

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ

до виконання розрахункового завдання
з дисципліни «Програмування»
для студентів першого курсу всіх форм навчання
спеціальностей «Комп'ютерна інженерія» та «Кібербезпека»

Затверджено редакційно-видавничою
радою університету,
протокол № 2 від 24.05.2018 р.

Харків
НТУ «ХПІ»
2018

Методичні вказівки до виконання розрахункового завдання з дисципліни «Програмування» для студентів першого курсу всіх форм навчання спеціальностей «Комп'ютерна інженерія» та «Кібербезпека» /уклад.: Давидов В. В., Далека В. Д., Молчанов Г. І. – Харків: НТУ «ХПІ», 2018. – 30 с.

Укладачі: В. В. Давидов
В. Д. Далека
Г. І. Молчанов

Рецензент І. С. Зиков

Кафедра обчислювальної техніки та програмування

ЗМІСТ

Перелік скорочень	4
Вступ	5
1 Вимоги до розрахункового завдання з дисципліни «Програмування»	6
2 Програмні документи	10
2.1 Види програмних документів	10
2.2 Шифри документів	10
2.3 Опис програмних документів	11
2.3.1 Загальні вимоги	11
2.3.2 Пояснювальна записка	12
2.3.3 Оформлення схем програмної документації	17
3 Основні вимоги до оформлення текстових документів	19
3.1 Загальні вказівки	19
3.2 Правила оформлення текстових документів	19
3.2.1 Нумерація сторінок	19
3.2.2 Побудова тексту	19
3.2.3 Виклад тексту	20
3.2.4 Оформлення ілюстрацій	21
3.2.5 Побудова таблиць	21
3.2.6 Написання формул	22
3.2.7 Оформлення приміток	22
3.2.8 Посилання на нормативну літературу	23
3.2.9 Оформлення списку джерел інформації	23
3.2.10 Оформлення додатків	24
Список джерел інформації	25
Додаток А. Приклад аркуша затвердження пояснювальної записки	26
Додаток Б. Приклад титульного аркуша пояснювальної записки	27
Додаток В. Приклад змісту пояснювальної записки	27
Додаток Г. Приклад тексту програми	29
Додаток Д. Приклад структури програми та схеми алгоритму	30

ПЕРЕЛІК СКОРОЧЕНЬ

ГОСТ	– Державний стандарт
ЄСПД	– Єдина система програмної документації
ДСТУ	– Державний стандарт України
КІТ	– комп'ютерні та інформаційні технології
ОТП	– обчислювальна техніка та програмування
РЗ	– розрахункове завдання
СТВУЗ	– Стандарт вищого навчального закладу
ХПІ	– Харківський політехнічний інститут
ISO	– (International Organization for Standardization) – міжнародна організація по стандартизації,

ВСТУП

Розрахункове завдання (РЗ) – це самостійна робота студента, що виконується під керівництвом викладача, вона передбачає розв'язання конкретного практичного навчального завдання з використанням відомого, а також(або) самостійно вивченого теоретичного матеріалу. За час виконання роботи студент має показати набутий рівень знань з однієї або декількох дисциплін, вміння працювати з науково-технічною літературою, самостійно вирішувати поставлене індивідуальне завдання, забезпечуючи ефективність його реалізації, розробляти відповідні документи.

Розрахункове завдання з дисципліни «Програмування» є обов'язковим видом самостійної роботи студента, що передбачене навчальним планом для бакалаврів, які навчаються за спеціальністю «Комп'ютерна інженерія». Для майбутніх програмістів розрахункове завдання – це один з видів тренінгу, тематика якого повинна відповідати сучасному стану та перспективам розвитку комп'ютерних технологій. Саме тому керівник має вирішувати такі питання: які об'єкти запропонувати студенту для тренінгу, використання яких засобів вимагати при вирішенні поставлених завдань; також він повинен закласти типові вимоги до виконання інших самостійних робіт.

Стосовно першого питання. Відомо, що з початку розвитку обчислювальної техніки утворилося два основних напрями її використання: для виконання чисельних розрахунків та в автоматичних або автоматизованих інформаційних системах обробки даних. Останні набули широкого впровадження в життєдіяльність людини. Тому доцільним буде запропонувати до розгляду розробку програм для інформаційних систем, що вимагатиме від студента знання алгоритмів обробки як чисельних, так і нечисельних даних, вирішувати питання розробки інтерфейсу користувача.

Стосовно технології програмування. Натепер беззаперечним лідером є об'єктно-орієнтоване програмування, яке дозволяє розкласти проблему на пов'язані між собою завдання. Кожне з них стає самостійним об'єктом, що містить свої власні коди та дані, які стосуються цього об'єкта. У цьому випадку вихідне завдання в цілому спрощується, і програміст одержує можливість розробляти більші за обсягом програми. З погляду програмування подібний підхід значно спрощує розробку і налагодження програм. Саме до такого підходу необхідно заохочувати студента з самого початку його навчання.

1 ВИМОГИ ДО РОЗРАХУНКОВОГО ЗАВДАННЯ З ДИСЦИПЛІНИ «ПРОГРАМУВАННЯ»

Тема роботи. Розробка інформаційно-довідкової системи.

Мета роботи. Закріпити отримані знання з дисципліни «Програмування» шляхом виконання типового комплексного завдання.

Завдання до роботи. Кожний студент отримує індивідуальне завдання. Варіант завдання обирається за номером прізвища студента у журналі групи.

При виконанні завдання з розробки інформаційно-довідкової системи необхідно виконати наступне:

- 1) з табл. 1, відповідно до варіанта завдання, обрати прикладну галузь;
- 2) дослідити літературу стосовно прикладної галузі. За результатами аналізу літератури оформити перший, аналітичний розділ пояснювальної записки обсягом 2–3 сторінки;
- 3) для прикладної галузі розробити розгалужену ієрархію класів, яка складається з не менш ніж трьох класів, один з яких є «батьком» для інших (класів-спадкоємців). Класи повинні мати перевантажені оператори введення-виведення даних та порівняння;
- 4) розробити клас-контролер, що буде включати колекцію розроблених класів, та наступні методи роботи з цією колекцією:
 - а) читання даних з файлу та їх запис у контейнер;
 - б) запис даних з контейнера у файл;
 - в) сортування елементів у контейнері за вказаними критеріями: поле та напрям сортування, які задаються користувачем з клавіатури;
 - г) пошук елементів за вказаним критерієм (див. «Завдання для обходу колекції» в табл. 1);
- 5) розробити клас, який має відображати діалогове меню для демонстрації реалізованих функцій класу контролера;
- 6) оформити схеми алгоритмів функцій класів контролера та діалогового меню;
- 7) оформити документацію: пояснювальну записку (див. розділ 2 даних методичних вказівок).

