

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОТЕХНОЛОГІЙ І  
ПРИРОДОКОРИСТУВАННЯ**

**Факультет інформаційних технологій**

**Кафедра комп'ютерних наук**

**Терія розпізнавання образів та класифікація в системах штучного  
інтелекту**

**Лабораторна робота №2**

**«Дослідження властивостей методу потенційних функцій при  
розпізнаванні образів»**

**(4 години)**

**Київ - 2025**

**Мета роботи:** Засвоєння базових знань щодо способів класифікації образів на основі методу Потенційних функцій. Розробка програмної системи, що реалізує розпізнавання образів на основі зазначеного методу. Дослідження розробленої системи. Отримання практичних навичок з розробки програмних систем розпізнавання образів..

**Підготовка до роботи:** Вивчити і уявити теоретичні відомості щодо способів класифікації образів на основі методу потенційних функцій.

**Завдання:**

1. Ознайомитись з методичними матеріалами до лабораторної роботи.
2. Розробити програмний додаток на мові програмування C/C++ в середовищі розробки MS Visual Studio, що реалізує алгоритм розпізнавання образів на основі методу потенційних функцій.
3. Дослідити властивості методу потенційних функцій за допомогою розробленого програмного додатку на прикладах рішення задач класифікації. В якості похідних даних при дослідженні властивостей методу використати приклад, що наведений в матеріалах методичної розробки до лабораторної роботи і довільний власний приклад, який було використано при дослідженні в лабораторній роботі №1.
4. За результатами досліджень скласти звіт з поясненням проміжних результатів процесу вирішення задач класифікації та сформулювати обґрунтовані висновки щодо отриманих результатів дослідження.
5. Порівняти і обґрунтувати результати проведеного дослідження з результатами дослідження, що були отримані при виконанні дослідження в лабораторній роботі №1.

**Зміст звіту:**

1. Назва та мета роботи.
2. Теоретичні відомості.
3. Опис програмного засобу.
4. Відкоментований похідний код програмного додатку для дослідження методу потенційних функцій на мові C/C++ в середовищі розробки MS Visual Studio.
5. Скрин-шоти роботи програмного додатку з поясненнями щодо отриманих проміжних результатів вирішення задач класифікації і обґрунтуванням отриманих результатів дослідження.
6. Порівняти і обґрунтувати результати проведеного дослідження з результатами дослідження, що були отримані в лабораторній роботі №1.

7. Сформулювати висновки щодо отриманих результатів дослідження методу потенційних функцій при вирішенні задач класифікації.

## Теоретичні відомості

У класичному підході розпізнавання образів, в якому невідомий об'єкт для класифікації представляється у вигляді вектора елементарних ознак. Система розпізнавання на основі ознак може бути розроблена різними способами. Ці вектори можуть бути відомі системі заздалегідь в результаті навчання або передбачені в режимі реального часу на основі будь-яких моделей.

Простий алгоритм класифікації полягає в угрупованні еталонних даних класу з використанням вектора математичного очікування класу (середнього.

### *Статистична інтерпретація*

Ідея методу потенціальних функцій має фізичну природу. Як відомо, згідно закону Кулона вплив заряду на точку убуває пропорційно квадрату відстані до неї. Отже, потенціал може використовуватися як оцінка віддаленості точки від заряду. У цьому розумінні метод потенціальних функцій є різновидом методу найближчого сусіди. Якщо поле утворене декількома зарядами, потенціал у кожній точці поля дорівнює сумі потенціалів, створюваних у цій точці кожним із зарядів. Якщо заряди, що утворюють поле, розташовані компактно, потенціал поля досягає максимального значення в центрі тяжіння множини точок і убуває в міру видалення від нього. Таким чином, якщо точку в просторі ознак інтерпретувати як заряд, то ці міркування можна формалізувати і звести до обчислення певної суми функцій, що описують потенціали зарядів — точок навчальної множини.

Розглянемо задачу бінарної класифікації. Кожному образу поставимо у однозначну відповідність точку в просторі ознак  $X$ . В подальшому будемо припускати, що класи  $K_1$  і  $K_2$  не перетинаються. Як правило, топологічні властивості простору ознак допускають застосування малої леми Урисона, з якої випливає, що в просторі ознак  $X$  існує неперервна функція  $\Phi$ , яка набуває значення більше нуля в точках класу  $K_1$  і менше нуля в точках класу  $K_2$ . Зауважимо, що таких функцій може бути багато (і навіть нескінченно багато).

В ході навчання для кожного образу  $R_k$ , якому відповідає точка  $x_k$ , обчислюється функція  $U(x, x_k)$ , яка називається *потенціальною*, а побудова роздільної функції  $\Phi$  за навчальною послідовністю образів  $A_1, A_2, \dots, A_k, \dots$  зводиться до побудови послідовності потенціальних функцій  $U(x, x_1), U(x, x_2), \dots, U(x, x_k), \dots$ , що збігаються до функції  $\Phi$ .

Метод потенціальних функцій ґрунтується на припущенні, що існує система функцій  $\phi_1, \phi_2, \dots, \phi_k, \dots$ , яка дозволяє для будь-яких двох диз'юнктних класів знайти таке число  $N$ , що функцію  $\Phi$  можна подати у вигляді

$$\Phi(x) = \sum_{k=1}^N c_k \varphi_k(x). \quad (1)$$

Як правило простір  $X$  є гільбертовим, тому в новому просторі існує повна система функцій  $\varphi_1, \varphi_2, \dots, \varphi_k, \dots$ , за допомогою яких функцію  $\Phi$  можна подати у вигляді ряду Фур'є

$$\Phi(x) = \sum_{k=1}^{\infty} c_k \varphi_k(x) \quad (2)$$

Прагнучи подати роздільчу функцію  $\Phi$  у вигляді суми (а не ряду), уведемо в розгляд  $N$ -вимірний простір  $Y$ , на який простір ознак  $X$  відображається за правилом  $y_k = \varphi_k(x)$ ,  $k=1, 2, \dots, N$ .

