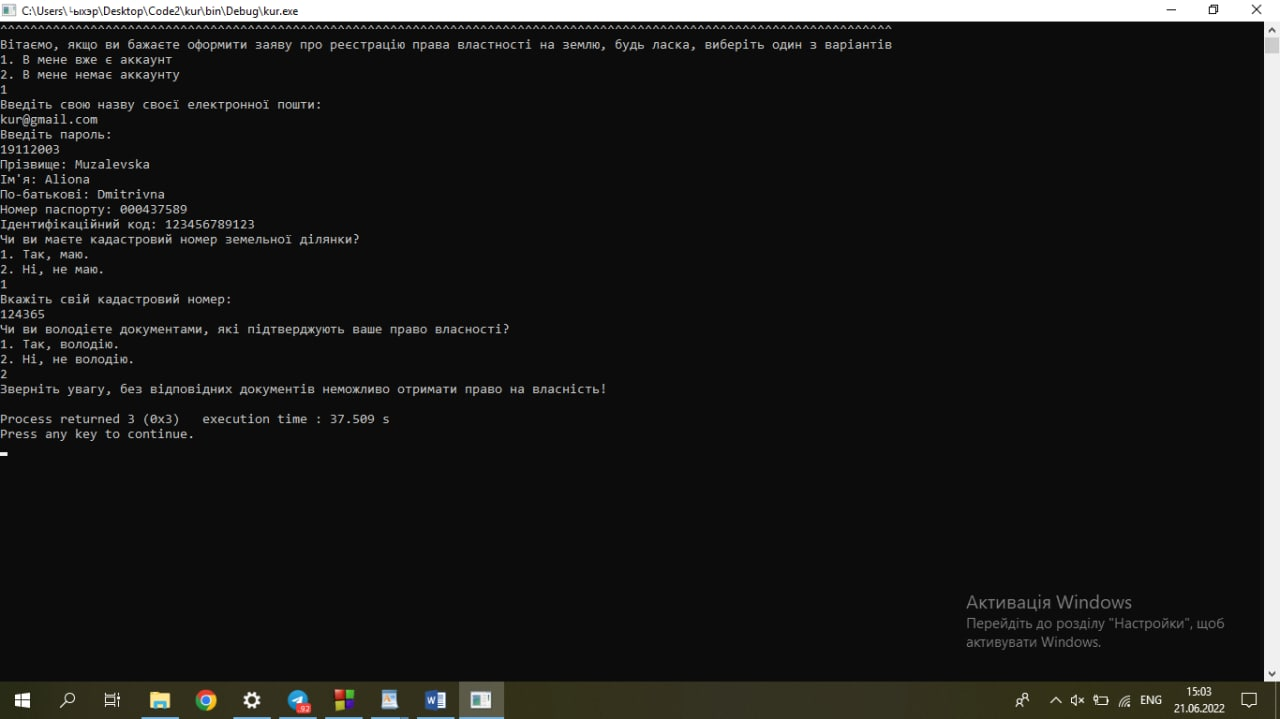
Розглянемо другий випадок:

Перевірка користувача на наявність кадастрового номера та документа, що підтверджує отримання права власності на земельну ділянку (використовую перший випадок вибору):