Увага. Текст програми та результати роботи програми мають бути подані в додатках.

Вимоги:

- усі класи повинні мати конструктори та деструктори;
- якщо функція не змінює поля класу, вона має бути декларована як константна;
- рядки повинні бути типу *string*;
- при перевантаженні функції треба використовувати ключове слово *override*;
- програмний код усіх класів має бути 100 % *doxygen*-документований;
- у звіті текст програми слід оформляти стилем *Courier new* 8 пт, інтервал – одиничний; довжина рядка не повинна перевищувати 80 символів.

Додаткові вимоги на оцінку «добре»:

- виконання основного завдання та додаткових наступних вимог:
- додати обробку помилок; при цьому функція, що генерує виключення, при її декларуванні повинна мати ключове слово *throw*;
- виконати перевірку вхідних даних за допомогою регулярних виразів.

Додаткові вимоги на оцінку «відмінно»:

- виконати завдання відповідно до вимог на оцінку «добре» та додаткові наступні вимоги:
- критерій для пошуку та сортування задавати у вигляді функтора;
- розробити клас-тестер, основною метою якого буде перевірка коректності роботи класу-контролера.

Таблиця 1 – Індивідуальні завдання

№ з/п	Прикладна галузь	Базовий клас	Завдання для обходу колекції
1	Учнівська молодь	Студент	Обчислити витрати держави на видачу стипендії групі за один навчальний рік
2	Навчальний корпус	Аудиторія	Обчислити середню площу аудиторій в навчальному корпусі
3	Кафедра	Співробітник	Обчислити середню заробітну плату співробітників
4	Директорія ПК	Файл	Знайти кількість файлів, які мають атрибут «Прихований» і є системними

Продовження табл. 1

№ з/п	Прикладна галузь	Базовий клас	Завдання для обходу колекції
5	Залізничне депо	Поїзд	Визначити, у скільки разів максимальна швидкість електропоїздів відрізняється від максимальної швидкості усіх поїздів залізничного депо
6	Самостійні роботи студентів	Розрахунково-графічне завдання	Визначити кількість РГЗ, що виконує студент за весь період навчання в інституті відповідно до навчального плану
7	Компоненти комп'ютерів	Запам'ятовуючий пристрій	Знайти запам'ятовуючі пристрої, які мають найменшу ціну за 1 Гб
8	Світ	Країна	Визначити, яка країна має найменшу щільність населення
9	Навчання	Випускна кваліфікаційна робота	Визначити % магістерських робіт у порівнянні з бакалаврськими роботами
10	Роботи студентів	Самостійна робота	Визначити, яку кількість домашніх завдань виконує студент за семестр
11	Навчальні заклади	Школа	Визначити кількість навчальних закладів з кількістю працівників більше за середній показник
12	Література	Підручник	Обчислити середній обсяг (у сторінках) електронних ресурсів
13	Програмне забезпечення	Програма, що виконується	Отримати список програм, розмір яких більше заданого розміру (наприклад 100 Кбайт). Зі списку виключити «трояни»
14	Компоненти програм	Бібліотека, що підключається (модуль)	Визначити, у скільки разів обсяг бібліотек, які динамічно підключаються, менше, ніж загальний обсяг бібліотек
15	Електронні пристрої	Робот	Визначити кількість робітв зазначеного виробника з найкращим WiFi модулем
16	Комп'ютерна техніка	Комп'ютер (PC)	Знайти ноутбук з максимальною діагоналлю

Продовження табл. 1

№ з/п	Прикладна галузь	Базовий клас	Завдання для обходу колекції
17	Мобільні пристрої	Телефон (наприклад Nokia 1100)	Визначити телефон з найменшою щільністю пікселів
18	Монітори	LED монітор	Визначити кількість моніторів у колекції з сенсорними датчиками
19	Академічна група	Студент	Визначити, хто із студентів за результатами сесії має заборгованості і який % становлять їх заборгованості з програмування
20	Змагання	Олімпіада	Визначити, на якій олімпіаді була найбільша кількість учасників
21	Соціум	Людина	Визначити, який % студентів технічних вузів займаються за дистанційною формою навчання
22	Література	Книга	Визначити кількість книг з програмування Шилдта і скільки в них питань для самоперевірки
23	Транспорт	Міський транспорт	Визначити кількість пасажирів, що перевозить міський електричний транспорт
24	Двигун внутрішнього згоряння	Двигун	Обчислити, у скільки разів обсяг споживаного палива роторних двигунів менше обсягу споживаного палива поршневих двигунів
25	Морський флот	Корабель	Визначити співвідношення пасажирських та вантажних кораблів
26	Навчання	Навчальні дисципліни	Визначити, який відсоток у навчальному плані складають фахові дисципліни
27	Об'єктно-орієнтоване програмування	Клас	Визначити, який клас має найбільшу кількість приватних методів
28	Двері	Дверна коробка	Знайти найдорожчу профільну коробку з алюмінію

2 ПРОГРАМНІ ДОКУМЕНТИ

2.1 Види програмних документів

ДЕРЖСТАНДАРТ ГОСТ 19.101-77 визначає види програмних документів. Програмними документами є такі документи, що містять відомості, необхідні для розробки, супроводу та експлуатації програм [4]. Для можливості посилання на програмні документи у ГОСТах їм привласнені відповідні коди, що для експлуатаційних документів наведені в табл. 2.