Таким чином, роздільча функція  $\Phi$  відображається в лінійну функцію  $\sum_{k=1}^N a_k y_k(x)$  і набуває значення більше нуля в точках класу  $K_1$  і менше нуля в точках класу  $K_2$ . Легко бачити, що в просторі  $Y$  функція  $\Phi$  є лінійною (відносно точок  $y$ ).

Потенціальна функція розглядається як функція двох векторів

$$\Phi(x) = \sum_{k=1}^{\infty} \alpha_k^2 \varphi_k(x) \varphi_k(x^*) \quad (3)$$

де  $\varphi_1, \varphi_2, \dots, \varphi_k, \dots$  — лінійно-незалежна система функцій;  $\alpha_k^2$  — дійсні числа, що не обертаються в нуль одночасно;  $x^*$  — точка, яка обчислюється в ході навчання.

Вважатимемо, що функції  $\varphi_k$  і  $U(x, x^*)$  обмежені на просторі ознак. Першому образу  $P_1$  ставимо у відповідність точку ознак  $x_1$ , за якою будується потенційна функція

$$\Phi_1(x) = \begin{cases} U(x, x_1), & \text{якщо } x \in K_1, \\ -U(x, x_1), & \text{якщо } x \in K_2. \end{cases} \quad (4)$$

На  $k$ -му кроці отримуємо потенціальну функцію  $\Phi_k$ .

На  $(k+1)$ -му кроці, обчислюючи потенціал в точці  $x_{k+1}$  можемо отримати наступні значення потенціалу:

1.  $x_{k+1} \in K_1 \Rightarrow \Phi_k(x_{k+1}) > 0 \Rightarrow \Phi_{k+1}(x) = \Phi_k(x)$ ,
2.  $x_{k+1} \in K_2 \Rightarrow \Phi_k(x_{k+1}) < 0 \Rightarrow \Phi_{k+1}(x) = \Phi_k(x)$ ,
3.  $x_{k+1} \in K_1 \Rightarrow \Phi_k(x_{k+1}) < 0 \Rightarrow \Phi_{k+1}(x) = \Phi_k(x) + \Phi(x, x_{k+1})$ ,
4.  $x_{k+1} \in K_2 \Rightarrow \Phi_k(x_{k+1}) > 0 \Rightarrow \Phi_{k+1}(x) = \Phi_k(x) - \Phi(x, x_{k+1})$ .

Таким чином,

$$\Phi_k(x) = \sum_{x_i^- \in V_1} U(x, x_i) + \sum_{x_j^+ \in V_2} U(x, x_j), \quad (5)$$

де  $x_i^-$  — точки з класу  $K_1$ , на яких класифікатор робить помилку.

Корекцію роздільної функції можна здійснювати по-різному, тому існують декілька алгоритмів, заснованих на методі потенціальних функцій.

**Варіант 1.** Вважатимемо, що істинне значення функції  $\Phi_{k+1}(x)$  в точці  $x_{k+1}$  відомо заздалегідь. Тоді корекцію роздільної функції можна здійснити так:

$$\Phi_{k+1}(x) = \Phi_k(x) + \alpha_{k+1} \operatorname{sgn}[\Phi(x_{k+1}) - \Phi_k(x_{k+1})]U(x, x_{k+1}) \quad (6)$$

где  $\Phi(x_{k+1})$  — істинне значення роздільної функції в точці  $x_{k+1}$ ;  $\alpha_k$  — будь-яка послідовність чисел, що задовольняє умовам:

$$\alpha_k \rightarrow 0, \sum_{k=1}^{\infty} \alpha_k = \infty, \sum_{k=1}^{\infty} \alpha_k^2 < \infty.$$

Інакше кажучи,  $\{\alpha_k\} \in (c_0 \cap l_2) \setminus l_1$ .

**Теорема 1.** Нехай  $x_k$  — послідовність незалежних випадкових точок з простору ознак  $X$ ,  $P(x_i)$  — ймовірність появи точки  $x_i$ , а  $\Phi(x)$  — функція, що має вигляд  $\Phi(x) = \sum_{k=1}^N c_k \phi_k(x)$ . Тоді послідовність функцій  $\Phi_k(x)$ ,  $k=1,2,\dots$ , що задаються рекурентними співвідношеннями (6) задовольняють умову

$$P \left\{ \lim_{k \rightarrow \infty} \int_X |\Phi(x) - \Phi_k(x)| P(x) dx = 0 \right\} = 1. \quad (7)$$

**Варіант 2.** Другий варіант виглядає так.

$$\Phi_0(x) = 0,$$

$$\Phi_{k+1}(x) = \Phi_k(x) + \frac{1}{\lambda} [\Phi(x_{k+1}) - \Phi_k(x_{k+1})]U(x, x_{k+1}) \quad (8)$$

де

$$\lambda = \frac{1}{2} \max U(x, x^*)$$

**Теорема 2.** За умов теореми 1 послідовність функцій  $\Phi_k(x)$ ,  $k=1,2,\dots$ , що визначаються співвідношенням (8) задовольняє умову

$$P \left\{ \lim_{k \rightarrow \infty} \int_X (\Phi(x) - \Phi_k(x))^2 P(x) dx = 0 \right\} = 1. \quad (9)$$

Ті ж формули можна використати для послідовного наближення коефіцієнтів  $c_k$  в поданні (1).

