

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Циклова комісія комп'ютерних та інформаційних дисциплін

КУРСОВИЙ ПРОЕКТ

з дисципліни: "Об'єктно-орієнтоване програмування"

Тема проекту: "Розробка прикладної програми з використанням класу
«СОБАКА»"

Студентки 2 курсу ОПК-217 групи
напряму підготовки 6.050101 комп'ютерні науки
спеціальності 5.05010101 "Обслуговування
програмних систем і комплексів"
Хахалєвої Євгенії Вікторівни
Керівник: викладач Коломієць П.Ю.

Національна шкала _____

Кількість балів: – Оцінка: ECTS –

Голова комісії: _____ П.Ю. Коломієць

Члени комісії: _____ Р.І. Манєва

_____ Г.О. Кононова

м. Харків 2019

ВІДОМІСТЬ ДОКУМЕНТІВ

[illegible]

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"**

**Спеціальність: 5.05010101 "Обслуговування програмних систем та комплексів"
Циклова комісія "Комп'ютерних та інформаційних дисциплін"**

**Завдання на курсовий проект студентки групи ОПК-217 Хахалєвої Є.В.
Тема: "Собака"**

Обов'язкові поля:

- Номер по порядку (призначається автоматично);
- Порода (вибирається випадковим чином з заготовлених варіантів зчитуються з файлу);
- Кличка (вибирається випадковим чином з заготовлених варіантів зчитуються з файлу);
- Вік (генерується випадковим чином);
- Місце зайняте на вставці (вибирається випадково, таким чином, щоб не було однакових місць і всі місця були зайняті);
- Кількість тварин однієї породи (по одній змінної на кожну можливу породу (або масив з відповідною кількістю елементів));
- Лічильник кількості примірників.
-

Можливості класу:

- Автоматична ініціалізація при створенні екземпляра;
- Можливість ручної ініціалізації не вирахував полів при додаванні примірника;
- Висновок даних на екран;
- Ручна зміна зайнятого місця (зі зміною місць інших тварин).

Можливості програми:

- Вибір початкової кількості примірників (не більше 20);
- Висновок всіх примірників по порядку;
- Висновок одного примірника з заданим порядковим номером;
- Висновок всіх примірників відсортованих за алфавітом;
- Висновок всіх примірників по колу, починаючи з будь-якого;
- Висновок примірників за різними ознаками (породі, віком та ін.);
- Висновок призових місць;
- Додавання / видалення примірників;
- Меню користувача, що дозволяє використовувати всі можливості програми і класу, дізнатися інформацію про автора, а також вийти з програми.

Вимоги до програми:

- Організація даних у вигляді списку (по можливості кругового);
- Обов'язкове використання принципів ООП (особливо інкапсуляції);
- Максимально передбачені помилки користувача;
- Використання статичних методів;
- Гарне оформлення.

Програма:

Запросити користувача кількість примірників класу (не більше 20) і створити список обраної ним довжини. Потім запропонувати користувачеві меню, що дозволяє запускати публічні методи (крім конструктора і деструктора), виводити дані на екран, дізнатися інформацію про автора, а також вийти з програми. Меню програми заиклити.

Завдання видано: 21 березня 2019 р.

Термін захисту курсового проекту: травень 2019 р.

Заступник директора

з навчальної роботи

Голова циклової комісії

Керівник курсового проекту

_____ Ігнатенко О.І.

_____ Коломієць П.Ю.

_____ Коломієць П.Ю.

РЕФЕРАТ

Звіт до КП: 64 с., 5 рис., 1 табл., 10 джерел інформації, 3 додатки.

Ключові слова: ПРОГРАМУВАННЯ, ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ, ООП, КЛАСИ, ОБ'ЄКТИ, ЕКЗЕМПЛЯРИ, ПОЛЯ, ВЛАСТИВОСТІ, МЕТОДИ, ФУНКЦІЇ, ЗМІННІ, КОНСТАНТИ, МАСИВИ, ДИНАМІЧНІ МАСИВИ, C++, VISUAL STUDIO, MICROSOFT, СПИСОК, ПРОГРАМА, СОБАКА, ЕЛЕМЕНТИ, ПРОЦЕДУРИ.

Об'єктом дослідження являються класи та їх властивості на прикладі мови програмування C++.

Мета роботи – розробити прикладну програму, що дозволить побудувати на основі класу найпростішу електронну таблицю, що містить собак, яка дозволить створювати, змінювати та виводити на екран дані з неї.

Розробка виконувалася на мові програмування C++, у якості середовища розробки було вибрано Microsoft Visual Studio 2017., також використовувалися різноманітні можливості мови програмування C++, такі як: виведення тексту в консольне вікно, зчитування даних з файлу, використання класів, робота з рядками та векторами, елементарні арифметичні дії з числами, оператори розгалуження та цикли.

Розроблена програма зі зрозумілим інтерфейсом, яка призначена для формування списку собак, які зайняли місце на виставці. У ході програми користувач може змінювати, добавляти та видаляти дані у списку.

Користувачеві надається зручне меню, за допомогою якого він буде користуватися даним списком. У програмі передбачена максимальна кількість помилок користувача різних типів.

РЕФЕРАТ

Отчет к КП: 64 с., 5 рис., 1 табл., 10 источников информации, 3 приложения.

Ключевые слова: ПРОГРАММИРОВАНИЕ, ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ, ООП, КЛАССЫ, ОБЪЕКТЫ, ЭКЗЕМПЛЯРЫ, ПОЛЯ, СВОЙСТВА, МЕТОДЫ, ФУНКЦИИ, ПЕРЕМЕННЫЕ, КОНСТАНТЫ, МАСИВЫ, ДИНАМИЧЕСКИЕ МАСИВЫ, C++, VISUAL STUDIO, MICROSOFT, СПИСОК, ПРОГРАММА, СОБАКА, ЭЛЕМЕНТЫ, ПРОЦЕДУРЫ.

Объектом исследования являются классы и их свойства на примере языка программирования C ++.

Цель работы – разработать приложение, которое позволит построить на основе класса простейшую электронную таблицу, содержащую собак, которая позволит создавать, изменять и выводить на экран данные из нее.

Разработка выполнялась на языке программирования C++, в качестве среды разработки был выбран Microsoft Visual Studio 2017., также использовались различные возможности языка программирования C++, такие как: вывод текста в консольное окно, считывание данных из файла, использование классов, работа с строками и векторами, элементарные арифметические действия с числами, операторы ветвления и циклы.

Разработана программа с понятным интерфейсом, которая предназначена для формирования списка собак, которые заняли место на выставке. В ходе программы пользователь может изменять, добавлять и удалять данные в списке.

Пользователю предоставляется удобное меню, с помощью которого он будет пользоваться данным списком. В программе предусмотрена максимальное количество ошибок пользователя различных типов.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"**

Циклова комісія комп'ютерних та інформаційних дисциплін

ЗАТВЕРДЖЕНО
Заступник директора
з навчальної роботи
_____ О.І. Ігнатенко

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КУРСОВОГО ПРОЕКТУ**

з дисципліни: "Об'єктно-орієнтоване програмування"

Тема: "Розробка прикладної програми з використанням класу «СОБАКА»"

Шифр ХКТК – ОПК217.5.05010101.22

Виконав студент гр. ОПК-217_____Хахалевої Є.В.

Керівник _____Коломієць П.Ю.

м. Харків 2019 р.

ЗМІСТ

Вступ.....	4
1 Огляд предметної області та постановка задачі.....	5
1.1 Предметна область.....	5
1.2 Технічне завдання.....	5
1.3 Програмні засоби.....	6
2 Алгоритмічне забезпечення програмного засобу.....	8
2.1 Вхідні данні та розрахунки.....	8
2.2 Опис побудови списку у програмному засобі.....	9
3 Керівництво користувача.....	10
Висновки.....	13
Перелік джерел інформації.....	14
Додаток А Блок-схема програми.....	15
Додаток Б Код програми.....	34
Додаток В Ілюстративні матеріали.....	45

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ООП – Об’єктно-орієнтоване програмування.

ОС – операційна система.

ПЗ – програмне забезпечення.

MS – Microsoft.

VS – Visual Studio.

ВСТУП

Курсова робота, основні положення якої будуть розглядатися у розділах написана на мові C++.

У розділах буде йтися мова про предметну область курсової роботи, засоби розробки даної роботи, тощо

Написана програма буде включати в себе роботу з файлами, векторами, рядками, класами, методами, полями, статичними змінними, операторами розгалуження та циклами, побудову алгоритму, передбачти помилки. Також необхідно слідувати «гарному стилю програмування» та принципам ООП.

Буде пояснюватися яким саме чином отримані вхідні дані та чому вони саме такі. Будуть наведені формули, які застосовувалися у завданні, детальний опис усіх полей та методів класу та взаємозалежність між ними. Також буде наведена блок-схема, з якої видно основний алгоритм дій. Розглянемо основні положення меню користувача зі ілюстраціями та поясненнями.

Особа увага буде приділена опису побудови списку у даному програмному засобі. Буде пояснено як саме влаштован список у програмі, що з ним можна робити і як це буде працюватиме. Обов'язковим також буде розділ з інструкцією до розробленої програми з ілюстраціями та необхідними поясненнями.

Метою цього курсового проекту є: закріплення знань з принципів ООП та програмування на C++, які ми здобули під час проходження курсу «Об'єктно-орієнтованого програмування».

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Предметна область

Предметною областю даного курсового проекту є виставка собак. На виставці кожній собаці надається місце, яке вона заняла на цій виставці. Разом з інформацією про заняте місце у списку собак обов'язково повинні бути прізвисько, порода та рік собак.

Виставка собак - публічний захід з оцінки відповідності собак стандартам певної породи, з метою виявлення кращого представника породи і їх подальшого розведення. Оцінку проводять експерти, які мають відповідну ліцензію.

Порівняння собак проводиться в різних класах відповідно до їх статі і віку. Експерт оглядає кожну собаку окремо на предмет відповідності стандарту породи, наявності недоліків і дискваліфікує вад. Диктує асистентові в рингу опис кожної тварини, виставляє оцінку від "незадовільно" до "відмінно". Потім, в кожному класі визначаються призери і переможці, після чого переможці класів порівнюються на звання «Кращий юніор» і «Кращий кобель», «Кращий ветеран». Володарі цих титулів потім змагаються за звання «Кращого представника породи».

1.2 Технічне завдання

Завданням цього курсового проекту було створити клас «Собака», який буде включати такі приховані поля: номер за порядком, породу, прізвисько, кількість однакових пород, рік собаки. Також створити лічильник кількості примірників та колекцію лічильників для кожного з відтінків, які будуть публічними статичними полями.