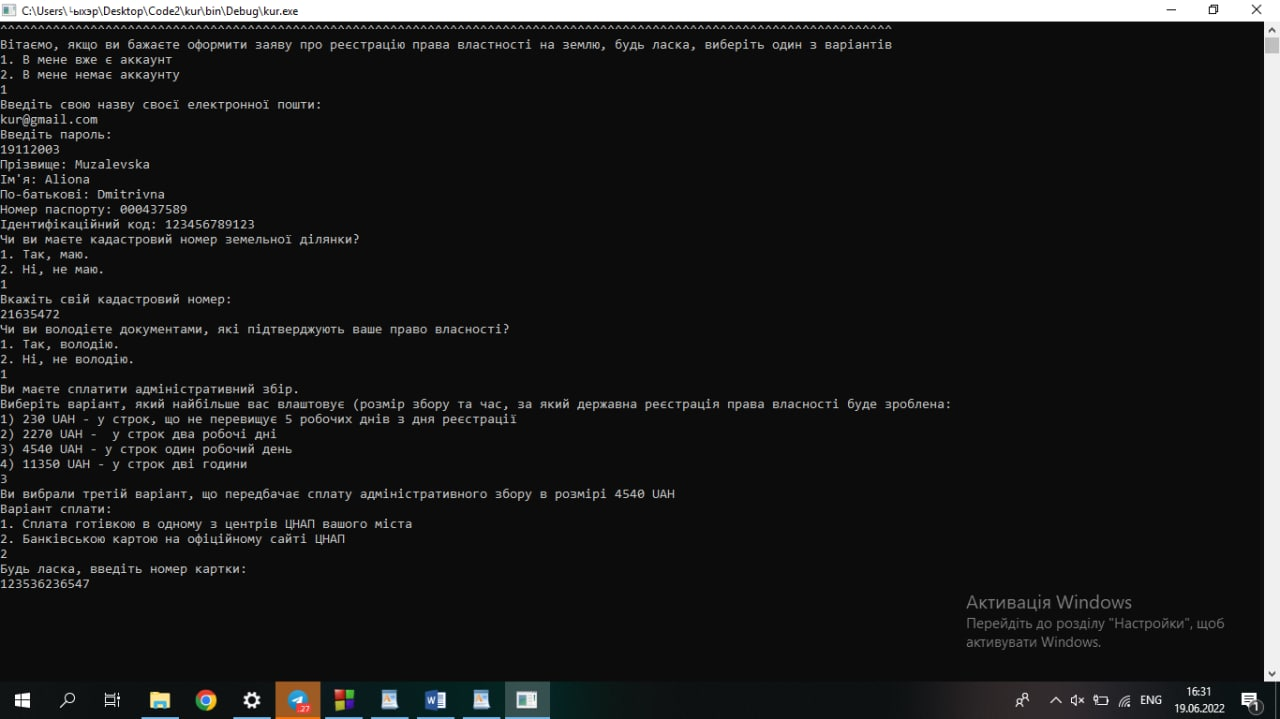
*Додаток 3*

Користувач підтвердив, що має кадастровий номер та ввів його. Також підтвердив, що документи, які підтверджують право на оформлення ділянки також є. Якщо наприклад написати цифру 2 в одній з тих перевірок система закінчить своє виконання на попередить про відсутність кадастрового номера чи документів відповідно.