**Варіант 1.**

$$c_k^{(n+1)} = c_{k+1}^{(n)} + \alpha_{k+1} \operatorname{sgn} \left[ \Phi(x_{k+1}) - \sum_{k=1}^N c_k \phi_k(x_{k+1}) \right] \phi_k(x_{k+1}), \quad i=1,2,\dots,n$$

**Варіант 2.**

$$c_{k+1}^{(n+1)} = c_{k+1}^{(n)} + \frac{1}{\lambda} \left[ \Phi(x_{k+1}) - \sum_{k=1}^N c_k \phi_k(x_{k+1}) \right] \phi_k(x_{k+1}), \quad i=1,2,\dots,n$$

## Геометрична інтерпретація

Нехай в просторі  $X$  існує функція  $\Phi(x)$ , що розділяє множини  $A \subset K_1$  і  $B \subset K_2$  і задовольняє умови (1) і (2). Тоді в просторі  $Y$  існує роздільна площина  $\Gamma$ , що проходить через початок координат із напрямлюючим вектором  $\vec{c}$ . Відобразимо множину  $A$  симетрично відносно початку координат у множину  $B'$  (тобто замінимо кожний вектор  $x$  вектором  $-x$ ) і отримаємо множину  $S = A \cup B'$ . За припущенням множини  $A$  і  $B'$  розділяються площиною  $\Gamma$ , тобто множина  $S$  лежить по один бік від площини  $\Gamma$ . Розглянемо послідовності точок  $M = \{x_1, x_2, \dots, x_n, \dots\}$  з простору  $X$  і їх образи  $M^* = \{y_1, y_2, \dots, y_n, \dots\}$  в просторі  $Y$ .

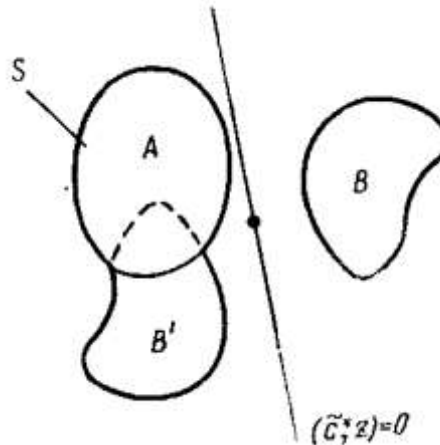


Рис. 1. Роздільна площина [1]

Потенціальна функція у просторі  $Y$  може бути подана як

$$U(Z, Z^*) = ZZ^*, \quad (10)$$

де  $Z = \{z_1, z_2, \dots\} = \{\alpha_1 \phi_1(x), \alpha_2 \phi_2(x), \dots\}$  і  $Z^* = \{z_1^*, z_2^*, \dots\} = \{\alpha_1 \phi_1(x^*), \alpha_2 \phi_2(x^*), \dots\}$ .

Отже, співвідношення (5) можна переписати як

$$\Phi_k(Z) = \sum_{Z^{p^{(k)}} \in M^*} (Z, Z^p) \quad (11)$$

Виправлення помилки відбувається тоді, коли  $\Phi_k(Z) < 0$ . Отже, корекція помилки на  $(k+1)$ -му кроці відбувається, коли

$$z_{k+1} \sum_{m=1}^k z_m < 0. \quad (12)$$

Таким чином, перша точка  $Z^{(1)}$  із множини  $M^*$  призводить до побудови площини  $U_1(Z) = (Z, Z^1) = 0$  із напрямлюючим вектором  $Z^{(1)}$ . Якщо наступна точка з множини  $M^*$  лежить у тому ж підпросторі, що й напрямлюючий вектор  $Z^1$ , то помилка відсутня і роздільна площина не уточнюється. Якщо ж наступна точка потрапляє у протилежний підпростір, відбувається корекція. При цьому попередній напрямлюючий вектор складається із вектором точки, що вимагала корекції, і їхня сума береться як новий напрямлюючий вектор.

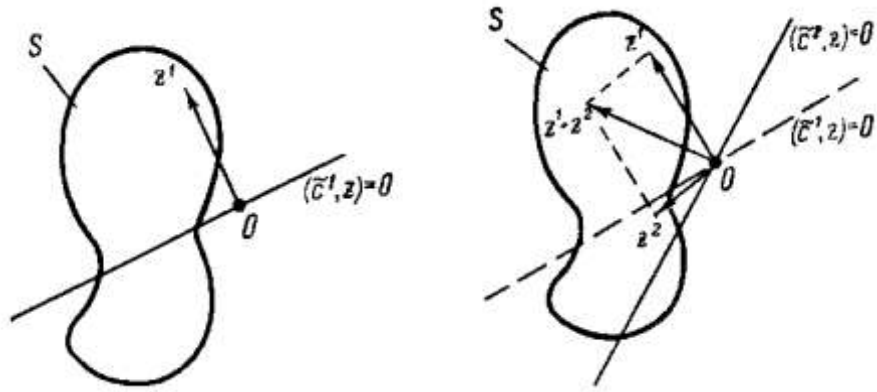


Рис. 2. Корекція роздільної площини [1]

Після  $k$  корекцій напрямлюючий вектор дорівнює  $\sum_{m=1}^k Z_m$  і площина, що проходить через початок координат, приймається як роздільна.

### Приклад

Нехай образи, що характеризуються векторами ознак, задаються таблицею

	X1	X2	X3	X4
Зріст	180	60	190	40
вага	90	20	100	10

Нехай образи еталони, необхідні для навчання системи розпізнавання задаються векторами ознак. Обчислимо середні значення Сер за кожною ознакою.