Дати можливість користувачеві змінювати будь-який з примірників класу, видаляти та добавляти елемент. Можливість вивести список повністю, за алфавітом та по колу. Також користувач має змогу побачити призові місця, які

займають собаки та може запросити список собак за ознакою(порода, прізвисько, рік і т.д.).

Користувачеві також треба надати зручне меню, за допомогою якого він буде вибирати, що він хоче зробити з списком. У написаній програмі користувач має змогу вийти з програми, натиснувши на клавішу esc. Також він має змогу побачити автора програми, вибравши відповідний пункт меню.

Дані про собак, які зайняли місце на виставці повинні бути організовані у вигляді списку(бажано кругового). Ініціалізація даних повинна бути автоматичною. Програму необхідно зациклювати. Повинна бути передбачена максимальна кількість помилок користувача різних типів.

1.3 Програмні засоби

При виконанні курсового проекту використовувалися мова програмування C++ та середовище розробки Microsoft Visual Studio 2017.

C++ - компільований, статично типізований мова програмування загального призначення.

Підтримує такі парадигми програмування, як процедурне програмування, об'єктно-орієнтоване програмування, узагальнене програмування. Мова має багату стандартну бібліотеку, яка включає в себе поширені контейнери і алгоритми, введення-виведення, регулярні вирази, підтримку багатопоточності і інші можливості. C++ поєднує властивості як високорівневих, так і низькорівневих мов. У порівнянні з його попередником - мовою C, - найбільшу увагу приділено підтримці об'єктно-орієнтованого і узагальненого програмування.

C++ – одна з наймогутніших та найпопулярніших мов програмування, яка використовується для розробки ОС, програм, ігор, ПЗ для серверів, а також драйверів різноманітних пристроїв.

C++ – це мова програмування, розроблена Б'ярном Страуструпом у 1979 році, котра спочатку називалася «Сі з класами», а згодом, через 4 роки, була

названа «C++». Ця мова програмування базується на C та унаслідувала багато бібліотек та синтаксис цієї мови. Та на відміну від C, C++ підтримує ООП.

Існує досить багато реалізацій мови C++ для різних ОС, але в курсовому проєкті використовувалася MS Visual C++, та середовище розробки MS Visual Studio 2017.

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

За допомогою VS дуже легко писати програми, так як це середовище розробки містить досить «продвинутий» редактор коду, зручні інструменти для відладки та компілятор, тобто для написання програми достатньо лише Visual Studio.

2 АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОГО ЗАСОБУ

2.1 Вхідні дані та розрахунки

1. Номер за порядком – призначається автоматично;
2. Рік собаки – генерується випадковим чином за допомогою функції rand();
3. Прізвисько – зчитується з файлу;
4. Порода – зчитується з файлу.
5. Місце на виставці – генерується випадковим чином з вектора.
6. Лічильник кількості примірників – являє собою статичну змінну, яка інкрементується, при запуску конструктора, та декрементується, при запуску деструктора.
8. Кількість однакових порід – статичний масив, заповнюється таким чином: у програмі створено метод, який інкрементує один з елементів масиву залежно від назви породи. При запуску деструктора відповідний елемент масиву декрементується. Наприклад:

Таблиця 2.1

Номер	Порода	Рік	Прізвисько	Місце на виставці	Однакових порід
1	Мопс	2	Бобік	5	1
2	Мопс	10	Шарик	1	2

В залежності від кількості елементів вектор заповнюється місцями на виставці. Потім генерується випадковий індекс вектора та значення під цим індексом присвоюється полю. Також в залежності від породи собаки, якщо цих порід більше ніж одна однакових, лічильник кількості однакових порід збільшується.

2.2 Опис побудови списку у програмному засобі

Колекцію в програмі було реалізовано за допомогою списку. Список у свою чергу був реалізований за допомогою класу, в якому були всі необхідні методи для роботи із списком (добавлення, видалення і т.д.).

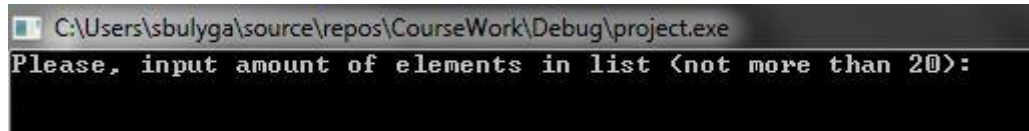
В інформатиці, список (англ. List) - це абстрактний тип даних, що представляє собою упорядкований набір значень, в якому деяке значення може зустрічатися більше одного разу. Примірник списку є комп'ютерної реалізацією математичного поняття кінцевої послідовності. Примірники значень, що знаходяться в списку, називаються елементами списку (англ. Item, entry або element); якщо значення зустрічається кілька разів, кожне входження вважається окремим елементом.

Терміном список також називається кілька конкретних структур даних, що застосовуються при реалізації абстрактних списків, особливо зв'язкових списків.

В список, який використовується у програмі, можна додати елемент та видалити. Також його можна вивести на екран повністю за порядком, за алфавітом, по колу, починаючи з будь-якого елементу. Можна вивести елемент за його порядковим номером та елементи за ознакою (порода, прізвище, рік і т.д.). Який саме метод буде викликатися з класу залежить від того, який пункт меню вибере користувач.

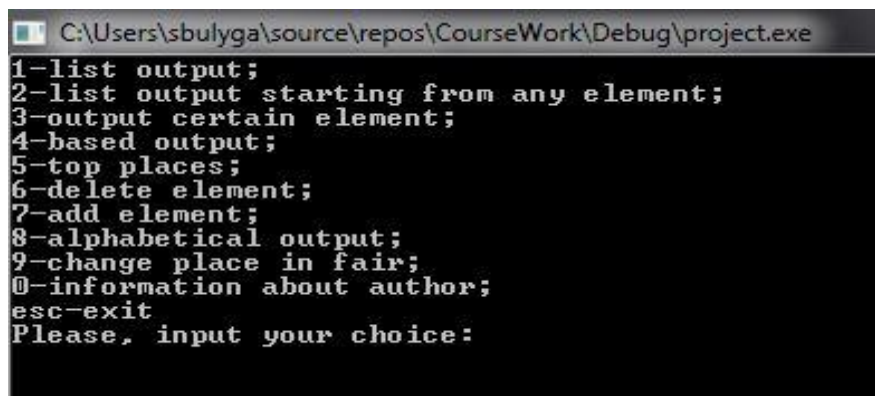
3 КЕРІВНИЦТВО КОРИСТУВАЧА

При запуску програми користувач бачить запит про кількість екземплярів (мал 3.1).



Малюнок 3.1 – Запит про кількість примірників

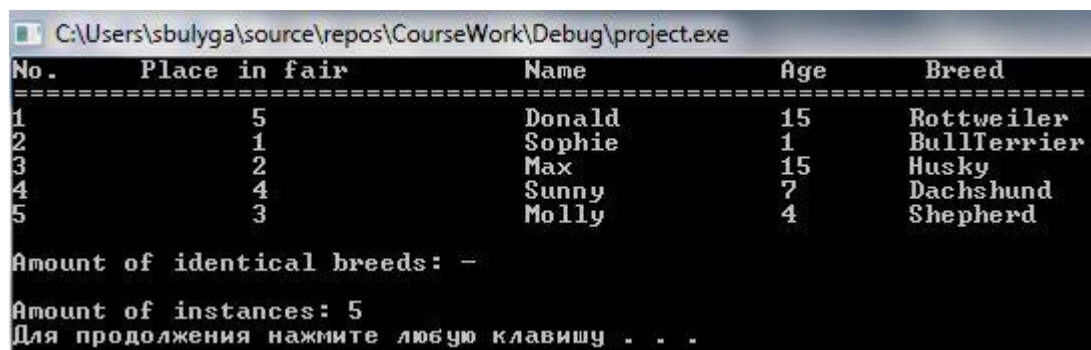
Потім користувач попадає у меню (мал 3.2).



Малюнок 3.2 – Меню

При виборі:

1. "list output" користувачу на екран виводиться весь список за порядком(мал 3.3).



Малюнок 3.3 – Вивід усіх примірників

2. “list output starting from any element” користувач бачить запит про те, з якого елемента розпочати. Потім виводиться весь список (мал 3.4).

No.	Place in fair	Name	Age	Breed
3	2	Max	15	Husky
4	4	Sunny	7	Dachshund
5	3	Molly	4	Shepherd
1	5	Donald	15	Rottweiler
2	1	Sophie	1	BullTerrier

Малюнок 3.4 – Вивід по колу

3. “output certain element” користувач бачить запит про номер елемента, який він хочить побачити.

4. “based output” користувач бачить запит на чому буде базуватися вивід. В залежності від вибору будить виведен елемент або елементи з заданою ознакою.

5. “top places” користувач бачить призові місця, які зайняли собаки на виставці (мал. 3.5)

No.	Place in fair	Name	Age	Breed
2	1	Sophie	1	BullTerrier
3	2	Max	15	Husky
5	3	Molly	4	Shepherd

Малюнок 3.5 Вивід призових місць

6. “delete element” користувач бачить запит про елемент, який він хоче видалити.

7. “add element” користувач бачить запит про те як він хоче додати елемент (ввести самому або вибрати) і в залежності від вибору добавляється елемент.

8. “alphabetical output” виведення списку за алфавітом.

9. “change place in fair” користувач бачить запит про елемент, місце якого на виставці він хоче змінити та нове місце.
10. “information about author” користувач бачить інформацію про автора програми.
11. esc “exit” вихід з програми.

ВИСНОВОК

Курсова робота, основні положення якої були розглянуті у розділах написана на мові C++. У розділах йшлися мова про предметну область курсової роботи, засоби розробки даної роботи, тощо.

Написана програма включала в себе роботу з файлами, векторами, рядками, класами, методами, полями, статичними змінними, операторами розгалуження та циклами, побудову алгоритму, передбачала помилки.

Пояснювалося яким саме чином отримані вхідні дані та чому вони саме такі. Були наведені формули, які застосовувалися у завданні, детальний опис усіх полів та методів класу та взаємозалежність між ними. Розглянуті основні положення меню користувача зі ілюстраціями та поясненнями.