Таблиця 2 – Види експлуатаційних програмних документів

Код виду документа	Вид документа	Вміст документа
Немає коду	Специфікація	Склад програми та перелік документації на неї
12	Текст програми	Програмний код з коментарями
13	Опис програми	Відомості про логічну структуру та функціонування програми
34	Керівництво оператора	Необхідні відомості для забезпечення роботи оператора при виконанні програми
51	Програма й методика випробувань	Вимоги, що підлягають перевірці при випробуваннях програми, а також порядок і методи їхнього контролю
81	Пояснювальна записка	Схеми та опис алгоритмів, опис функціонування програми, обґрунтування прийнятих рішень

2.2 Шифри документів

Для ідентифікації документа ЄСПД ГОСТ 19.103-77 визначає структуру його шифру [5]. На рис. 1 подано структуру шифру для специфікації, а на рис. 2 – для інших програмних документів.

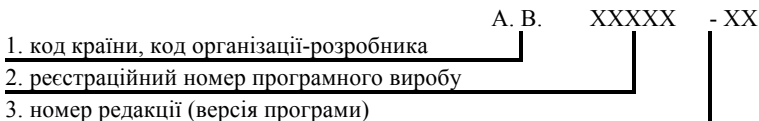


Рисунок 1 – Структура шифру специфікації

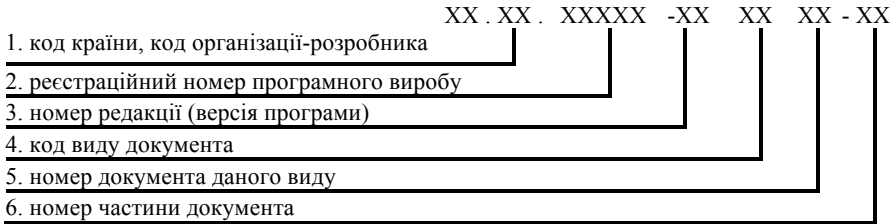


Рисунок 2 – Структура шифру інших програмних документів (крім специфікації)

Для студентських робіт, присвячених розробленню програм та програмних документів, на кафедрі ОТП вирішено, що у загальній частині шифру замість коду країни та організації (перша кодова група) необхідно вказувати назву факультету та індекс навчальної групи студента, а замість реєстраційного номера програмного виробу (друга кодова група) – п'ятизначний номер залікової книжки. Таким чином, наприклад, шифр документа «Пояснювальна записка», що розроблена студентом з групи КІТ25А із заліковою книжкою з номером 20176, буде такий: КІТ.25А.20176-01 81 01-1.

Відповідно до вимог ГОСТ 19.106-78 шифр документа проставляється на кожному листі текстової частини документа у верхньому його колонтитулі посередині.

2.3 Опис програмних документів

2.3.1 Загальні вимоги

Згідно з вимогами ДЕРЖСТАНДАРТ ГОСТ 19.101-77 [7] програмні документи включають:

- вступну частину;
- основну частину;
- додатки.

Вступна частина програмного документа містить:

- аркуш затвердження (див. Додаток А). Це завжди перший аркуш документа та він не нумерується;
- титульний аркуш (див. Додаток Б), це другий аркуш документа; його номер – 1, але номер на цьому аркуші не ставиться;
- анотацію – стисла інформація (до 5 рядків) про зміст документа;

пишеться українською, російською та англійською мовами на одному аркуші. Номер на цьому аркуші не ставиться;

- зміст (мають усі документи крім специфікації);
- перелік умовних позначень, символів, одиниць, скорочень і термінів (цей аркуш може бути відсутнім у документі).

Зверніть увагу. Нумеруються всі аркуші документа, починаючи з «Титульного аркуша». Проставляються номери тільки з другого аркуша «Змісту». Номер аркуша записується у верхньому його колонтитулі справа.

На усіх аркушах (починаючи з анотації) у верхньому колонтитулі посередині проставляється шифр документа.

В розрахунковому завданні необхідно розробити тільки один програмний документ – «Пояснювальну записку».

2.3.2 Пояснювальна записка

Код – 81. Оформлення документа «Пояснювальна записка» регламентуються ЄСПД ГОСТ 19.404–79.

Документ повинен містити вступ і такі розділи:

- призначення та галузь застосування;
- постановку завдання до розробки;
- опис вхідних та вихідних даних;
- опис складу технічних та програмних засобів;
- список джерел інформації.

У «Вступі» можна дати, наприклад, таку інформацію:

Інформаційно-довідкові системи – це такі програмні оболонки, що призначені для керування базами даних і використовуються, наприклад, для ведення статистики, складання каталогів тощо. Так, інформаційні системи підтримують ведення...

Необхідно навести інформацію про прикладну галузь, що розробляється.

У розділі **«Призначення та галузь застосування»** вказується призначення програми, дається характеристика області застосування.

Наприклад. *Призначення розробки – оперування даними про ...; дозволяє виконувати пошук за зразком, сортування, додавання даних, їх видалення тощо. Може використовуватися в ...*

У розділі **«Постановка завдання до розробки»** розміщують наступне:

- аналітичний огляд інформації з порушеного питання (про об'єкти прикладної галузі);

- завдання на розробку програми;
- опис застосовуваних математичних методів, обґрунтовується вибір сучасної технології програмування, вказується середовище програмування;
- відповідно до обраного шляху проектування словесно описується обрана структура даних, розроблена структура програми та алгоритм її функціонування. Обґрунтовується коректність алгоритму або окремих його фрагментів;

- аналогічно описуються розроблювані функції.

Аналітичний огляд інформації можна почати так:

В основі функціонування інформаційно-довідкових систем лежить обробка інформації. Режими її обробки можуть бути такими: пакетний, діалоговий, реального часу [2].

Діалоговий режим забезпечує безпосередню взаємодію користувача з системою. Ініціатором діалогу може бути як користувач, так і ЕОМ. В останньому випадку на кожному кроці користувачу повідомляється, що треба робити. ...В роботі буде реалізовано....

Дані, що обробляються, в оперативній пам'яті можуть зберігатися у вигляді масиву або лінійного (одно- або двонаправленого) списку [1]

До переваг масиву можна віднести ..., до недоліків –

До переваг списку можна віднести ..., до недоліків –

Виходячи з ... в розроблюваній програмі для подання даних буде реалізовано

В завданні на розробку, виходячи з аналітичного огляду, конкретизується перелік завдань щодо розробки.