Об'єкти для навчання (належать $\Omega_1$ )					
$\Omega_1$	Y1	Y2	Y3	Y4	Сер
Зріст	180	170	200	189	184,75
вага	90	75	110	80	88,75

Об'єкти для навчання (належать $\Omega_2$ )					
$\Omega_2$	Y1	Y2	Y3	Y4	Сер
Зріст	100	70	90	40	75
вага	50	20	30	20	30

$$\begin{aligned} & \Gamma_{\Omega_1} \quad \Gamma_{\Omega_2} \\ & [(184,75-180)^2 + (88,75-90)^2]^{1/2} = 4,912 \quad [(75-180)^2 + (30-90)^2]^{1/2} = 120,9 \\ & [(184,75-60)^2 + (88,75-20)^2]^{1/2} = 142,4 \quad [(75-60)^2 + (30-20)^2]^{1/2} = 18,03 \\ & [(184,75-190)^2 + (88,75-100)^2]^{1/2} = 12,4 \quad [(75-190)^2 + (30-100)^2]^{1/2} = 134,6 \\ & [(184,75-40)^2 + (88,75-10)^2]^{1/2} = 164,8 \quad [(75-40)^2 + (30-10)^2]^{1/2} = 40,31 \end{aligned}$$

В якості потенційної функції виберемо

$$f = f_0 / (1 + (ar)^2),$$

де  $a = 0,01$ ;  $f_0 = 1$ ;  $r$  - відстань об'єкту до еталонного (середнього) об'єкта класу.  $\Gamma_{\Omega_1}$ ,  $\Gamma_{\Omega_2}$  - відстань від об'єкта до еталонного об'єкта класу  $\Omega_1$ ,  $\Omega_2$ .

$f_{\Omega_1}$ ,  $f_{\Omega_2}$  - потенційні функції для класів  $\Omega_1$ ,  $\Omega_2$ .

$\Gamma_{\Omega_1}$		$\Gamma_{\Omega_2}$
4,912		120,9
142,4		18,03
12,4		134,6
164,8		40,31
$f_{\Omega_1}$		$f_{\Omega_2}$
0,997593	$\Omega_1$	0,406226477
0,330275	$\Omega_2$	0,968515414
0,984857	$\Omega_1$	0,355654696
0,269114	$\Omega_2$	0,860222742

Метод потенційних функцій є різновидом методу найближчих сусідів, в якому вага класу сусіда залежить від його відстані до даної точки. Для найближчих сусідів або для всіх прикладів даного класу обчислюємо значення функції:

$$\phi_k = \frac{1}{N_k} \sum_{i=1}^{N_k} \phi(\mathbf{x} - \mathbf{x}_i), \quad \mathbf{x} \in \text{класу } k,$$

$\mathbf{x} \in$  класу  $k$ , де підсумовування ведеться за прикладами  $k$ -того класу, а в якості "потенціалів", що створюються кожним прикладом можуть виступати функції ( $a, \phi_0$ - параметри).

Часто використовуються такі потенційні функції

$$\phi(r) = \frac{\phi_0}{1 + (ar)^2}, \quad \phi(r) = \phi_0 e^{-(ar)^2}, \quad \phi(r) = \frac{\phi_0}{e^{-ar} + e^{ar}}.$$

## ***Варіанти завдань***

Класифікувати довільні реальні образи (об'єкти) з лабораторної роботи №1.

## **Додаток А**

### ***Приклад програмного коду для дослідження***

**Мета:** Засвоєння базових знань щодо способів класифікації образів на основі методу Потенційних функцій. Розробка програмної системи, що реалізує розпізнавання образів на основі зазначеного методу. Дослідження розробленої системи. Отримання практичних навичок з розробки програмних систем розпізнавання образів.

**Завдання(код програми):**

```
#include
<iostream>
#include <vector>
#include
<Windows.h>
#include
<fstream>
#include
<iomanip> using
namespace std;
class Categories // Клас категорій країн за розвиненістю
{
public:
double GDP; // ВВП
double Index_HD; //
ІЛР
double Level_Education; // Рівень
освіти double Level_Medicine; // Рівень
медицини double Index_Corruption; //
Індекс корупції double Budget; //
Бюджет
// Параметризований конструктор базового класу:
Categories(double Sign1 = 0.0, double Sign2 = 0.0, double Sign3 = 0.0, double Sign4 = 0.0,
double Sign5 = 0.0,
double Sign6 = 0.0) :
GDP(Sign1), Index_HD(Sign2), Level_Education(Sign3), Level_Medicine(Sign4),
Index_Corruption(Sign5), Budget(Sign6)
{
```

```

}
// Функція для визначення середнього значення для кожної з ознак категорії на
основі векторів еталонних даних :
void Avg_Vector(vector<Categories>& vector1, double& Avg1, double& Avg2,
double& Avg3, double& Avg4, double& Avg5, double& Avg6)
{
Avg1 = Avg2 = Avg3 = Avg4 = Avg5 = Avg6 = 0.0;
for (const auto& i : vector1)
{
Avg1 += i.GDP;
Avg2 += i.Index_HD;
Avg3 +=
i.Level_Education;
Avg4 +=
i.Level_Medicine; Avg5
+= i.Index_Corruption;
Avg6 += i.Budget;
}
Avg1 /= 6;
Avg2 /= 6;
Avg3 /= 6;
Avg4 /= 6;
Avg5 /= 6;
Avg6 /= 6;
}
// Функція визначення евклідової відстані:
void Recognition(vector<double>Under_Avg, vector<double>Med_Avg,
vector<double>Well_Avg, vector<double>High_Avg,
vector<Categories>Vector_Value, double& x1, double& x2, double& x3, double& x4)
{
for (int i = 0; i < Vector_Value.size(); i++)
{
x1 = sqrt(pow((Under_Avg[0] - Vector_Value[i].GDP), 2) + pow((Under_Avg[1] -
Vector_Value[i].Index_HD), 2) +
pow((Under_Avg[2] - Vector_Value[i].Level_Education), 2) +
pow((Under_Avg[3] - Vector_Value[i].Level_Medicine), 2) +
pow((Under_Avg[4] - Vector_Value[i].Index_Corruption), 2) +
pow((Under_Avg[5] - Vector_Value[i].Budget), 2));
x2 = sqrt(pow((Med_Avg[0] - Vector_Value[i].GDP), 2) + pow((Med_Avg[1] -
Vector_Value[i].Index_HD), 2) +
pow((Med_Avg[2] - Vector_Value[i].Level_Education), 2) +
pow((Med_Avg[3] - Vector_Value[i].Level_Medicine), 2) +
pow((Med_Avg[4] - Vector_Value[i].Index_Corruption), 2) +
pow((Med_Avg[5] - Vector_Value[i].Budget), 2));
x3 = sqrt(pow((Well_Avg[0] - Vector_Value[i].GDP), 2) +
pow((Well_Avg[1] - Vector_Value[i].Index_HD), 2) +