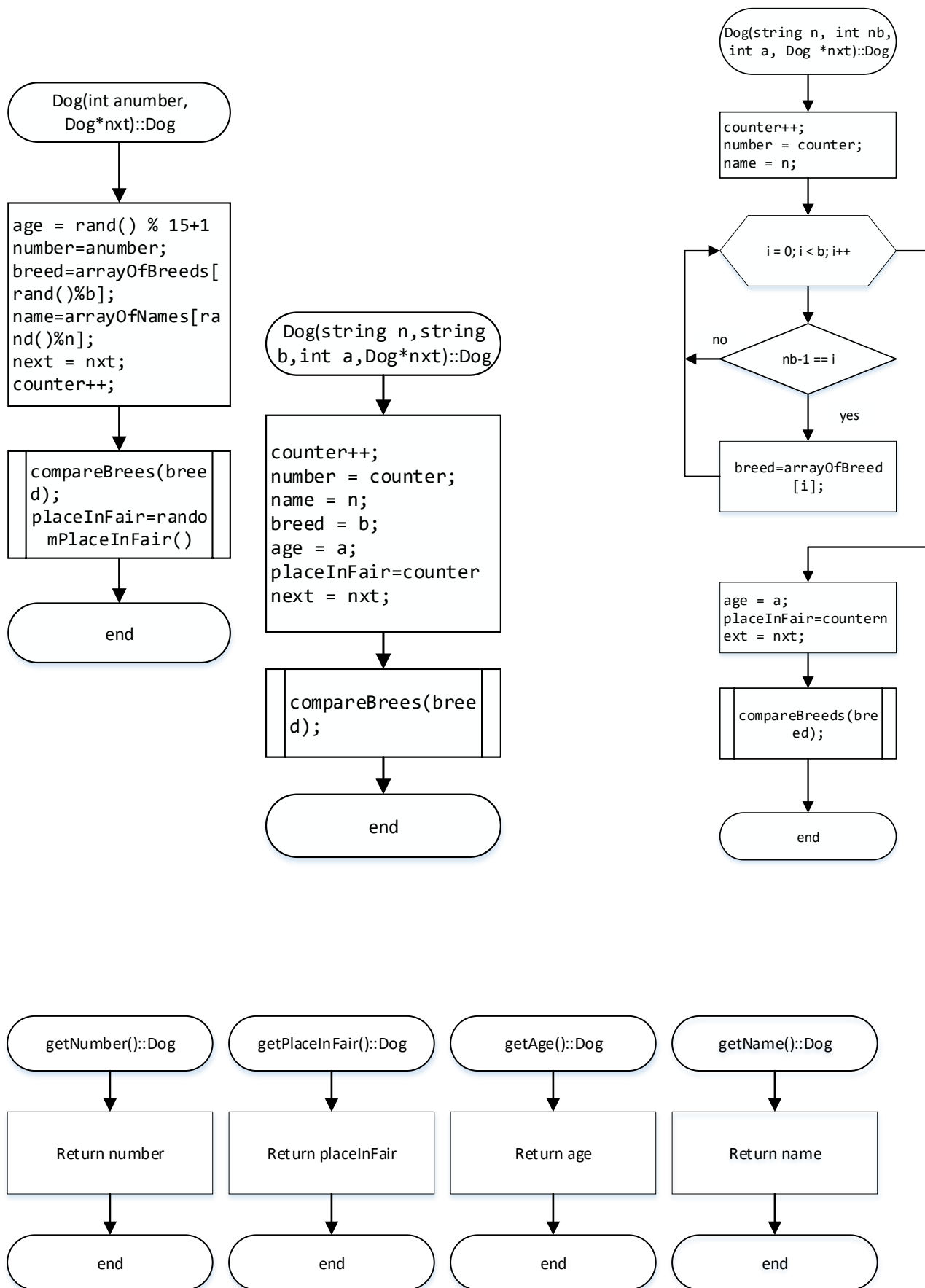
Особа увага приділялася опису побудови списку у даному програмному засобі. Пояснено як саме влаштован список у програмі, що з ним можна робити і як це буде працюватиме. Обов'язковим також був розділ з інструкцією до розробленої програми з ілюстраціями та необхідними поясненнями.

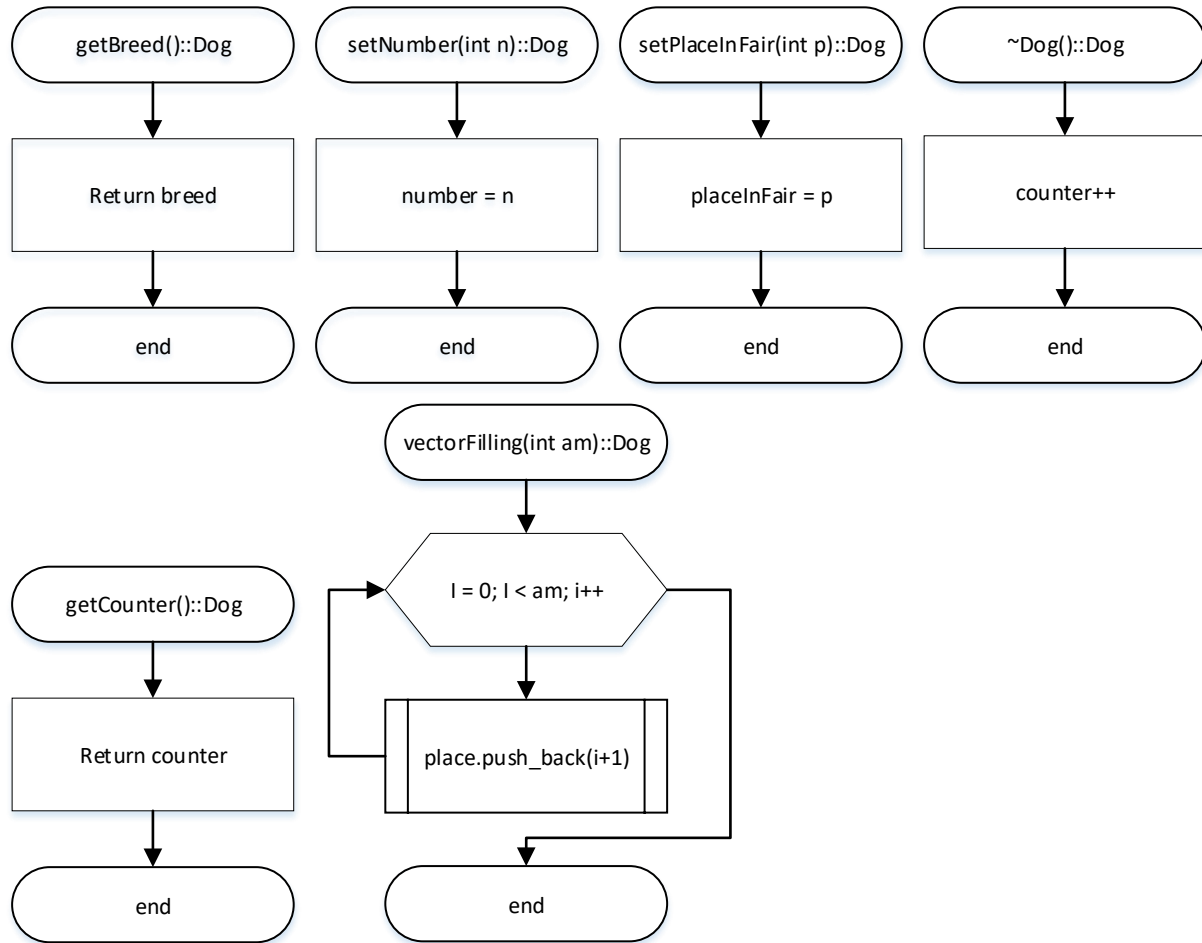
Метою цього курсового проекту було закріплення знань з принципів ООП та програмування на C++, які ми здобули під час проходження курсу «Об'єктно-орієнтованого програмування».