При виборі мови і середовища програмування можна зазначити таке:

Об'єктно-орієнтоване програмування має такі переваги: Мова програмування ... забезпечує Тому для розробки програми обрано мову програмування ..., а середовище програмування –

При розробленні структур даних слід виходити з заданого об'єкта та його властивостей. Тут можна запропонувати таке:

Абстрактний клас Animal – базовий (рис. 3). Його наслідують класи Cow (рис.4, а) та Hen (рис.4, б). . .

На рис. 5 показано внутрішню структуру та відносини розроблених класів у вигляді UML-діаграми.

Приватні дані	
float	height висота тварини, см Детальніше...
float	mass маса тварини, кг Детальніше...
string	name ім'я тварини Детальніше...

Рисунок 3 – Поля класу *Animal*

Приватні дані	
bool	horns Наявність рогів. Детальніше...
int	amountOfMilkPerDay Кількість молока, яке корова дає за день.

а

Приватні дані	
int	amountOfEggsPerDay Кількість знесених яєць в день.

б

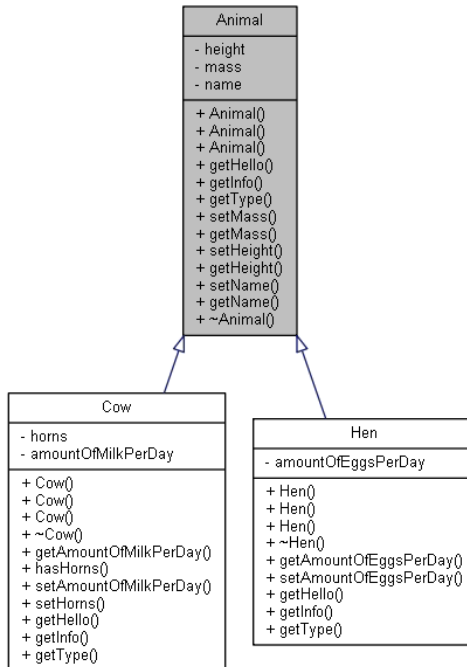
Рисунок 4 – Додаткові поля класу *Cow* (а) та *Hen* (б)

Рисунок 5 – Схема ієрархії розроблених класів

Дані про всіх тварин будуть заноситися до списку. Для цього розроблено клас *MuFarm* з такими полями, як на рис. 6.

Приватні дані

```
vector< Animal * > farmAnimals
Масив тварин на фермі.
```

Рисунок 6 – Поля класу *MyFarm*

Розроблені методи класу *MyFar* подані на рис. 7.

Загальнодоступні елементи

MyFarm (void)	Конструктор ферми. Детальніше...
~MyFarm (void)	Деструктор ферми. Детальніше...
void addAnimal (Animal *const animal)	Додавання тваринного на ферму. Детальніше...
Animal * operator[] (int index) const	Перевантажений оператор індексування. Детальніше...
int getAnimalCount () const	Визначення кількості тварин на фермі. Детальніше...
void writeFarmToFile (string filename) const	Збереження інформації про ферму в файл. Детальніше...
void restoreFromFile (string filename) const	Відновлення інформації про ферму з файлу. Детальніше...
template<class SortPredicate >	
void sortFarmAnimals (SortPredicate &sortPredicate)	Сортування тварин ферми по заданому критерію. Детальніше...
template<class Criterion >	
vector< Animal * > getAnimalsByCriterion (Criterion &critierion) const	Пошук тварин ферми по заданому критерію. Детальніше...

Рисунок 7 – Методи класу *MyFarm*

При описі структури програми можна вказати, наприклад, таке:

Програма складається з таких модулів:

- *інтерфейсного, який містить функцію main і функції реалізації діалогу;*
- *модуля, що містить опис класу Animal;*
- *модуля, що містить опис класу Cow;*
- *модуля, що містить опис класу Hen;*
- *модуля, що містить опис класу ферми - MyFarm;*
- *модуля, що містить модульні тести;*
-

На рис. 8 подано структуру проекту розробленого програмного

продукту.

Повний список файлів.












 Animal.cpp	Реалізація абстрактного класу, що описує тварину
 Animal.h	Базовий клас - тварина
 ConsoleUtil.cpp	Реалізація класу для відображення діалогового меню
 ConsoleUtil.h	Утиліта для роботи з консоллю
 Cow.cpp	Реалізація класу, що описує корову
 Cow.h	Опис класу, що описує корову
 Hen.cpp	Реалізація класу, що описує курку
 Hen.h	Опис класу, що описує курку
 main.cpp	Клас, що демонструє роботу з реалізованими класами на прикладі діалогового меню з користувачем
 MyFarm.cpp	Реалізація класу, що описує ферму, на якій живуть тварини - корови і курки
 MyFarm.h	Описание класса, описывающего ферму, на которой живут животные - коровы и курицы

Рисунок 8 – Структура проекту

При описі методів класів та функцій слід висвітлити таке:

- призначення;
- посилання на схему алгоритму функції (якщо є);
- порядок роботи.

При опису головної функції *main* можна зазначити, наприклад, таке:

Функція main координує роботу всіх функцій програми.

Схема алгоритму функції подана на рис.

Порядок роботи: В циклі видається меню, вводиться команда оператора (блок ...), яка розпізнається у блоці ... і, залежно від значення команди керування передається відповідній функції. Програма припиняє роботу після введення команди «...».

В розділі **«Опис вхідних та вихідних даних»** подається обґрунтований вибір методу організації вхідних і вихідних даних, перераховується необхідний набір і джерело вхідних даних, спосіб їх одержання розроблювальною програмою.

Наприклад: *Вхідні дані можуть вводитися оператором з клавіатури в такій послідовності ..., або з типізованого файлу.*

Для вихідних даних аргументується вибір носія для їх зберігання та спосіб відображення для користувача.

Наприклад: *Вихідні дані зберігаються в типізованому файлі. Типізований файл обрано тому, що у порівнянні з текстовим файлом він має такі переваги До недоліків слід віднести таке*

В розділі **«Список джерел інформації»** наводять перелік науково-технічних публікацій, нормативно-технічних документів і інших науково-

технічних матеріалів. Докладна інформація про надання переліку джерел подана в п. 3.2.9 даних вказівок.

Увага. В разі необхідності ДЕРЖСТАНДАРТ ГОСТ 19.101-77 дозволяє об'єднувати окремі розділи документа і додавати інші.