```

```

pow((Well_Avg[2] - Vector_Value[i].Level_Education), 2) + pow((Well_Avg[3] -
Vector_Value[i].Level_Medicine), 2) +
pow((Well_Avg[4] - Vector_Value[i].Index_Corruption), 2) + pow((Well_Avg[5] -
Vector_Value[i].Budget), 2));
x4 = sqrt(pow((High_Avg[0] - Vector_Value[i].GDP), 2) + pow((High_Avg[1] -
Vector_Value[i].Index_HD), 2) +
pow((High_Avg[2] - Vector_Value[i].Level_Education), 2) +
pow((High_Avg[3] - Vector_Value[i].Level_Medicine), 2) +
pow((High_Avg[4] - Vector_Value[i].Index_Corruption), 2) +
pow((High_Avg[5] - Vector_Value[i].Budget), 2));
}
cout << "\n" << x1 <<
endl; cout << x2 <<
endl;
cout << x3 <<
endl; cout << x4
<< "\n\n";
}
// Визначення значення за потенційною функцією:
void PotencialFunction(double r1, double r2, double r3, double r4, double& res_one,
double& res_two,
double& res_three, double& res_four)
{
const int f0 = 1;
const double a = 0.01;
res_one = f0 / (1 + pow((a * r1),
2)); res_two = f0 / (1 + pow((a *
r2), 2)); res_three = f0 / (1 +
pow((a * r3), 2)); res_four = f0 /
(1 + pow((a * r4), 2));
cout << "\n" << setprecision(20) << res_one <<
endl; cout << setprecision(20) << res_two <<
endl;
cout << setprecision(20) << res_three <<
endl; cout << setprecision(20) <<
res_four << "\n\n";
}
};
class UnderDeveloped : public Categories // Похідний клас малорозвиненої категорії :
{
public:
string Name_Country; // Назва країни
// Параметризований конструктор для даного класу:
UnderDeveloped(string Name_Country = "", double Sign1 = 0.0, double
Sign2 = 0.0, double Sign3 = 0.0, double Sign4 = 0.0, double Sign5 = 0.0,
double Sign6 = 0.0) : Name_Country(Name_Country), Categories(Sign1,

```

```

Sign2, Sign3, Sign4, Sign5, Sign6) {}
};
class MediumDeveloped : public Categories
{
public:
string Name_Country; // Назва країни
// Параметризований конструктор для даного класу:
MediumDeveloped(string Name_Country = "", double Sign1 = 0.0,
double Sign2 = 0.0, double Sign3 = 0.0, double Sign4 = 0.0, double Sign5
= 0.0,
double Sign6 = 0.0) : Name_Country(Name_Country), Categories(Sign1,
Sign2, Sign3, Sign4, Sign5, Sign6) {}
};
class WellDeveloped : public Categories
{
public:
string Name_Country; // Назва країни
// Параметризований конструктор для даного класу:
WellDeveloped(string Name_Country = "", double Sign1 = 0.0,
double Sign2 = 0.0, double Sign3 = 0.0, double Sign4 = 0.0, double
Sign5 = 0.0,
double Sign6 = 0.0) : Name_Country(Name_Country), Categories(Sign1,
Sign2, Sign3, Sign4, Sign5, Sign6) {}
};
class HighlyDeveloped : public Categories
{
public:
string Name_Country; // Назва країни
// Параметризований конструктор для даного класу:
HighlyDeveloped(string Name_Country = "", double Sign1 = 0.0, double
Sign2 = 0.0, double Sign3 = 0.0, double Sign4 = 0.0, double Sign5 = 0.0,
double Sign6 = 0.0) : Name_Country(Name_Country), Categories(Sign1,
Sign2, Sign3, Sign4, Sign5, Sign6) {}
};
int main()
{
system("chcp 1251");
cout << fixed; // Маніпулятор для виведення повного числа
HANDLE ColorText = GetStdHandle(STD_OUTPUT_HANDLE); // Дескриптор вікна
консолі
// Змінні для значення загальних відстаней від об'єктів до
категорій double x1 = 0.0;
double x2 =
0.0; double
x3 = 0.0;
double x4 =