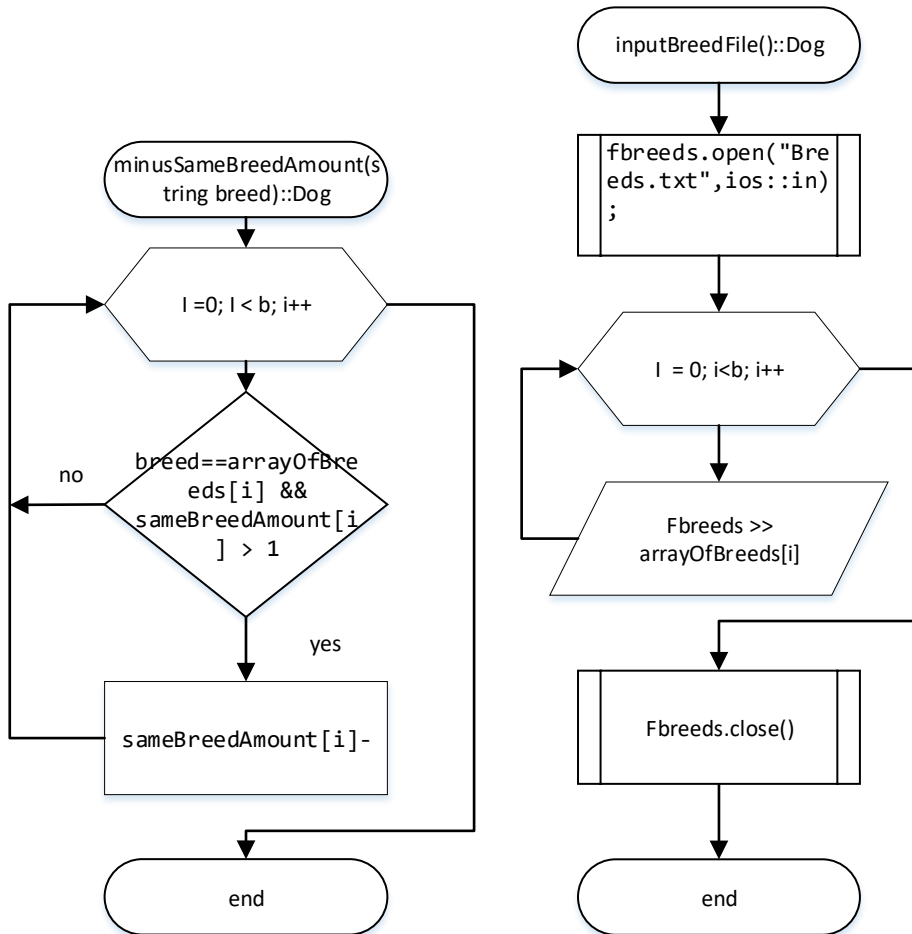
ПЕРЕЛІК ДЖЕРЕЛ ІНФОРМАЦІЇ

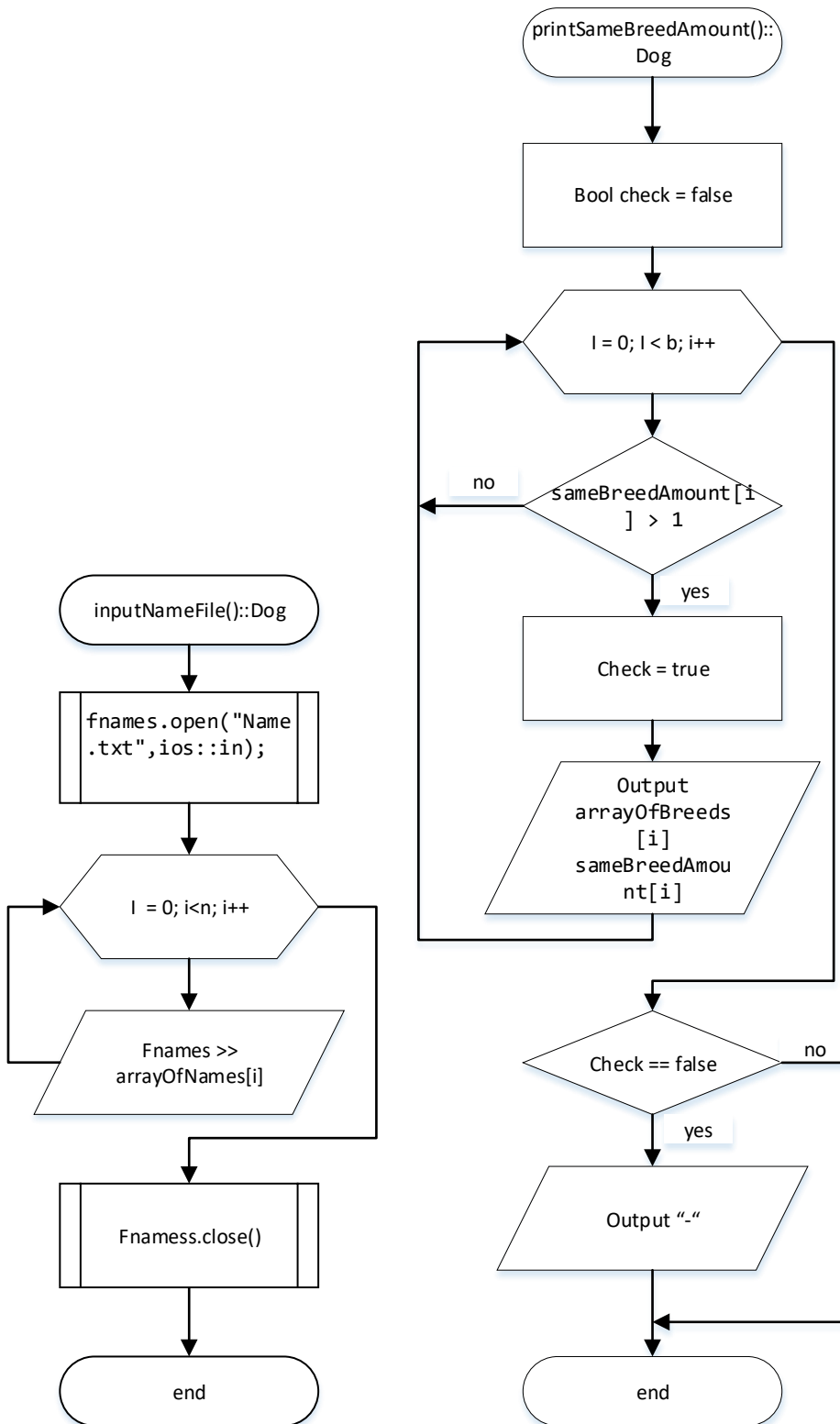
- 1 Буч Г. Объектно-ориентированный анализ и проектирование. – СПб: Бином, Невский диалект, 1998.
- 2 Архангельский А.Я. Программирование в С++ Builder 6. – М.: «Бином», 2003.
- 3 Хортон А. Visual C++ 2005: базовый курс. – М: «Вильямс», 2007.
- 4 Макконнелл С. Совершенный код. – СПб: Питер, 2007.
- 5 Уроки программирования С++ с нуля // [Электронный ресурс] // режим доступа: <https://code-live.ru/tag/cpp-manual/>
- 6 С++ для начинающих // [Электронный ресурс] // режим доступа: <http://valera.asf.ru/cpp/book/>
- 7 Основы программирования С/С++ // [Электронный ресурс] // режим доступа: <http://cppstudio.com/>
- 8 Язык программирования С++ // [Электронный ресурс] // режим доступа: <http://cppstudio.com/cat/274/>
- 9 С++ // [Электронный ресурс] // режим доступа: <http://kvodo.ru/category/lessons/cplusplus>
- 10 Википедия // [Электронный ресурс] // режим доступа: <https://ru.wikipedia.org/>