Приклад аркуша затвердження пояснювальної записки подано у додатку А, титульного аркуша – у додатку Б, змісту – у додатку Г, структури програми – у додатку Д.

2.3.3 Оформлення схем програмної документації

Алгоритм – це будь-яка обчислювальна процедура, що описує несуперечливий хід виконання завдання від входу до отримання результату. Алгоритм відображає послідовність обчислювальних кроків, які дозволяють перетворити вхідні дані у вихідні.

При поданні алгоритму у схематичному вигляді використовуються спеціальні графічні символи, скорочений список яких подано в табл. 3. У стандарті ДСТУ ISO 5807:2016 (аналог – ЄСПД ГОСТ 19.701-90) визначені графічні символи, що призначені для використання в документації з оброблення даних, і правила їх використання [13].

Символи в схемі повинні бути розташовані рівномірно. Варто дотримуватися розумної довжини з'єднань і мінімального числа довгих ліній. Форми символів, що установлені стандартом не повинні змінюватися. Символи повинні бути, по можливості, одного розміру.

У схемах може використовуватися ідентифікатор символів, який використовується для посилань на них у документах. Ідентифікатор символу повинний розташовуватися ліворуч над символом.

Потоки даних у схемах показуються лініями. Напрямок потоку зліва направо і зверху вниз вважається стандартним. У випадках, коли необхідно внести ясність у схему (наприклад, при з'єднаннях), на лініях використовуються стрілки. Якщо потік має напрямок, відмінний від стандартного, стрілки повинні вказувати цей напрямок.

У схемах варто уникати перетинання ліній. Дві або більше вхідних ліній можуть поєднуватися в одну вихідну лінію, в цьому випадку місце об'єднання повинне бути зміщено.

Приклад:



Лінії в схемах повинні підходити до символу або ліворуч, або зверху, а виходити або праворуч, або знизу. При необхідності лінії в схемах варто розривати для запобігання зайвим перетинанням або занадто довгим лініям, а

також у випадках, коли схема розміщується на декількох сторінках.

Таблиця 3 – Список (скорочений) графічних символів для побудови схем

№ п/п	Найменування	Позначення і розмір	Функція
1	Пуск-зупинка	 $a=0,5b$	Початок, кінець, переривання процесу обробки даних або виконання програми
4	Процес	 $b=1,5a$	Виконання операцій, в результаті яких змінюється значення, форма уявлення або розташування даних a
5	Зумовлений процес		Використання раніше створених і окремо описаних алгоритмів або програм
6	Скорочена деталізація		Частина програми, що деталізується. Над горизонтальною лінією розміщується назва (Name)
7	Вирішення, умова		Вибір напрямку виконання алгоритму або програми залежно від певних змінних умов (+ да, - ні)
8	Модифікація		Виконання операцій (операції циклу), які змінюють команди або групи команд, що змінюють програму
9	Введення-виведення		Перетворення даних у форму, придатну для обробки (введення) або відображення результатів обробки (виведення)
12	З'єднувач		З'єднувач оформляє розрив ліній, який буває внутрішнім і зовнішнім
11	Коментар		Зв'язок між елементом схеми і поясненням
12	Лінія		Зображення потоків даних

Якщо фрагменти схем описуються в текстовому документі, наприклад, у пояснювальній записці, в описі програми з метою підкреслити важливі або оригінальні логічні особливості розроблювального виробу, то ці фрагменти вставляють у текст у вигляді рисунків з підписуночним підписом.

3 ОСНОВНІ ВИМОГИ ДО ОФОРМЛЕННЯ ТЕКСТОВИХ ДОКУМЕНТІВ

3.1 Загальні вказівки

Згідно з вимогами стандарту СТВУЗ-ХПП-3.01-2010 текстові документи оформляються на аркушах формату А4 (210*297 мм) [3]. Документи виконуються за допомогою комп'ютерної техніки на одній стороні аркуша білого паперу, а українською мовою.

Розмір полів: верхнє – 20 мм, лівє – 25 мм, нижнє – 15 мм, правє – 15 мм. Друк виконується шрифтом Times New Roman текстового редактора Word розміром 14 пунктів з полуторним міжрядковим інтервалом.

Рисунки (у тому числі схеми алгоритмів) виконуються машинним способом із використанням рекомендованого пакету Visio.

3.2 Правила оформлення текстових документів

3.2.1 Нумерація сторінок

Нумерація сторінок документа наскрізна (включаючи додатки). Номери сторінок проставляються арабськими цифрами у правому верхньому куті без крапки.

Починають проставлять номери сторінок з другого аркуша «Змісту».

3.2.2 Побудова тексту

Написання текстових документів потрібно виконувати за вимогою ДСТУ 3008 –95 та стандарту СТВУЗ-ХПП-3.01-2008 [1, 3].

Текстовий документ може містити суцільний текст з абзацами і текст, розбитий на графи (специфікації, таблиці, відомості і т.п.). Не допускаються будь-які прикраси тексту зміною шрифту, підкресленням слів, фраз або назв елементів рубрикації, застосуванням різнобарвних букв і т. д.

Структурні елементи: «ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ», «ВСТУП», «ВИСНОВОК», «СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ» не нумерують, а їхні назви служать заголовками структурних елементів.

Кожен розділ тексту повинен починатися з нової сторінки і мати порядкову нумерацію в межах усього документа арабськими цифрами без крапки, наприклад: 1, 2, 3 і т.д.

Нумерація підрозділу складається з номера розділу і порядкового номера підрозділу, розділених крапкою. Після номера підрозділу крапку не

ставлять, наприклад: 1.1, 1.2 і т.д.

Якщо в тексті є пункти, то в межах кожного підрозділу вони повинні мати порядкову нумерацію і їхній номер складається з номера розділу, номера підрозділу, пункту, розділених крапкою. Після номера пункту крапку не ставлять, наприклад: 1.1.1, 1.1.2 і т.д.

Підпункти тексту нумеруються окремо в межах кожного пункту. Їх номери складаються з номера розділу, порядкових номерів підрозділу, пункту і підпункту, розділених крапкою, наприклад: 2.1.1.1, 2.1.1.2, 2.1.1.3 і т.д. Після номера підпункту крапку не ставлять.