```

```

0.0;
// Об'єкти класів для їх ініціалізації вказаними значеннями ознак:
UnderDeveloped Class_Und;
MediumDeveloped
Class_Med;
WellDeveloped
Class_Well;
HighlyDeveloped
Class_High;
// Змінні для збереження введених значень ознак
категорії string Name_Country = "";
double GDP = 0.0;
double Index_HD =
0.0;
double Level_Education =
0.0; double Level_Medicine =
0.0; double Index_Corruption
= 0.0; double Budget = 0.0;
// Вектори категорій для еталонних даних ознак:
vector<Categories> Vector_Under;
vector<Categories> Vector_Medium;
vector<Categories> Vector_Well;
vector<Categories> Vector_Highly;
// Додання еталонних об'єктів з відповідними значеннями ознак до векторів
категорій: Vector_Under.push_back(Categories(1259, 0.510, 0.6, 5.7, 9, 7100000000));
Vector_Under.push_back(Categories(520, 0.511, 0.38, 6.1, 16, 7000000000));
Vector_Under.push_back(Categories(463, 0.528, 0.7, 5.3, 15, 1320000000));
Vector_Under.push_back(Categories(572, 0.486, 0.41, 4.8, 24, 3460000000));
Vector_Under.push_back(Categories(857, 0.485, 0.5, 5.9, 24, 8400000000));
Vector_Under.push_back(Categories(720, 0.485, 0.58, 5.1, 26, 2500000000));
Vector_Medium.push_back(Categories(9126, 0.820, 0.75, 7.5, 45, 17000000000));
Vector_Medium.push_back(Categories(9296, 0.759, 0.65, 6.2, 40, 30000000000));
Vector_Medium.push_back(Categories(2213, 0.645, 0.7, 5.8, 40, 40000000000));
Vector_Medium.push_back(Categories(10204, 0.824, 0.8, 7.0, 38, 40000000000));
Vector_Medium.push_back(Categories(4562, 0.718, 0.7, 6.5, 37, 17000000000));
Vector_Medium.push_back(Categories(3888, 0.718, 0.8, 5.9, 34, 10000000000));
Vector_Well.push_back(Categories(52559, 0.947, 0.91, 9.2, 80, 180000000000));
Vector_Well.push_back(Categories(48947, 0.929, 0.88, 9.0, 78, 730000000000));
Vector_Well.push_back(Categories(67433, 0.938, 0.98, 9.5, 85, 120000000000));
Vector_Well.push_back(Categories(56667, 0.944, 0.88, 9.1, 77, 600000000000));
Vector_Well.push_back(Categories(76192, 0.957, 0.96, 9.3, 84, 240000000000));
Vector_Well.push_back(Categories(45484, 0.919, 0.8, 9.0, 65, 140000000000));
Vector_Highly.push_back(Categories(63416, 0.926, 0.89, 8.5, 67, 580000000000));
Vector_Highly.push_back(Categories(85882, 0.955, 0.93, 9.0, 76, 865000000000));
Vector_Highly.push_back(Categories(60959, 0.944, 0.92, 9.2, 82, 392000000000));
Vector_Highly.push_back(Categories(76192, 0.957, 0.95, 9.3, 84, 240000000000));

```

```
Vector_Highly.push_back(Categories(56667, 0.944, 0.90, 9.1, 77, 600000000000));
Vector_Highly.push_back(Categories(63020, 0.947, 0.89, 9.5, 88, 1760000000000));
// Змінні для збереження середнього значення кожної кількісної ознаки
категорії double Under_Avg1 = 0.0;
double Under_Avg2 =
0.0; double
Under_Avg3 = 0.0;
double Under_Avg4 =
0.0; double
Under_Avg5 = 0.0;
double Under_Avg6 =
0.0; double
Medium_Avg1 = 0.0;
double Medium_Avg2 =
0.0; double
Medium_Avg3 = 0.0;
double Medium_Avg4 =
0.0; double
Medium_Avg5 = 0.0;
double Medium_Avg6 =
0.0; double Well_Avg1
= 0.0; double
Well_Avg2 = 0.0;
double Well_Avg3 =
0.0; double Well_Avg4
= 0.0; double
Well_Avg5 = 0.0;
double Well_Avg6 =
0.0; double
Highly_Avg1 = 0.0;
double Highly_Avg2 =
0.0; double
Highly_Avg3 = 0.0;
double Highly_Avg4 =
0.0; double
Highly_Avg5 = 0.0;
double Highly_Avg6 =
0.0;
// Об'єкти класів для виклику функції визначення середнього значення:
UnderDeveloped
Und;
MediumDeveloped
Med;
WellDeveloped
Well;
HighlyDeveloped
```

```

High;
// Виклик функції середнього значення:
Und.Avg_Vector(Vector_Under, Under_Avg1, Under_Avg2, Under_Avg3,
Under_Avg4, Under_Avg5, Under_Avg6);
Med.Avg_Vector(Vector_Medium, Medium_Avg1, Medium_Avg2, Medium_Avg3,
Medium_Avg4, Medium_Avg5, Medium_Avg6);
Well.Avg_Vector(Vector_Well, Well_Avg1, Well_Avg2, Well_Avg3, Well_Avg4,
Well_Avg5, Well_Avg6);
High.Avg_Vector(Vector_Highly, Highly_Avg1, Highly_Avg2, Highly_Avg3, Highly_Avg4,
Highly_Avg5, Highly_Avg6);
// Вектори зі збереженими середніми значеннями кількісних ознак для кожної
категорій: vector<double>Und_Avg{
Under_Avg1,Under_Avg2,Under_Avg3,Under_Avg4,Under_Avg5,Under_Avg6 };
vector<double>Med_Avg{
Medium_Avg1,Medium_Avg2,Medium_Avg3,Medium_Avg4,Medium_Avg5,Medium
_Avg6 }; vector<double>Well_Avg{
Well_Avg1,Well_Avg2,Well_Avg3,Well_Avg4,Well_Avg5,Well_Avg6 };
vector<double>High_Avg{
Highly_Avg1,Highly_Avg2,Highly_Avg3,Highly_Avg4,Highly_Avg5,Highly_Avg6 };
SetConsoleTextAttribute(ColorText, 7); // Зміна кольору тексту
консолі cout << "\nВведіть дані про країну..." << endl;
SetConsoleTextAttribute(ColorText, 3); // Зміна кольору тексту
консолі cout << "\n=== Класифікація країн за розвиненістю
===>" << "\n\n"; bool next = true;
char symbol = {};
do // Цикл do...while:
{
SetConsoleTextAttribute(ColorText, 6);
// Ввід значень ознак об'єкту:
cout << "Введіть назву країни
=> "; cin >> Name_Country;
cout << "Введіть ВВП
=> "; cin >> GDP;
cout << "Введіть індекс людського розвитку
=> "; cin >> Index_HD;
cout << "Введіть індекс рівня освіти
=> "; cin >> Level_Education;
cout << "Введіть індекс рівня доступу до медичної
допомоги => "; cin >> Level_Medicine;
cout << "Введіть індекс корупції
=> "; cin >> Index_Corruption;
cout << "Введіть бюджет країни
=> "; cin >> Budget;
cout <<
"\n=====
<< endl;