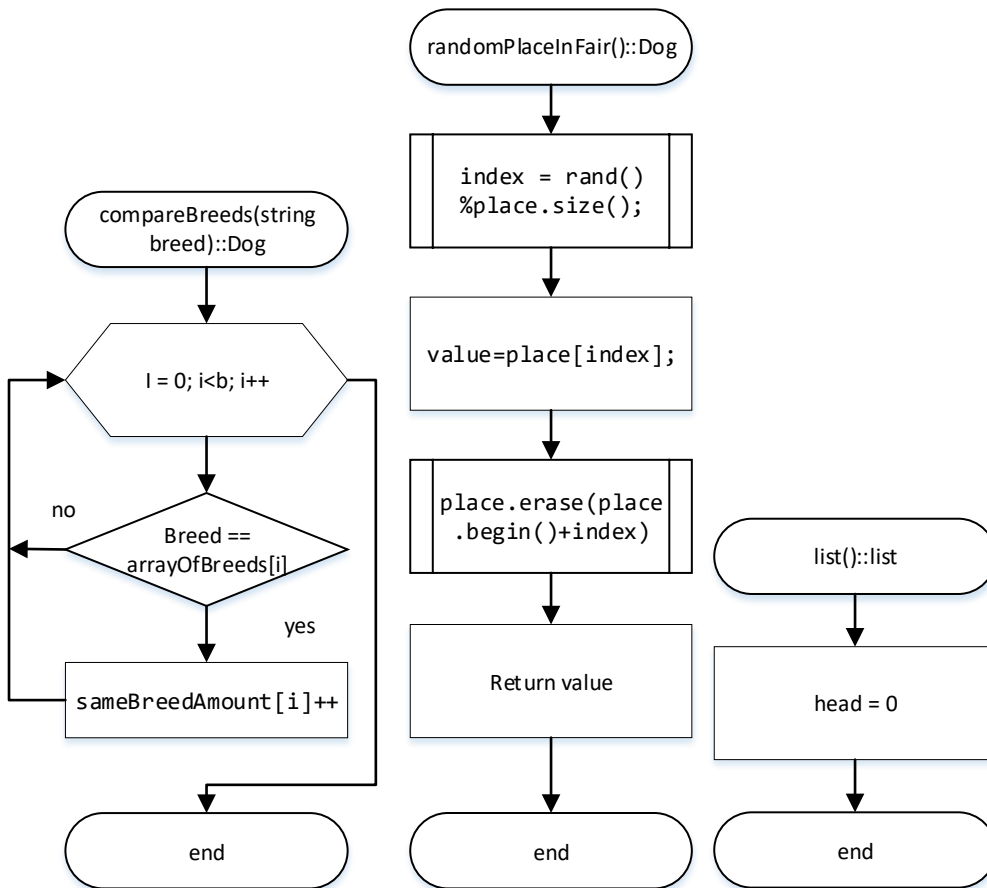
ДОДАТОК А БЛОК-СХЕМА ПРОГРАМИ

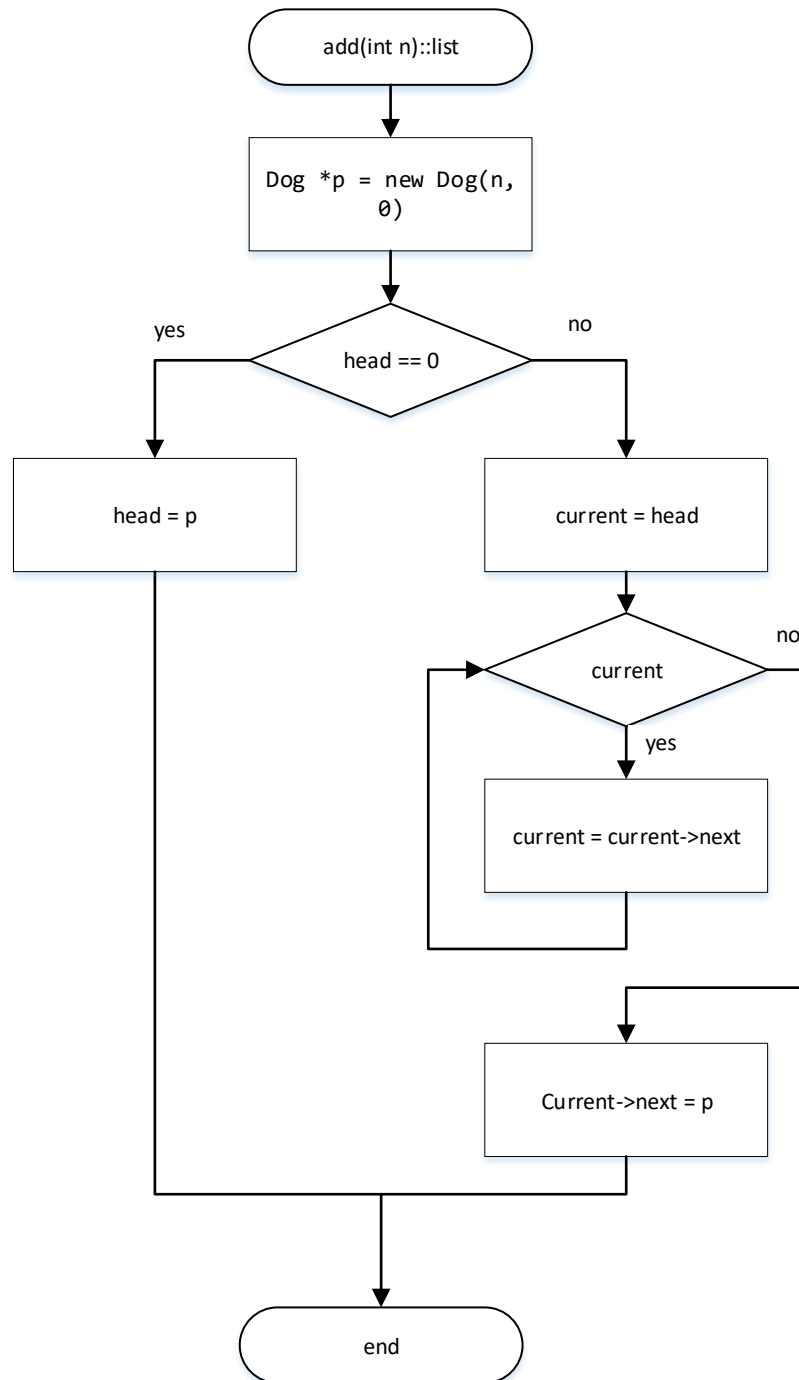


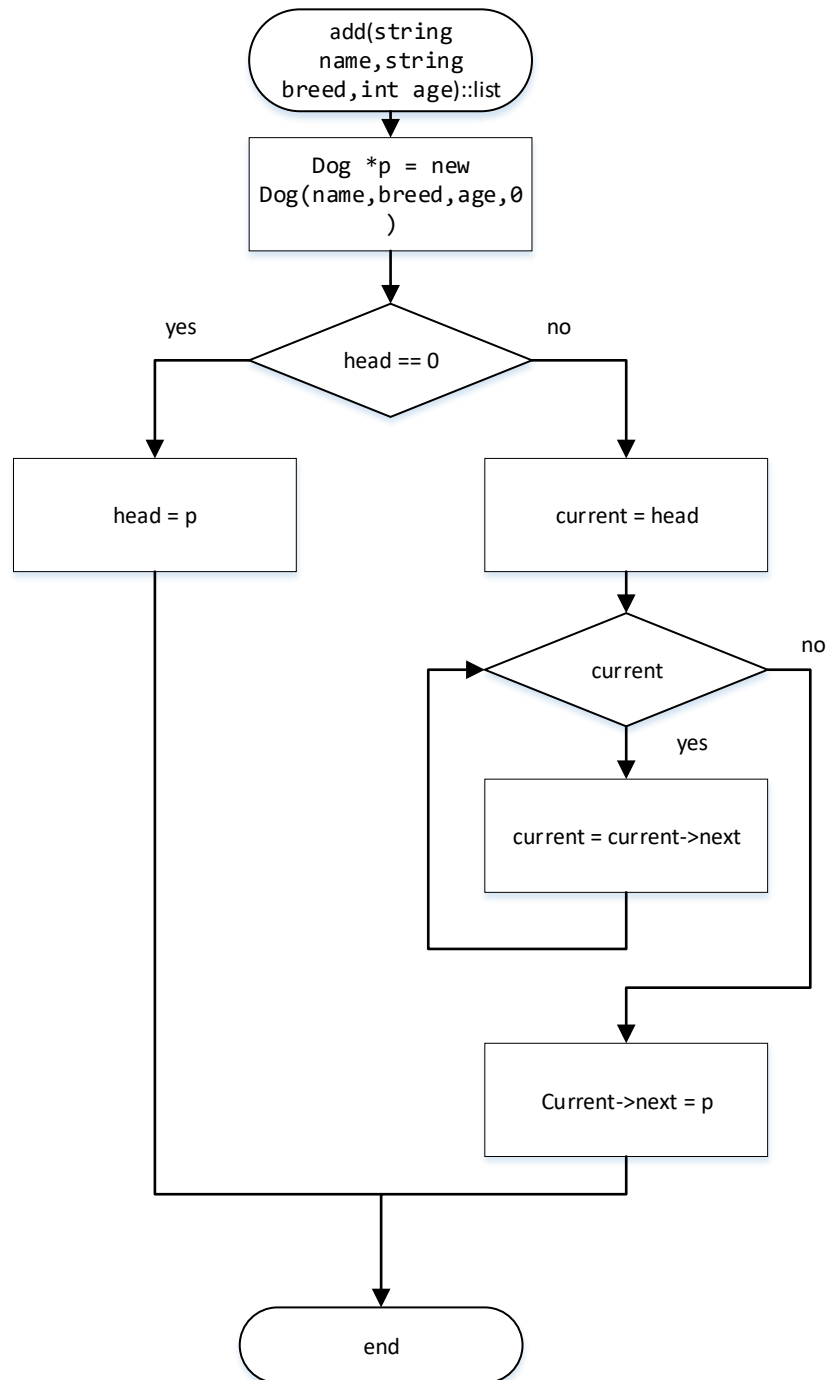