*Додаток 4*

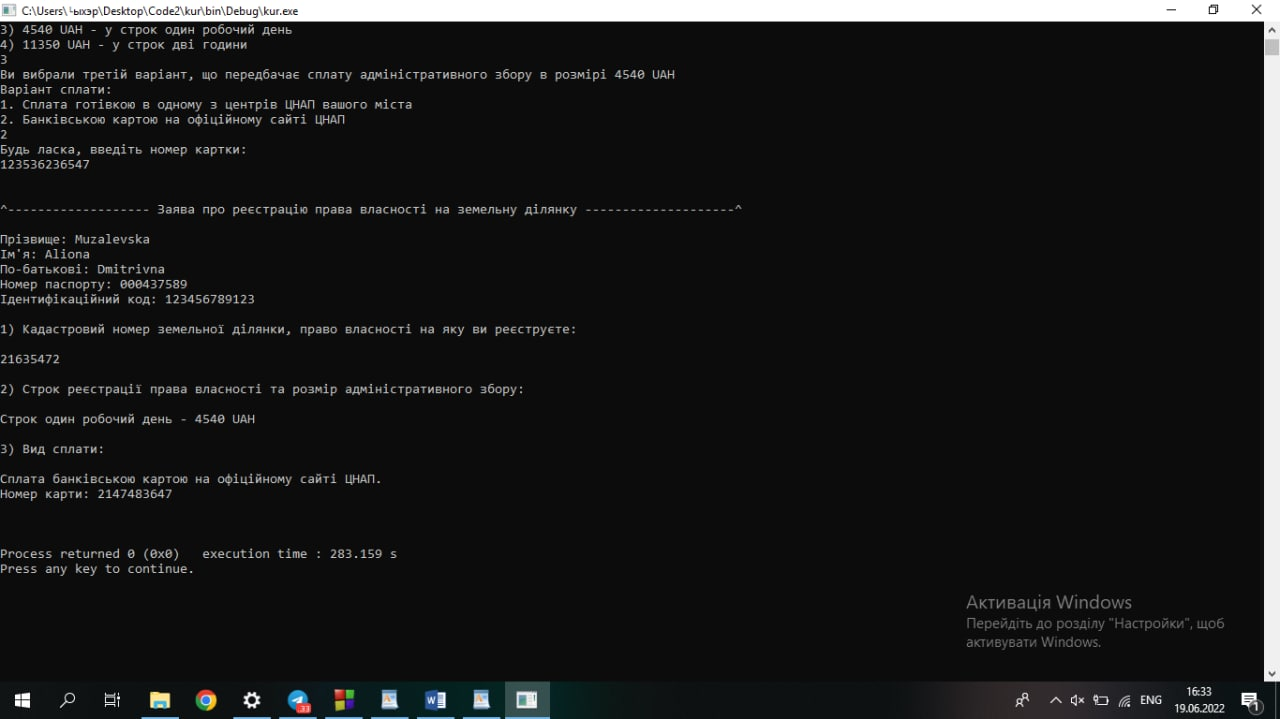
Вибір терміну та розміру адміністративного збору, вибір способу оплати:



*Додаток 5*

Користувач обрав варіант, тобто оформлення заяви протягом одного робочого дня за ціною 4540 грн. Сплачувати адміністративний збір буде банківською карткою, тому програма просить ввести номер картки користувача. Якщо користувач захоче вибрати варіант сплати готівкою його попередять, що це треба зробити в одному з центрів ЦНАП його міста.

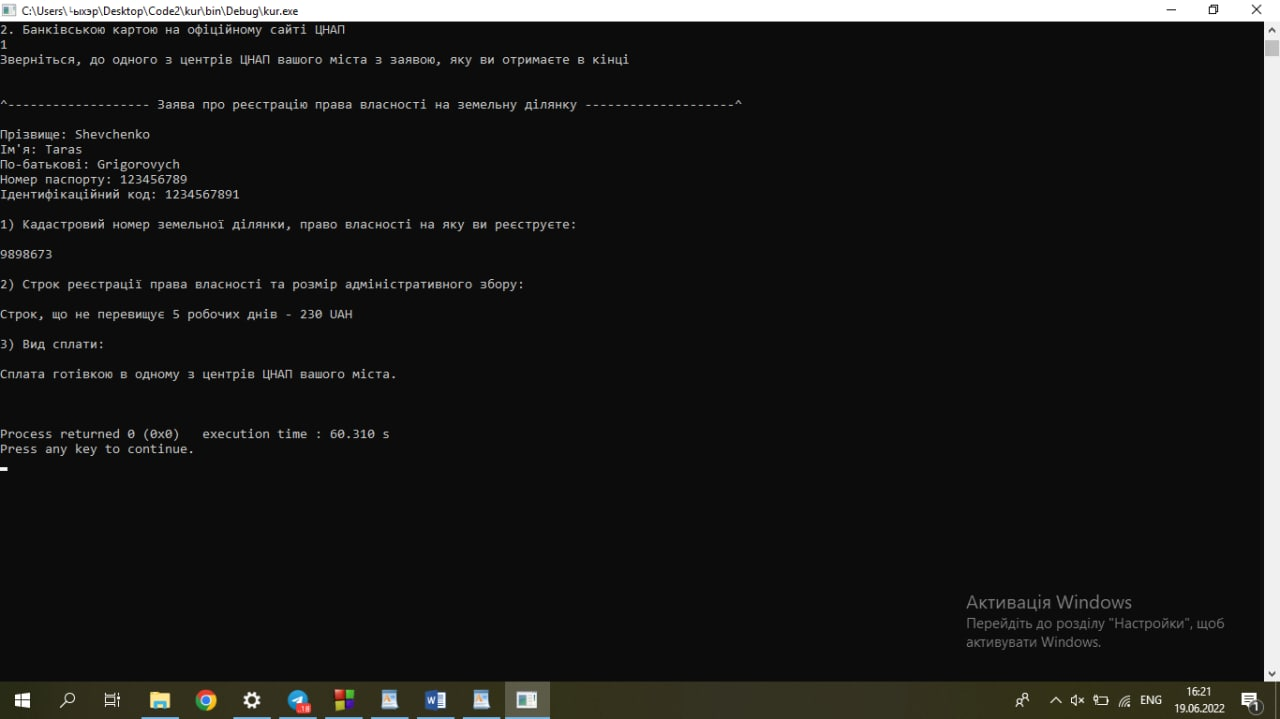
Результат програми – заява про реєстрацію права власності на земельну ділянку. Маємо ті самі два випадки: коли користувач був авторизований та коли самостійно вводить персональні дані. Розглянемо перший випадок:



*Додаток 6*

Тут нам вивело дані користувача, який був зареєстрований в системі, введений ним кадастровий номер ділянки, термін оформлення, розмір адміністративного збору та спосіб оплати.

Другий випадок (користувач самостійно вводив дані) має такий вигляд:



*Додаток 7*

# **Висновки**

Отож, можемо підвести підсумок, що в результаті виконання курсової роботи ми створили робочу модель системи отримання права власності на земельну ділянку у державі Україна, яка був створений за допомогою об'єктно-орієнтованої парадигми, на базі мови програмування С++.

Завдяки детальному вивченню предметної області та набутих знань вдалося застосувати основні інструменти об'єктно-орієнтованого програмування на практиці та удосконалити власні навички. Раціональне розбиття на класи покращило використання принципів об’єктно-орiєнтованого програмування (ООП). Програма має в собі схожості до реально існуючою системи, що засвідчує її функціональність.

Дана курсова робота допомогла мені краще розібратися в основах об’єктно-орієнтованому програмуванні, дала змогу продемонструвати набуті знання на практиці, поглибити їх.

# **Список використаних джерел**

1. Загальний огляд технологій об'єктно-орієнтованого програмування. [Електронний ресурс] – Режим доступу до ресурсу: <https://studopedia.su/20_51386_viznachennya-ponyattya-obiektno-oriientovanogo-programuvannya.html>
2. Конструктор і деструктор класу в C ++ [Електронний ресурс] – Режим доступу до ресурсу: <https://purecodecpp.com/uk/archives/1764>.
3. Побудова та використання класів [Електронний ресурс] – Режим доступу до ресурсу: <https://revolution.allbest.ru/programming/01183476_0.html>
4. Як самостійно оформити право власності на земельну ділянку [Електронний ресурс] – Режим доступу до ресурсу: <https://cnap-slavutich.com.ua/news/813-yak-samostiyno-ofor...-na-zemelnu-dilyanku>
5. Як оформити свої права на отриману у спадок земельну ділянку, якщо їх не присвоєно кадастровий номер[Електронний ресурс] – Режим доступу до ресурсу: <https://sud.ua/ru/news/sudebnaya-praktika/137734-yak-oformiti-svoyi-prava-na-zemelnu-dilyanku-otrimanu-u-spadok-yakscho-zemelniy-dilyantsi-ne-prisvoyeno-kadastrovogo-nomeru>

# **Лістинг програми**

#include <iostream>

#include <string>

#include <stdlib.h>

using namespace std;

class Person

{

protected:

int findcad;

public:

virtual void Choice()

{

cout << "Виберiть варiант" << endl;

cin >> findcad;