Кожний розділ, пункт і підпункт повинні мати стислий заголовок, що відповідає змісту. Заголовки розділів розташовують посередині рядка і записують прописними буквами, а підрозділів, пунктів і підпунктів починають з абзацного відступу і записують малими (крім першої великої) жирними літерами. Заголовки не підкреслюють і крапку наприкінці не ставлять. Перенос слів у заголовках не допускається. Якщо заголовок складається з двох чи більше частин, їх розділяють крапкою.

Відстань між заголовком і наступним текстом повинна бути такою, як між рядків у тексті, а між заголовком і попереднім текстом – 12 пунктів. Не допускається розміщати найменування розділу, підрозділу, пункту і підпункту в нижній частині сторінки, якщо після нього розташований тільки один рядок тексту (заборона «вісячих рядків»).

3.2.3 Виклад тексту

Виклад тексту повинен бути коротким і чітким. Текст викладають згідно зі стандартом і технічними умовами, прийнятими в науково-технічній літературі, тобто виклад тексту ведеться від третьої особи в дійсному способі, вживаючи дієслова невизначеної форми. Наприклад, замість «приймаю» або «приймаємо» треба писати «приймаються», «дані розрахунку наводяться...», «у розділі розглядається...».

Скорочення слів у тексті і підписах під ілюстраціями, як правило, не допускається. ГОСТ 2.316-2008 визначив виключення, що дозволяють скорочувати окремі слова і словосполучення тексту. Основні виключення:

- і т.п., і т.д., і т.ін., ін. – тільки наприкінці фрази;
- див., табл., рис., стор., вип., журн., л. (лист), п. (пункт), розд. (розділ), ст. (стаття) – при посиланнях і виносках у суцільному тексті.

Якщо в документі прийнята особлива система скорочення слів або

найменування, то в документі наводять перелік прийнятих скорочень.

Обороту «від...до» у тексті варто уникати, його можна замінити через межу 5–10 або крапки 40...60. Абстрактні числа до 9 пишуться словами, понад 9 – цифрами (три криві; 10 розділів і т.д.).

Числа з розмірністю пишуться тільки цифрами, наприклад, 3 км, 5 грн. і т.п. Дробові значення проставляються тільки цифрами – «1/4 ч».

Числові значення в тексті пишуться тільки від 0,1 до 1000. Для більших значень вводять кратні одиниці (мега – 10^6 ; гіга – 10^9 ; тера – 10^{12} ; пета – 10^{15} ; еска – 10^{18}), а для менших – дольові (санти – 10^{-2} ; мілі – 10^{-3} ; мікро – 10^{-6} ; нано – 10^{-9} ; піко – 10^{-12} ; фемто – 10^{-15} ; атто – 10^{-18}).

Наприклад, 110 терабайт, 0.15 А (0.15 ампер).

3.2.4 Оформлення ілюстрацій

Ілюстрації можуть бути розташовані по тексту документа відразу після першого посилання на них або на наступній сторінці. Крім того, вони можуть бути подані в додатках. Усі наведені в документі ілюстрації позначають словом «Рисунок» і нумерують арабськими цифрами в межах усього тексту (при їх невеликій кількості) або кожного розділу (при їх великій кількості). У випадку, коли ілюстрації нумерують у межах розділу, номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, розділених крапкою. Номер ілюстрації в тексті вказують так: Рисунок 1 або Рисунок 2.1 (перша ілюстрація другого розділу).

При посиланні в тексті на ілюстрацію треба вказати її повний номер, наприклад, рис. 1, рис. 1.5. Повторні посилання на раніше згадані ілюстрації дають зі скороченням слова «дивися», наприклад: див. рис. 6.2. У випадку наявності в документі однієї ілюстрації вона в обов'язковому порядку підлягає нумерації з обліком викладених вище положень.

Ілюстрації повинні мати тематичну назву, що починається з великої букви, без крапки наприкінці і розміщується під ілюстрацією. При необхідності ілюстрації доповнюють пояснювальними даними (через тире).

Наприклад: Рисунок 1.15 – Схема алгоритму програми

3.2.5 Побудова таблиць

У табличній формі подають інформацію, що не піддається відтворенню іншими способами (у вигляді схем, графіків та ін.).

Таблицю розташовують після першого посилання на неї або на наступній сторінці. У документі таблиці повинні відповідати вимогам ДСТУ

3008-95. При наявності в тексті декількох таблиць їх нумерують арабськими цифрами. Напис типу «Таблиця 2» або «Таблиця 1.2» (друга таблиця першого розділу) поміщають ліворуч над таблицею без лапок і крапки. На всі таблиці повинні бути посилання в тексті, наприклад, (табл. 3), при повторному посиланні – (див. табл. 3). Якщо в тексті одна таблиця, то їй привласнюють номер відповідно до викладених вище вимогам.

Таблиця в обов'язковому порядку повинна мати тематичний заголовок, який необхідно виконувати малими літерами (крім першої великої) і поміщати над таблицею разом зі словом Таблиця. Номер таблиці та її назва відокремлюються символом «тире».

Наприклад: Таблиця 2 – Призначення змінних

3.2.6 Написання формул

Формули набираються в редакторі формул Microsoft Equation 3.0 з такими налаштуваннями: основний текст 14 пунктів, великий індекс – 20 пунктів, дрібний індекс 7 пунктів, великий символ – 20 пунктів. Латинські символи – похилі, грецькі й кирилиця – прямі символи.

Формули, на які є посилання в тексті, повинні нумеруватися в межах розділу арабськими цифрами. Номер формули повинен складатися з номера розділу й порядкового номера формули, розділених крапкою. Номер формули варто брати в дужки й поміщати на правому полі на рівні нижнього рядка формули, до якої вона належить.

При посиланні в тексті на формулу необхідно вказувати її повний номер у дужках, наприклад: «У формулі (1.2)».

3.2.7 Оформлення приміток

У примітках до тексту і таблиць указують тільки довідкову інформацію, що її пояснює. Примітки розташовують безпосередньо після тексту, таблиці, ілюстрації, до яких вони мають відношення. На кресленнях примітки не дають.

Слово «Примітка» пишуть із великої букви з абзацного відступу, не підкреслюють, після даного слова ставлять крапку і з великої букви в тім же рядку дають текст примітки.