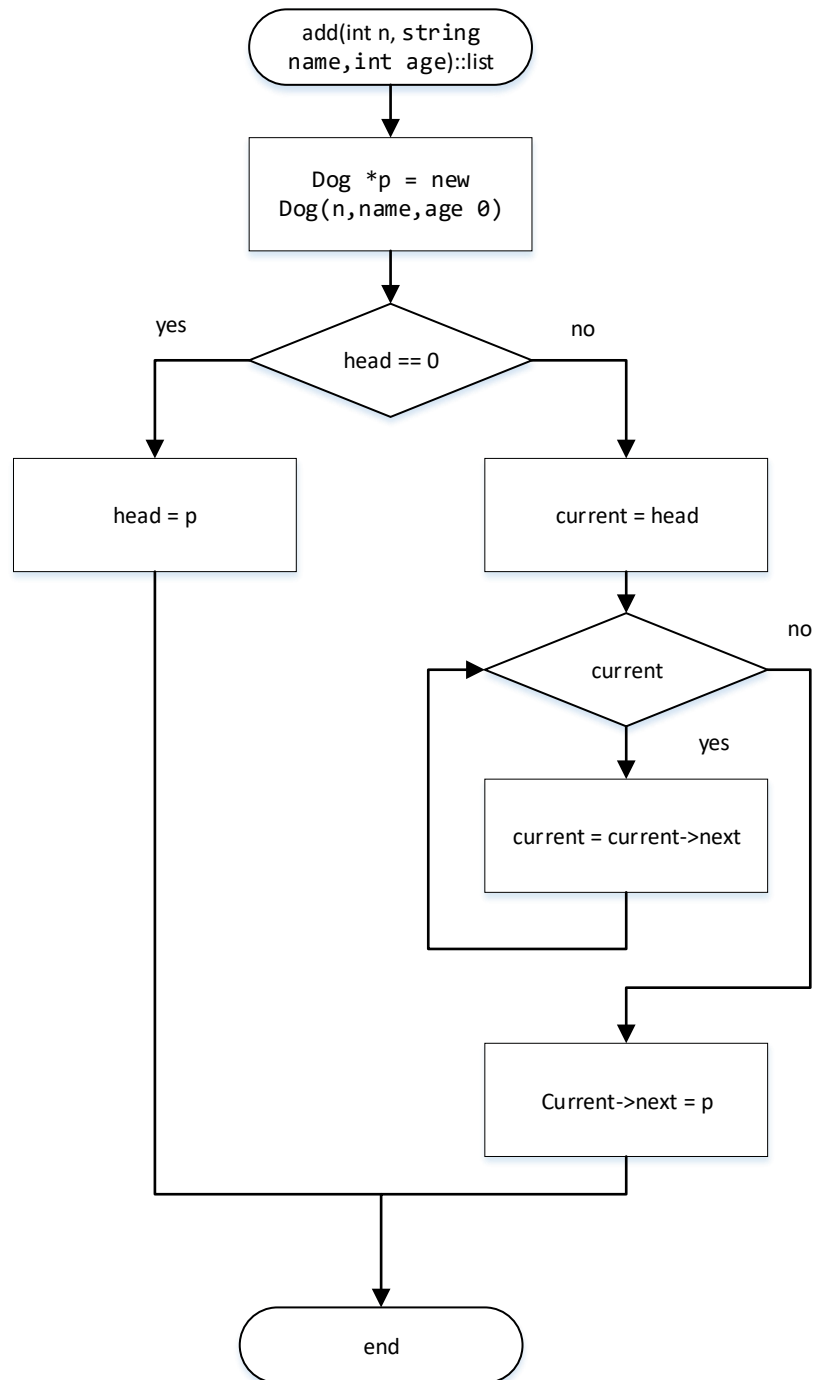


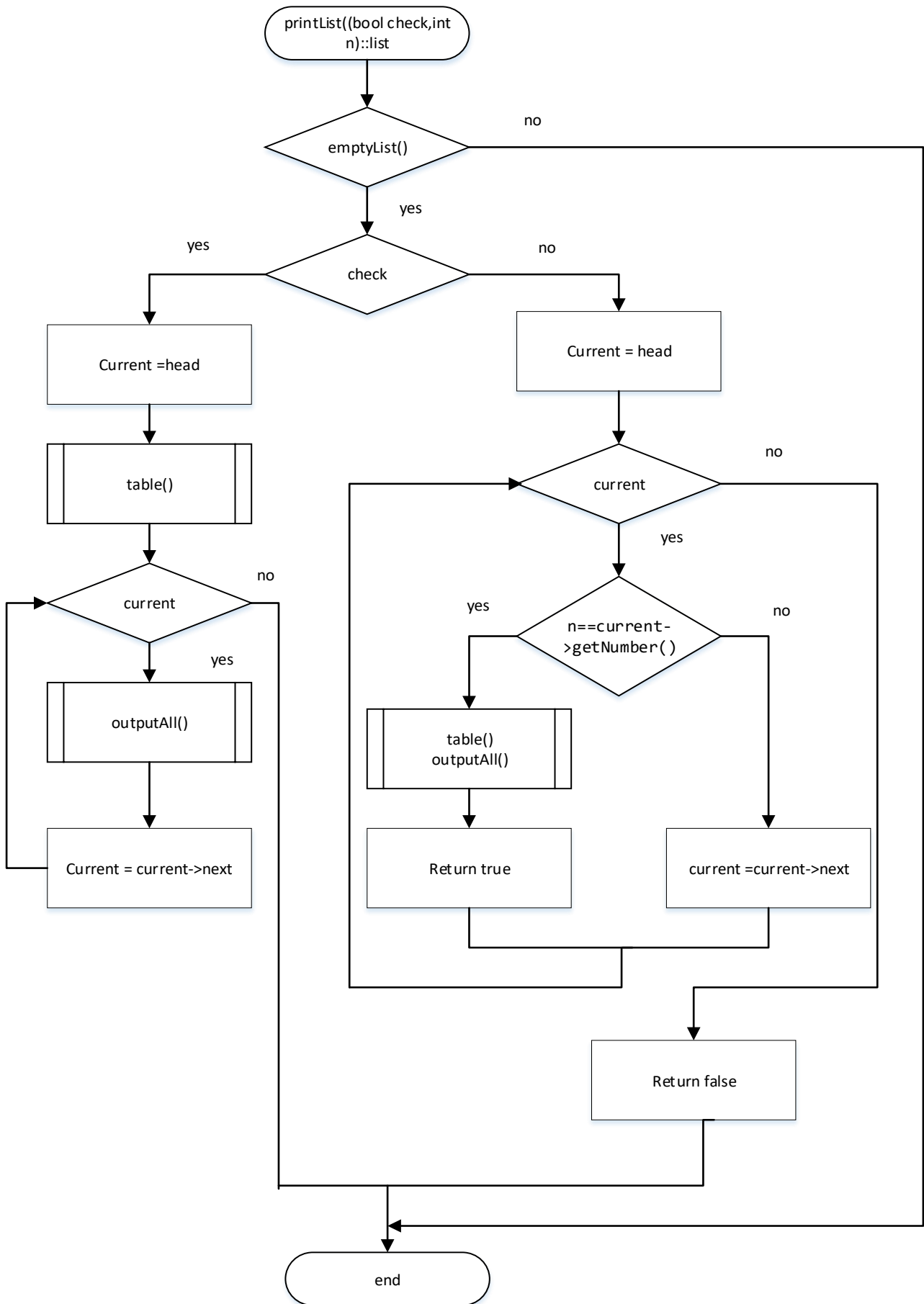


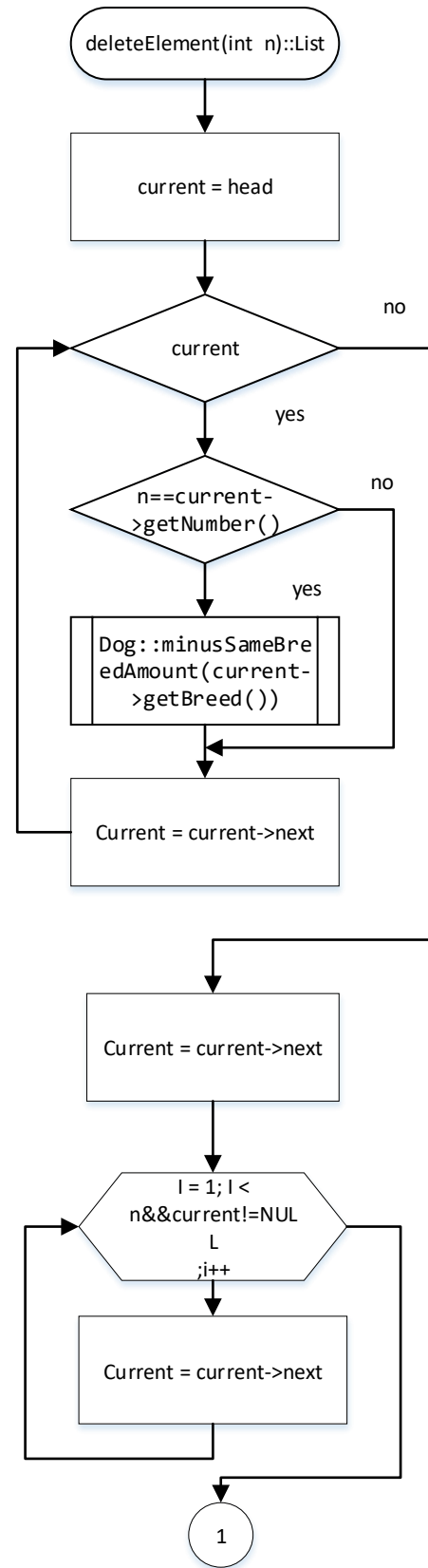
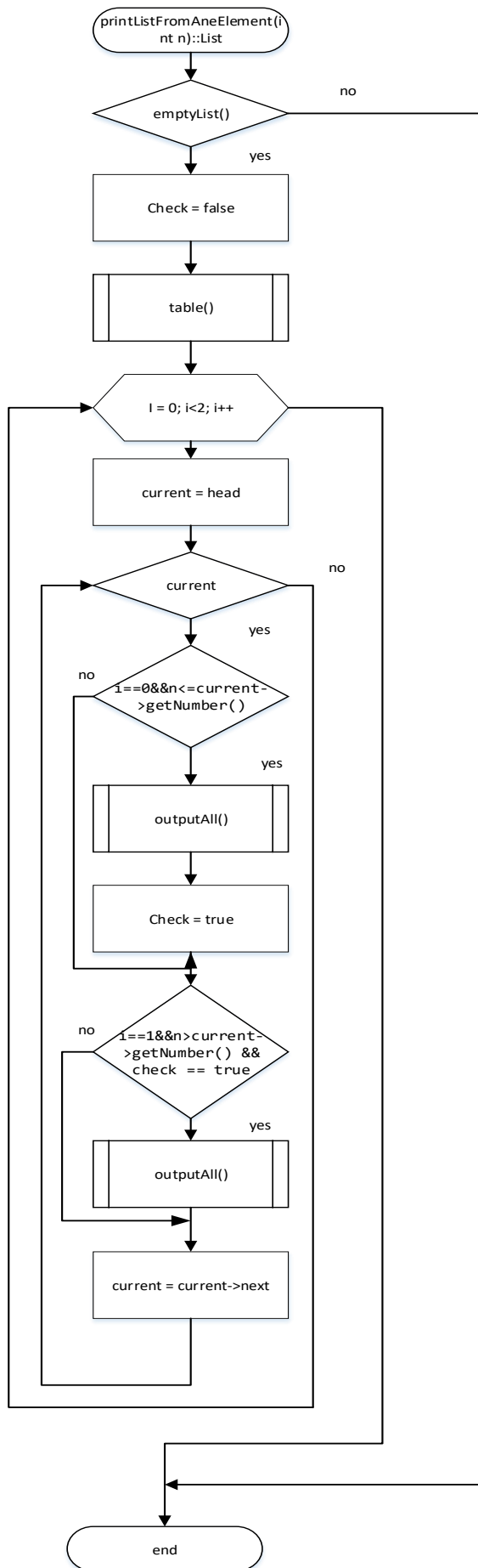