}

};

class Authorisation : public Person

{

public:

int option;

string email;

string password;

string name;

string surname;

string middle;

int passport;

int ident;

void Choice()

{

cout << "1. В мене вже є аккаунт" << endl;

cout << "2. В мене немає аккаунту" << endl;

cin >> option;

}

void For\_old\_ac()

{

cout << "Введiть свою назву своєї електронної пошти: " << endl;

while (cin >> email)

{

if (email != "kur@gmail.com")

{

cout << "Пошта не вiрна, перевiрте правильнiсть" << endl;

}

else

{

break;

}

}

cout << "Введiть пароль: " << endl;

while (cin >> password)

{

if (password != "19112003")

{

cout << "Пароль не вiрний, перевiрте правильнiсть" << endl;

}

else

{

break;

}

}

}

void Old\_ac()

{

cout << "Прiзвище: Muzalevska" << endl;

cout << "Iм'я: Aliona" << endl;

cout << "По-батьковi: Dmitrivna" << endl;

cout << "Номер паспорту: 000437589" << endl;

cout << "Iдентифiкацiйний код: 123456789123" << endl;

}

void New\_ac()

{

cout << "Ваше прiзвище: " << endl;

cin >> surname;

cout << "Ваше iм'я: " << endl;

cin >> name;

cout << "Ваше iм'я по-батьковi: " << endl;

cin >> middle;

cout << "Ваш номер паспорту: " << endl;

cin >> passport;

cout << "Ваш iдентифiкацiйний код: " << endl;

cin >> ident;

}

void Account()

{

if (option == 1)

{

For\_old\_ac();

Old\_ac();

}

else if (option == 2)

{

New\_ac();

}

}

};

class Cadastral\_number : public Person

{

private:

class Permission

{

public:

int doc;

void Rights()

{

cout << "Чи ви володiєте документами, якi пiдтверджують ваше право власностi?" << endl;

cout << "1. Так, володiю." << endl;

cout << "2. Нi, не володiю." << endl;

cin >> doc;

if (doc == 1)

{

cout << "Ви маєте сплатити адмiнiстративний збiр." << endl;

}

else

{

cout << "Звернiть увагу, без вiдповiдних документiв неможливо отримати право на власнiсть!" << endl;

abort();

}

}

};

public:

int amount;

int fee;

int cad\_number;

void Admin\_fee()

{

cout << "Виберiть варiант, який найбiльше вас влаштовує (розмiр збору та час, за який державна реєстрацiя права власностi буде зроблена:" << endl;

cout << "1) 230 UAH - у строк, що не перевищує 5 робочих днiв з дня реєстрацiї" << endl;

cout << "2) 2270 UAH - у строк два робочi днi" << endl;

cout << "3) 4540 UAH - у строк один робочий день" << endl;

cout << "4) 11350 UAH - у строк двi години" << endl;

cin >> amount;

}

void Response()

{

if (amount == 1)

{

cout << "Ви вибрали перший варiант, що передбачає сплату адмiнiстративного збору в розмiрi 230 UAH" << endl;

fee = 230;

}

else if (amount == 2)

{

cout << "Ви вибрали другий варiант, що передбачає сплату адмiнiстративного збору в розмiрi 2270 UAH" << endl;

fee = 2270;

}

else if (amount == 3)

{

cout << "Ви вибрали третiй варiант, що передбачає сплату адмiнiстративного збору в розмiрi 4540 UAH" << endl;

fee = 4540;

}

else if (amount == 4)

{

cout << "Ви вибрали четвертий варiант, що передбачає сплату адмiнiстративного збору в розмiрi 11350 UAH" << endl;

fee = 11350;

}

}

void Choice()

{

cout << "Чи ви маєте кадастровий номер земельної дiлянки?" << endl;

cout << "1. Так, маю." << endl;

cout << "2. Нi, не маю." << endl;

cin >> findcad;

}

void Response\_cad()

{

if (findcad == 1)