```

```

SetConsoleTextAttribute(ColorText, 15); // Зміна кольору тексту консолі
vector<Categories>Vector_Object; // Вектор для збереження введених значень ознак
об'єкту
// Додання введених значень в вектор об'єкту:
Vector_Object.push_back(Categories(GDP, Index_HD, Level_Education,
Level_Medicine, Index_Corruption, Budget));
Categories Obj_1; // Об'єкт базового класу
SetConsoleTextAttribute(ColorText, 14); // Зміна кольору тексту
консолі
Obj_1.Recognition(Und_Avg, Med_Avg, Well_Avg, High_Avg, Vector_Object, x1, x2, x3,
x4); // Виклик функції евклідової відстані
double F1_Object
= 0.0; double
F2_Object = 0.0;
double F3_Object
= 0.0; double
F4_Object = 0.0;
Obj_1.PotencialFunction(x1, x2, x3, x4, F1_Object, F2_Object, F3_Object, F4_Object);
if ((F1_Object > F2_Object) && (F1_Object > F2_Object) && (F1_Object > F4_Object))
// Порівняння загальних відстаней від об'єкта до категорії:
{
fstream Und_File; // Створення файлу для запису та читання
Und_File.open("Under_Countries.txt", ios_base::out | ios_base::app); // Відкриття
файлу для запису та додавання нових даних
if (!Und_File.is_open()) // Перевірка на успішність відкриття файлу
{
cout << "При відкритті файлу виникла помилка!" << endl;
}
else
{
// Запис вказаних даних країни до файлу
категорії: for (int i = 0; i <
Vector_Object.size(); i++)
{
Class_Und = UnderDeveloped(Name_Country, Vector_Object[i].GDP,
Vector_Object[i].Index_HD, Vector_Object[i].Level_Education,
Vector_Object[i].Level_Medicine, Vector_Object[i].Index_Corruption,
Vector_Object[i].Budget);
Und_File << fixed <<
"===== " <<
"\n"
<< "Назва країни: " << Class_Und.Name_Country << "\n"
<< "Валовий внутрішній продукт(ВВП): " << Class_Und.GDP << "$" << "\n"
<< "Індекс людського розвитку(ІЛР): " << Class_Und.Index_HD << "\n"
<< "Індекс рівня освіти: " << Class_Und.Level_Education << "\n"
<< "Індекс рівня доступу до медичної допомоги: " << Class_Und.Level_Medicine

```

```

<< "\n"
<< "Індекс корупції: " << Class_Und.Index_Corruption << "\n"
<< "Бюджет: " << Class_Und.Budget << "$" << "\n"
<< "======" <<
"\n\n";
}
}
Und_File.close(); // Закриття файлу
cout << "Дана країна відноситься до слаборозвиненої категорії." << endl;
}
else if ((F2_Object > F1_Object) && (F2_Object > F3_Object) && (F2_Object >
F4_Object)) // Порівняння значень потенційної функції:
{
fstream Med_File; // Створення файлу для запису та читання
Med_File.open("Medium_Countries.txt", ios_base::out | ios_base::app); // Відкриття
файлу для запису та додавання нових даних
if (!Med_File.is_open()) // Перевірка на успішність відкриття файлу
{
cout << "При відкритті файлу виникла помилка!" << endl;
}
else
{
// Запис вказаних даних країни до файлу
категорії: for (int i = 0; i <
Vector_Object.size(); i++)
{
Class_Med = MediumDeveloped(Name_Country, Vector_Object[i].GDP,
Vector_Object[i].Index_HD, Vector_Object[i].Level_Education,
Vector_Object[i].Level_Medicine, Vector_Object[i].Index_Corruption,
Vector_Object[i].Budget);
Med_File << fixed <<
"======" <<
"\n"
<< "Назва країни: " << Class_Med.Name_Country << "\n"
<< "Валовий внутрішній продукт(ВВП): " << Class_Med.GDP << "$" << "\n"
<< "Індекс людського розвитку(ІЛР): " << Class_Med.Index_HD << "\n"
<< "Індекс рівня освіти: " << Class_Med.Level_Education << "\n"
<< "Індекс рівня доступу до медичної допомоги: " << Class_Med.Level_Medicine
<< "\n"
<< "Індекс корупції: " << Class_Med.Index_Corruption << "\n"
<< "Бюджет: " << Class_Med.Budget << "$" << "\n"
<< "======" <<
"\n\n";
}
}
Med_File.close(); // Закриття файлу