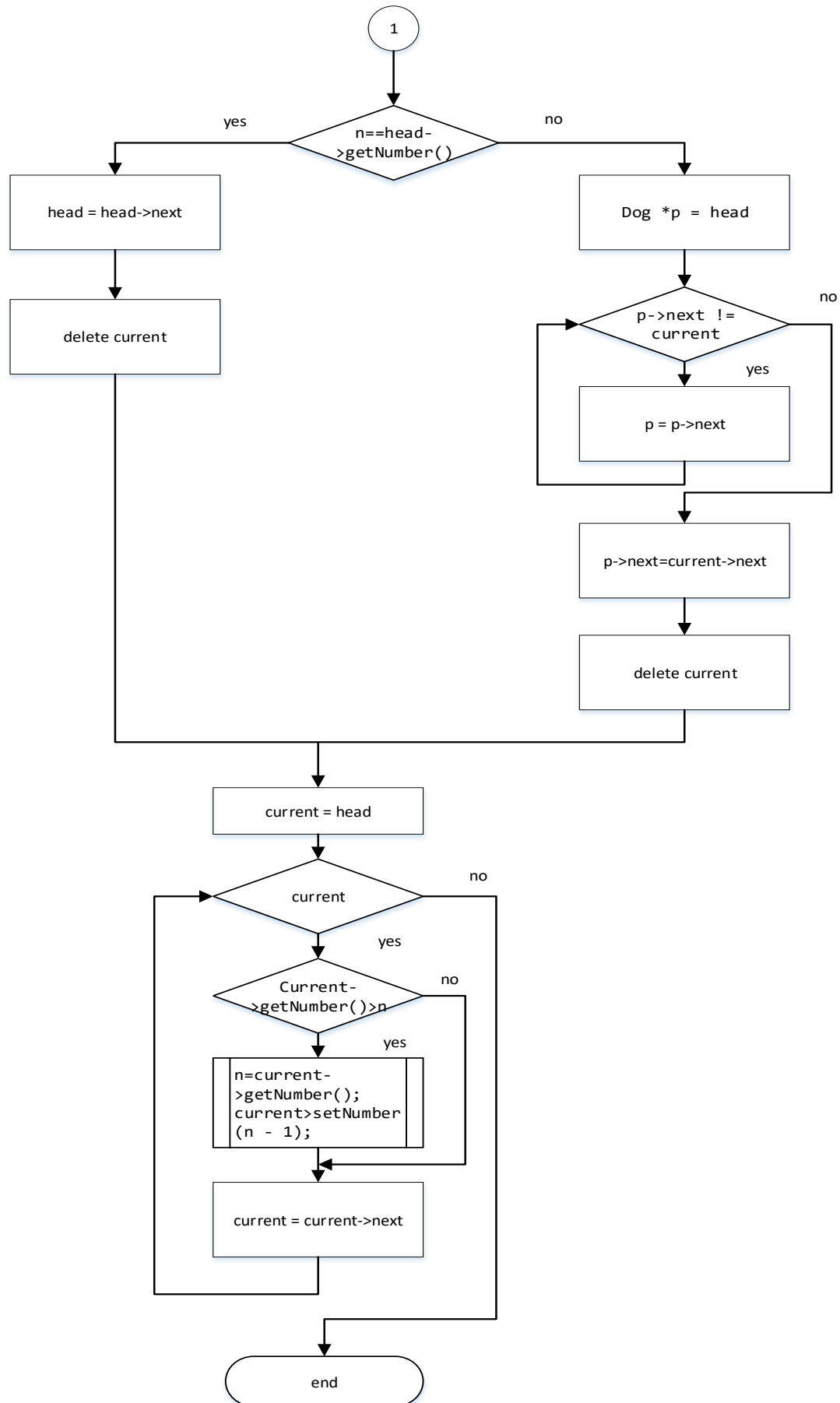


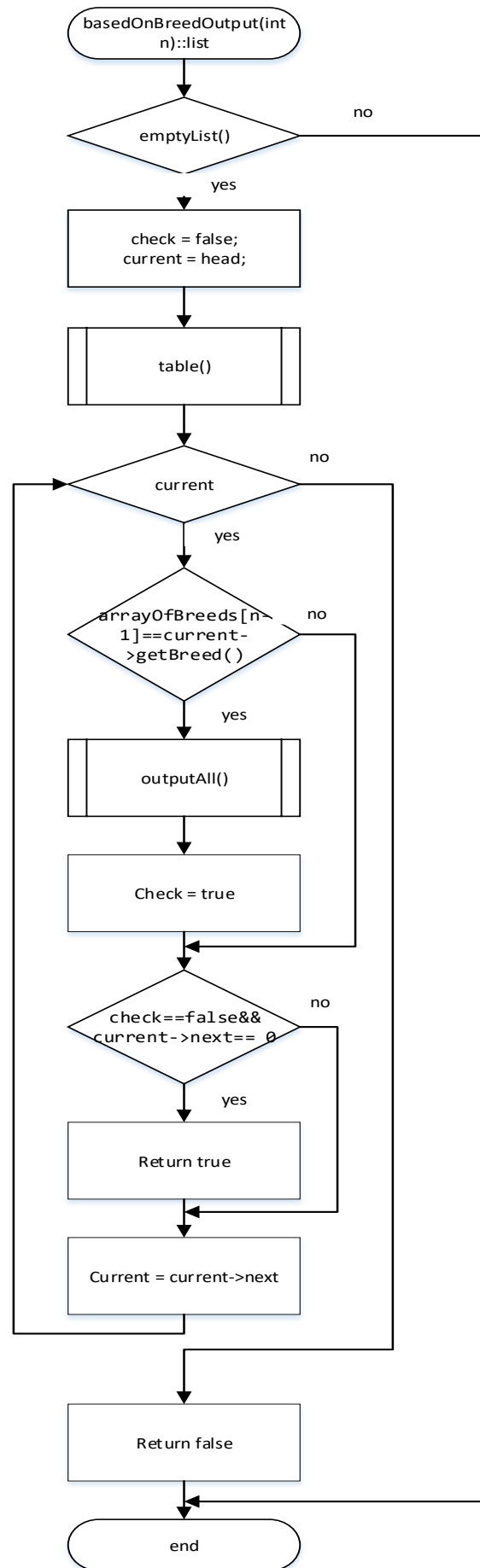
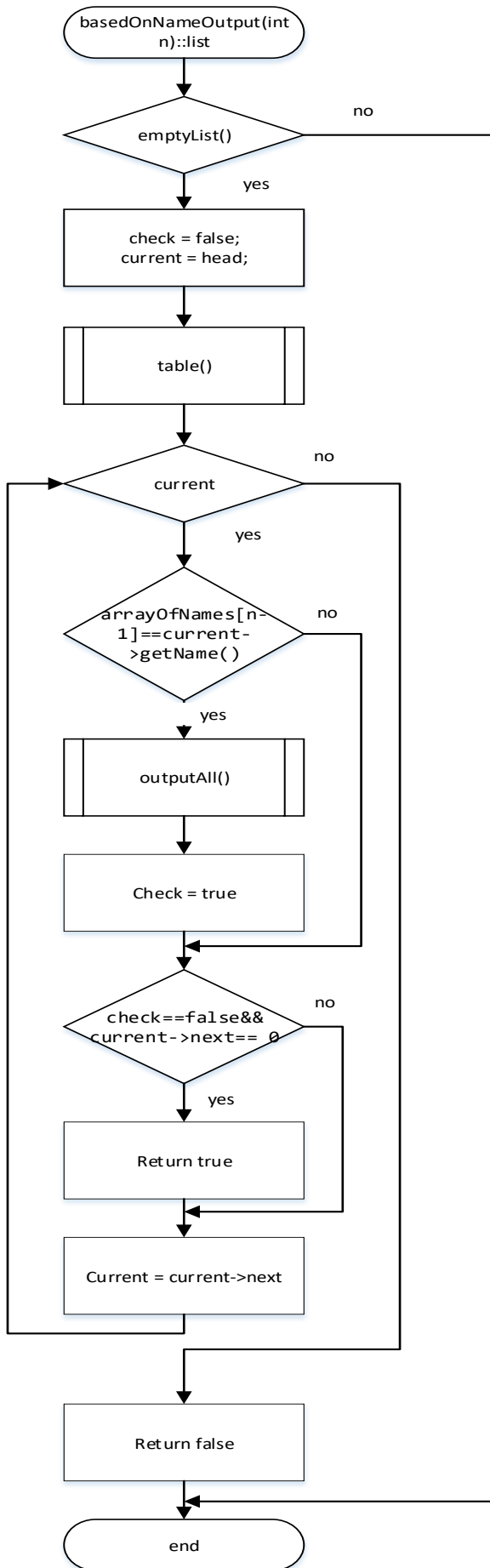


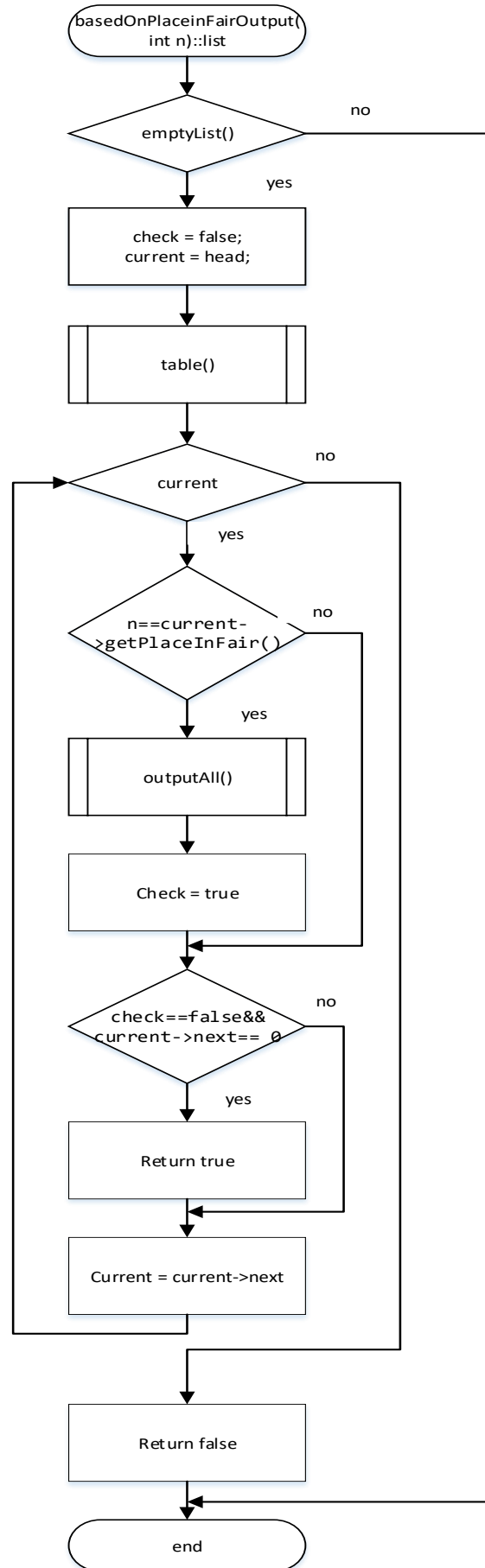
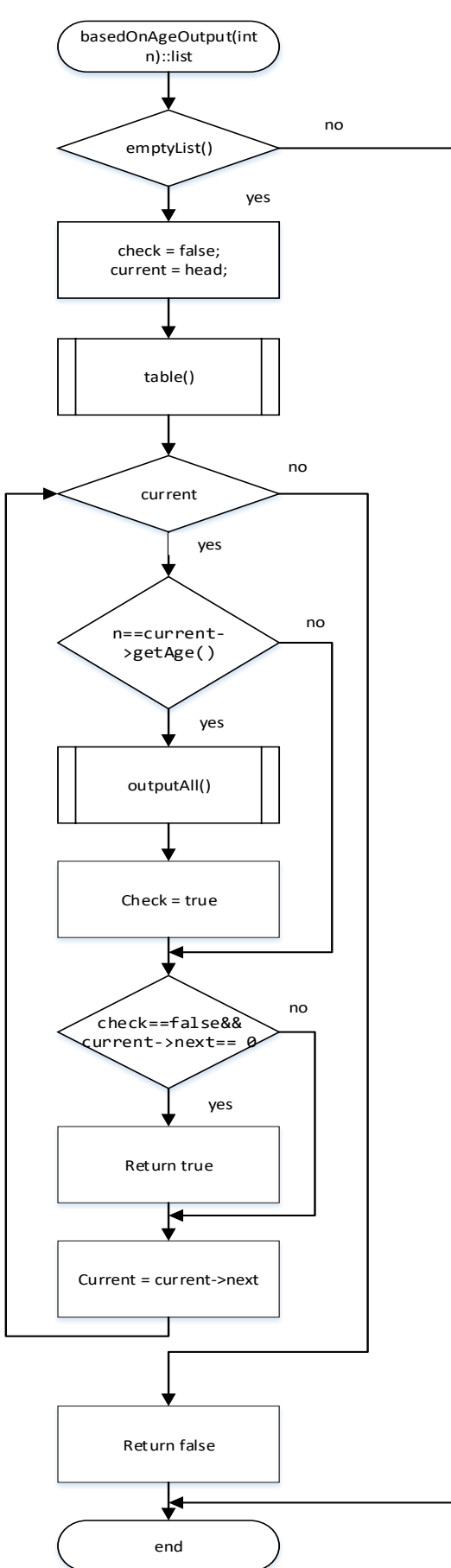