Якщо приміток декілька, то після слова «Примітки» ставлять двокрапку. Примітки нумерують арабськими цифрами з крапкою після номера. Кожну примітку записують з нового рядка з абзацу. Текст примітки починається з великої букви. Наприклад,

Примітки:

1 ...

2 ...

3.2.8 Посилання на нормативну літературу

Посилання на нормативну літературу (стандарти, правила, технічні умови, інструкції) і інші джерела (книги, статті, патенти) у тексті документа необхідно наводити у випадку, якщо в них містяться відомості, що підтверджують обґрунтованість прийнятих рішень, методик розрахунків, вибір формул, коефіцієнтів, нормативних величин. При посиланні потрібно вказувати у квадратних дужках порядковий номер джерела, наведеного наприкінці документа списку літератури, наприклад [1], [3,5,7], [3 – 6], [5,107 – 129 с.].

3.2.9 Оформлення списку джерел інформації

Перелік літературних джерел повинний бути поданий наприкінці документа, починаючи з нової сторінки, під заголовком «СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ». Він включає джерела, якими скористався студент у процесі виконання роботи і на які є посилання в тексті.

Літературні джерела у списку подаються в послідовності їх згадування у тексті документа. Порядкові номери описів у переліку є посиланнями в тексті (номерні посилання). Відомості про літературне джерело наводять на тій мові, на якій воно опубліковане.

Бібліографічне посилання містить наступні елементи:

а) для книг: прізвище і ініціали автора (або авторів); назва книги; місце видання – однією великою буквою «М.», «К», «Л.» відповідно для міст Москви, Києва і Ленінграда (стара назва) і цілком назва інших міст без слова «місто»; видавництво; рік видання і загальна кількість сторінок джерела, наприклад:

1. Шилдт, Герберт. Полный справочник по С++, 4-е издание.: Пер. с англ.: – М.: Изд. дом «Вильямс», 2004. – 800 с.;

б) для журнальних статей: прізвище і ініціали автора (авторів), повна назва статті, назва журналу, рік видання, випуск номера журналу, кількість сторінок статті, наприклад:

2. Бабій А.С. Програмна система // Вестник НТУ «ХПИ». Сборник научных трудов. Тематический выпуск «Информатика и моделирование». – Харьков: НТУ «ХПИ». – 2007. – № 19. с.12 – 16.;

г) для методичних вказівок: назва вказівки, прізвища і ініціали авторів, місце видання, видавництво, рік видання, кількість сторінок, наприклад:

3. Методические указания к выполнению лабораторных работ по курсу «Архитектура ЭВМ». Составители: Поворознюк А.И., Зыков И.С. – Харьков: ХПИ, 2000 – 79 с.;

д) для ресурсів інтернету: вказують прізвище автора статті, адресу сайта, назву статті, дату останнього відвідування сайта, наприклад:

4. Офіційний сайт SQLite / мова: англійська // <https://www.sqlite.org/>
25.05.2017 р.

3.2.10 Оформлення додатків

Додатки оформляють як продовження документа на наступних його аркушах. У додатках, що поміщають наприкінці пояснювальної записки, можуть бути включені матеріали допоміжного характеру, такі як: додаткові ілюстрації та таблиці, проміжні математичні докази, формули, розрахунки, тексти комп'ютерних програм, структури і змісти вхідних і вихідних даних (файлів, копій екранів з результатами роботи програми в текстовому і графічному вигляді), вид інтерфейсу користувача, інструкції, методики тестування.

Кожен додаток починається з нової сторінки і має заголовок, що записують малими літерами з першої великої, симетрично тексту. Посередині рядка над заголовком малими літерами, крім першої великої, вказують слово «Додаток» і велику букву, що позначає додаток. Додатки варто позначати послідовно великими буквами українського алфавіту, за винятком Є, З, І, Ї, Й, О, Ч, Ь.

Приклад:

Додаток А
Структура вхідних та вихідних даних

Текст кожного додатка може бути розбитий на розділи, підрозділи, пункти і підпункти з відповідною нумерацією згідно з зазначеними вище вимогами. Перед кожним номером проставляють позначення додатка (букву) і крапку, наприклад, А.2 – другий розділ додатка А; В.3.1 – підрозділ 3.1 додатка В; Д.4.1.2 – пункт 4.1.2 додатка Д і т.д.

Нумерація сторінок записки і додатків повинна бути наскрізна, причому у змісті пояснювальної записки необхідно перелічити всі додатки.

СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. ДСТУ 3008–95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення. – Чинний від 01.01.96 – Київ: Держстандарт України, 1995. – 37 с.
2. ГОСТ 7.1–2003, IDT : ДСТУ ГОСТ 7.1-2006. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання. – Чинний з 2007–07–01. – К. : Держспоживстандарт України, 2007.
3. СТВУЗ-ХПІ-3.01-2010. Текстові документи у сфері навчального процесу. Загальні вимоги до виконання. – Введ. 01.07.10.
4. ГОСТ 19.101–77. Единая система программной документации. Виды программ и программных документов. – Введ. с 01.01.80.
5. ГОСТ 19.103–77. ЕПКД. Обозначение программ и программных документов. – Введ. с 01.01.80.
6. ГОСТ 2.106–96. Единая система конструкторской документации. Текстовые документы. – Введ. с 13.11.96.
7. ГОСТ 19.105–78. ЕСПД. Общие требования к программным документам. – Введ. 01.01.1980.
8. ГОСТ 19.202–78. ЕСПД. Спецификация. Требования к содержанию и оформлению. – Введ. 01.01.80.
9. ГОСТ 19.401–78. ЕСПД. Текст программы. Требования к содержанию и оформлению. – Введ. с 01.01.80.
10. ГОСТ 19.402–78. ЕСПД. Описание программы. Требования к содержанию и оформлению. – Введ. с 01.01.80.
11. ГОСТ 19.505–79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению. – Введ. с 01.01.80.
12. ГОСТ 19.301–79. ЕСПД. Программа и методика испытаний. Требования к содержанию и оформлению. – Введ. с 01.01.81.
13. ГОСТ 19.701–90. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – Введ. с 01.01.92.