```

```

cout << "Дана країна відноситься до середньорозвиненої категорії." << endl;
}
else if ((F3_Object > F1_Object) && (F3_Object > F2_Object) && (F3_Object >
F4_Object)) // Порівняння значень потенційної функції:
{
fstream Well_File; // Створення файлу для запису та читання
Well_File.open("Well_Countries.txt",ios_base::out | ios_base::app); //Відкриття
файлу для запису та додавання нових даних
Well_File.close(); // Закриття файлу
cout << "Дана країна відноситься до добрерозвиненої категорії." << endl;
}
else if ((F4_Object > F1_Object) && (F4_Object > F2_Object) && (F4_Object >
F3_Object)) // Порівняння значень потенційної функції:
{
fstream High_File; // Створення файлу для запису та читання
High_File.open("Highly_Countries.txt", ios_base::out | ios_base::app); // Відкриття
файлу для запису та додавання нових даних
if (!High_File.is_open()) // Перевірка на успішність відкриття файлу
{
cout << "При відкритті файлу виникла помилка!" << endl;
}
else
{
// Запис вказаних даних країни до файлу категорії:
for (int i = 0; i < Vector_Object.size(); i++)
{
Class_High = HighlyDeveloped(Name_Country, Vector_Object[i].GDP,
Vector_Object[i].Index_HD,
Vector_Object[i].Level_Education, Vector_Object[i].Level_Medicine,
Vector_Object[i].Index_Corruption,
Vector_Object[i].Bud
get); High_File <<
fixed <<
"===== " << "\n"
<< "Назва країни: " << Class_High.Name_Country << "\n"
<< "Валовий внутрішній продукт(ВВП): " << Class_High.GDP << "$" << "\n"
<< "Індекс людського розвитку(ІЛР): " << Class_High.Index_HD << "\n"
<< "Індекс рівня освіти: " << Class_High.Level_Education << "\n"
<< "Індекс рівня доступу до медичної допомоги: " << Class_High.Level_Medicine
<< "\n"
<< "Індекс корупції: " << Class_High.Index_Corruption << "\n"
<< "Бюджет: " << Class_High.Budget << "$" << "\n"
<< "===== " <<
"\n\n";
}
}
}

```

```
High_File.close(); // Закриття файлу
cout << "Дана країна відноситься до сильнорозвиненої категорії." << endl;
}
SetConsoleTextAttribute(ColorText, 15); // Зміна кольору тексту консолі
// Пропозиція про ввід даних наступної країни:
cout << "Чи бажаєте ви вводити дані наступної країни? [Y(Так)/N(Ні)]"
<< endl; cin >> symbol;
if (symbol == 'Y')
{
next = true;
}
else if (symbol == 'N')
{
next = false;
cout << "Зупинка програми..." << endl;
Sleep(3000);
cout << "Виконання програми зупинено!" << endl;
}
else
{
next = false;
cout << "Такої команди не існує!" << endl;
}
} while (next == true);
}
```

## Демонстрація роботи програми:

```
Введіть дані про країну...

=== Класифікація країн за розвиненістю ===

Введіть назву країни => Пакистан
Введіть ВВП => 1500
Введіть індекс людського розвитку => 0.5
Введіть індекс рівня освіти => 0.6
Введіть індекс рівня доступу до медичної допомоги => 7
Введіть індекс корупції => 27
Введіть бюджет країни => 40000000000

=====

17366666666.666683
15666666666.666748
56500000000.002808
117575000000.001709

0.00000000000000003316
0.0000000000000000041
0.0000000000000000003
0.0000000000000000001

Дана країна відноситься до слаборозвиненої категорії.
Чи бажаєте ви вводити дані наступної країни? [Y(Так)/N(Ні)]
Y
Введіть назву країни => Франція
Введіть ВВП => 66000
Введіть індекс людського розвитку => 0.8
Введіть індекс рівня освіти => 0.87
Введіть індекс рівня доступу до медичної допомоги => 9.1
Введіть індекс корупції => 68
Введіть бюджет країни => 53100000000

=====

50836666666.6708984375000000000
33433333333.33868408203125000000
74000000000.00044250488281250000
68475000000.0000000000000000000

0.0000000000000000004
0.0000000000000000009
0.0000000000000000183
0.0000000000000000002

Дана країна відноситься до добрерозвиненої категорії.
```

Under\_Countries – Блокнот

Файл Правка Формат Вид Справка

```
=====
Назва країни: Пакистан
Валовий внутрішній продукт(ВВП): 1500.000000$
Індекс людського розвитку(ІЛР): 0.500000
Індекс рівня освіти: 0.600000
Індекс рівня доступу до медичної допомоги: 7.000000
Індекс корупції: 27.000000
Бюджет: 40000000000.000000$
=====
```

Дана програма класифікує нововведенні образи країн до визначених еталонних образів за методом потенційних функцій. Для даної програми використовуються ті ж дані, що були представлені при реалізації попереднього методу. Спочатку запускається програма, після чого користувачу потрібно ввести необхідні дані ознак для образу країни. Далі програма визначить середні значення для ознак всіх еталонних образів. Наступним буде визначення евклідової відстані на основі вказаних та визначених еталонних середніх значень. Після отриманих всіх необхідних даних, програма виконає потенційну функцію для кожного вхідного образу, підставивши туди його значення евклідової відстані. Результатом програми буде вивід значень потенційної функції та віднесення образу до відповідного вхідного образу. Образ буде віднесений до того еталону, де значення потенційної функції найбільше. На демонстрації можна побачити що країну Пакистан віднесло до слаборозвиненої країни, бо вихідне значення потенційної функції найбільше у слаборозвиненої категорії. Зокрема, при віднесенні країни до відповідної категорії тут також відбувається збереження даних країни в визначений текстовий файл.

### **Висновок:**

В ході лабораторної роботи було опрацьовано теоретичний матеріал по методу, що відноситься до категорії наближених сусідів – метод потенційних функцій. Було розглянуто приклад з використанням даного методу та реалізовано в середовищі Visual Studio програму за методом потенційних функцій з використанням даних та деяких функцій з попередньої лабораторної роботи.