{

cout << "Вкажiть свiй кадастровий номер: " << endl;

cin >> cad\_number;

}

else

{

cout << "Вам потрiбно отримати кадастровий номер. Бiльше детальної iнформацiю ви зможете знайти на офiцiйному сайтi Державного земельного кадастру." << endl;

abort();

}

}

Permission proper;

void Rights()

{

proper.Rights();

}

};

class Payment : public Authorisation

{

public:

int pay;

int card;

void Pay\_method()

{

cout << "1. Сплата готiвкою в одному з центрiв ЦНАП вашого мiста" << endl;

cout << "2. Банкiвською картою на офiцiйному сайтi ЦНАП" << endl;

cin >> pay;

if (pay == 1)

{

cout << "Звернiться, до одного з центрiв ЦНАП вашого мiста з заявою, яку ви отримаєте в кiнцi" << endl;

}

else if (pay == 2)

{

cout << "Будь ласка, введiть номер картки:" << endl;

cin >> card;

}

}

};

class Statement : public Authorisation, public Cadastral\_number, public Payment

{

public:

void Statement\_method(int option, string surname, string name, string middle, int ident, int passport, int cad\_number, int amount, int pay, int card)

{

cout << endl;

cout << endl;

cout << "^------------------- Заява про реєстрацiю права власностi на земельну дiлянку --------------------^" << endl;

cout << endl;

if (option == 1)

{

cout << "Прiзвище: Muzalevska" << endl;

cout << "Iм'я: Aliona" << endl;

cout << "По-батьковi: Dmitrivna" << endl;

cout << "Номер паспорту: 000437589" << endl;

cout << "Iдентифiкацiйний код: 123456789123" << endl;

}

else if (option == 2)

{

cout << "Прiзвище: " << name << endl;

cout << "Iм'я: " << surname << endl;

cout << "По-батьковi: " << middle << endl;

cout << "Номер паспорту: " << passport << endl;

cout << "Iдентифiкацiйний код: " << ident << endl;

}

cout << endl;

cout << "1) Кадастровий номер земельної дiлянки, право власностi на яку ви реєструєте: " << endl;

cout << endl;

cout << cad\_number << endl;

cout << endl;

cout << "2) Строк реєстрацiї права власностi та розмiр адмiнiстративного збору:" << endl;

cout << endl;

if (amount == 1)

{

cout << "Строк, що не перевищує 5 робочих днiв - 230 UAH" << endl;

}

else if (amount == 2)

{

cout << "Строк два робочi днi - 2270 UAH" << endl;

}

else if (amount == 3)

{

cout << "Строк один робочий день - 4540 UAH" << endl;

}

else if (amount == 4)

{

cout << "Строк 2 години - 11350 UAH" << endl;

}

cout << endl;

cout << "3) Вид сплати: " << endl;

cout << endl;

if (pay == 1)

{

cout << "Сплата готiвкою в одному з центрiв ЦНАП вашого мiста." << endl;

}

else if (pay == 2)

{

cout << "Сплата банкiвською картою на офiцiйному сайтi ЦНАП." << endl;

cout << "Номер карти: " << card << endl;

}

cout << endl;

cout << endl;

}

};

int main()

{

setlocale(LC\_ALL, "Ukrainian");

cout << "^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^" << endl;

cout << "Вiтаємо, якщо ви бажаєте оформити заяву про реєстрацiю права властностi на землю, будь ласка, виберiть один з варiантiв" << endl;

Authorisation auth;

Person\* city;

city = &auth;

city->Choice();

auth.Account();

Cadastral\_number cd;

city = &cd;

city->Choice();

cd.Response\_cad();

cd.Rights();

cd.Admin\_fee();

cd.Response();

cout << "Варiант сплати:" << endl;

Payment pay;

pay.Pay\_method();

Statement st;

st.Statement\_method(auth.option, auth.name, auth.surname, auth.middle, auth.ident, auth.passport, cd.cad\_number, cd.amount, pay.pay, pay.card);

}