Додаток А

Приклад аркуша затвердження пояснювальної записки

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХПІ»

Кафедра «Обчислювальна техніка та програмування»

Розрахункове завдання з програмування

Тема: «РОЗРОБКА ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ»

Пояснювальна записка

КІТ.15А.20176-01 81 01-1 -А3

Розробники

Виконав:

студент групи КІТ-15А

_____ / Іванов О.А./

Перевірив:

_____ /доц. Далека В.Д./

Додаток Б

Приклад титульного аркуша пояснювальної записки

ЗАТВЕРДЖЕНО

КІТ.15А.20176-01 80 01-1 -АЗ

Розрахункове завдання з програмування

Тема: «РОЗРОБКА ІНФОРМАЦІЙНО-ДОВІДКОВОЇ СИСТЕМИ»

Пояснювальна записка

КІТ.15А.20176-01 81 01-1

Аркушів 26

Харків 2018

Додаток В
Приклад змісту пояснювальної записки

ЗМІСТ

Вступ.....	3
1 Огляд літературних джерел.....	3
1.1 Поняття «Інформаційна система».....	4
1.2 Постановка завдання до розробки.....	6
2 Розробка інформаційно-довідкової системи.....	7
2.1 Розробка структур даних.....	7
2.2 Розробка структури програми.....	7
2.3 Розробка алгоритмів програми.....	8
2.3.1 Розробка алгоритму головної функції.....	9
2.3.2 Розробка методів класу <i>Animal</i>	9
2.3.3 Розробка методів класу <i>Cow</i>	10
2.3.4 Розробка методів класу <i>Hen</i>	11
2.3.5 Розробка методів класу <i>MyFarm</i>	12
2.3.6 Розробка методів класу <i>ConsoleUtil</i>	13
2.3.7 Розробка методів класу <i>FarmTest</i>	14
Висновки.....	15
Список джерел інформації.....	16
Додаток А. Текст програми.....	17
Додаток Б. Результати роботи програми	21

Додаток Г

Приклад тексту програми

```

/**
 * @file ConsoleUtil.cpp
 * Реалізація класу для відображення діалогового меню.
 *
 * @author davs
 * @version 2.0.0
 * @date 02-feb-2017
 */
#include "ConsoleUtil.h"

void ConsoleUtil::showMenuAndProcess(MyFarm& farm)
{
    int choose = 0;
    do {
        cout << " Ферма 1.0 written by Davydov, Daleka, Molchanov"
             << endl << endl;
        cout << "0 - Вихід" << endl;
        cout << "1 - Читати дані з файлу" << endl;
        cout << "2 - Записати дані у файл" << endl;
        cout << "3 - Додати дані про тварину" << endl;
        cout << "4 - Пошук тварини за ім'ям" << endl;
        cout << "5 - Показати усіх тварин" << endl;
        cout << "6 - Сортування тварин за ростом" << endl;
        cout << "Зробіть вибір: ";
        cin >> choose;
        switch (choose)
        {
            case 0:
                cout << "До побачення!" << endl;
                break;
            case 1:
                cout << "Ця функція не реалізована!" << endl;
                break;
            case 2:
                cout << "Ця функція не реалізована!" << endl;
                break;
            case 3:
                cout << "Ця функція не реалізована!" << endl;
                break;
            case 4:
                cout << "Ця функція не реалізована!" << endl;
                break;
            case 5:
                for (int i = 0; i < farm.getAnimalCount(); i++)
                {
                    cout << farm[i]->getInfo() << endl;
                }
                break;
            case 6:
                cout << "Ця функція не реалізована!" << endl;
                break;
            default:
                cout << "Поганий вибір. Спробуйте ще" << endl;
        }
    } while (choose != 0);
}

```

Додаток Д

Приклад структури програми та схеми алгоритму

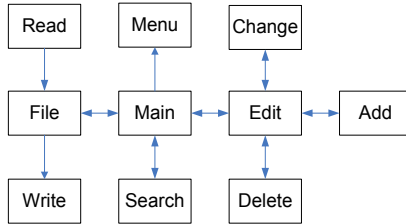


Рисунок Д.1 – Структура інформаційно-довідкової системи

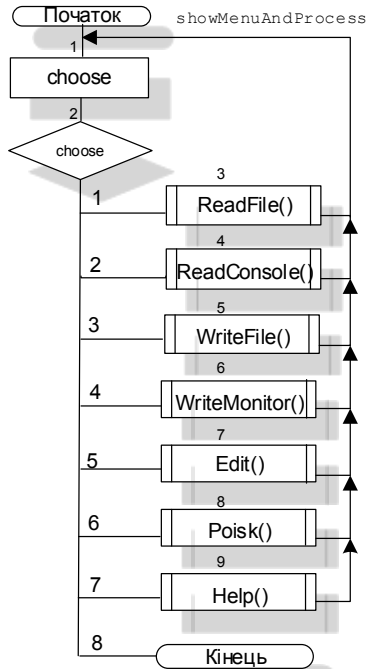


Рисунок Д.2 – Схема алгоритму функції showMenuAndProcess

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ
до виконання розрахункового завдання
з дисципліни «Програмування»
для студентів першого курсу всіх форм навчання
спеціальності «Комп'ютерна інженерія»

Укладачі: ДАВИДОВ Вячеслав Вадимович
ДАЛЕКА Валентина Дмитрівна
МОЛЧАНОВ Георгій Ігорович

Відповідальний за випуск проф. С. Г. Семенов

Роботу до видання рекомендував проф. М. Й. Заполовський

Редактор О. І. Шпільова

План 2018 р., поз. 136

Підп. до друку 20.06.2018 Формат 60x84 1/16. Папір офсетний.

Друк – ризографія. Гарнітура Times New Roman. Ум.-друк.арк 1,8.

Наклад 100 прим. Зам. № 07-18. Ціна договірна.

Видавничий центр НТУ «ХПІ»

Свідоцтво про державну реєстрацію ДК № 5478 від 21.08.2017 р.
61002, Харків, вул. Кирпичова, 2

Надруковано ТОВ «Видавництво «Форт»

Свідоцтво про внесення до Державного реєстру видавців
ДК №333 від 09.02.2001 р.

61023, м. Харків, а/с 10325. Тел. (057)714-09-08