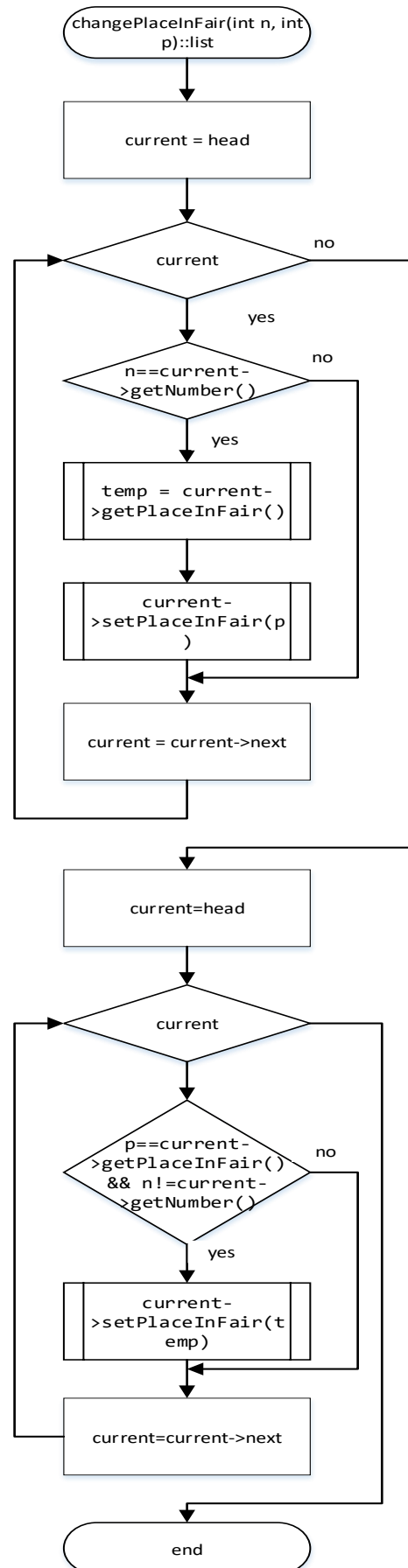
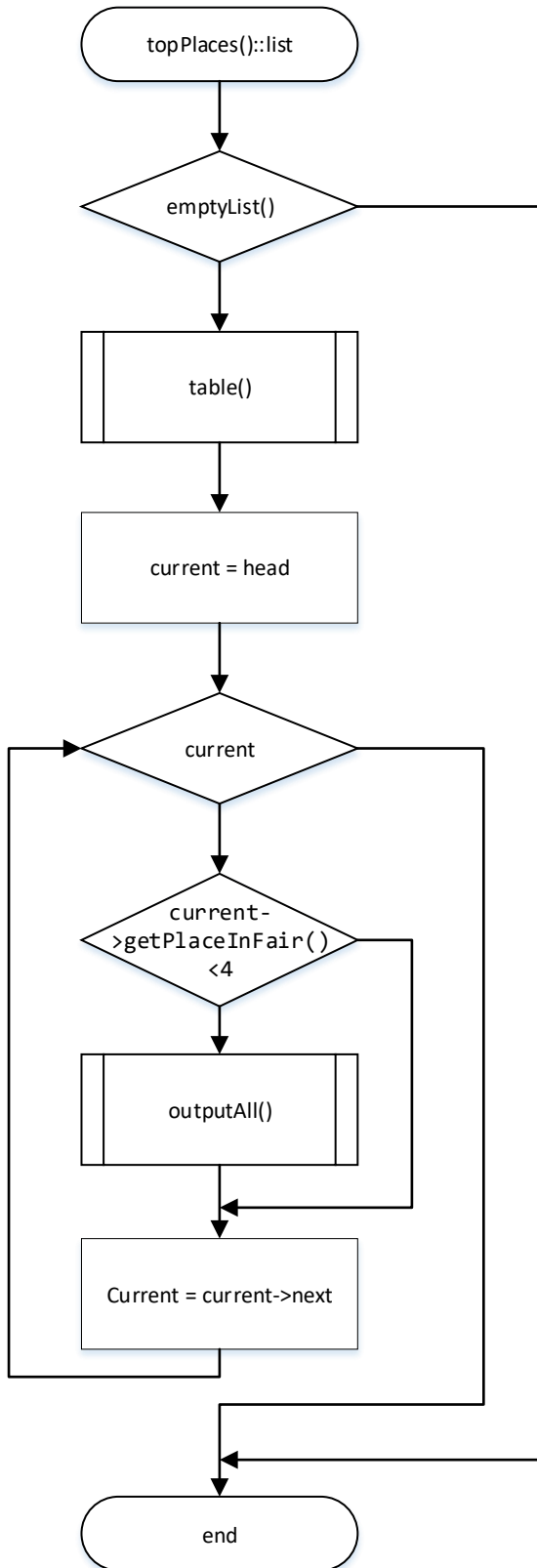


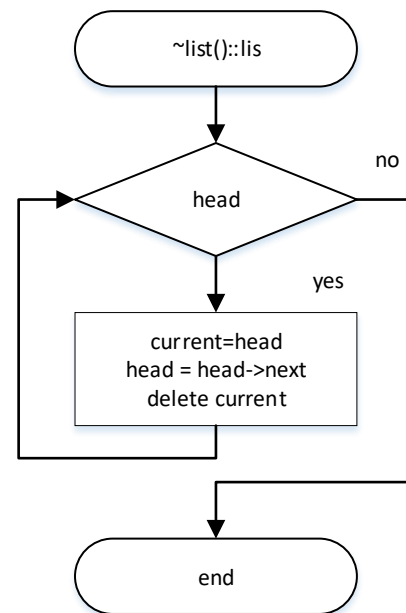
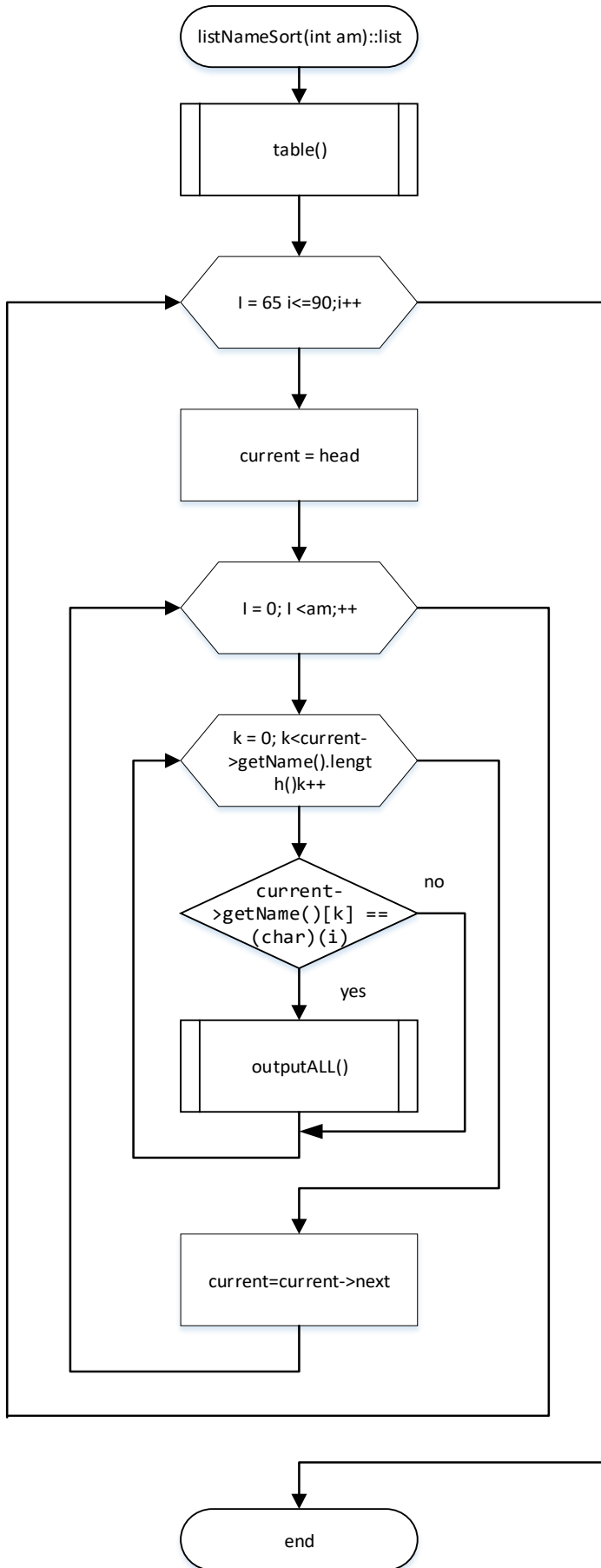


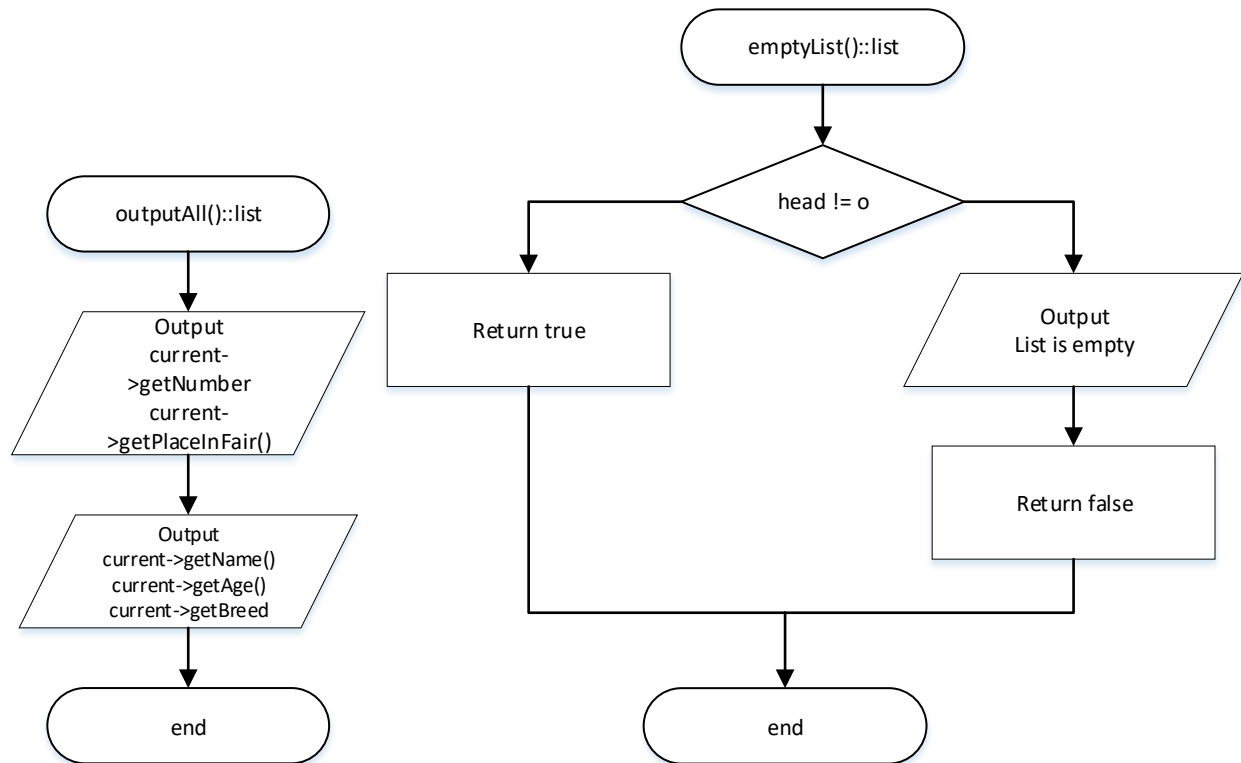


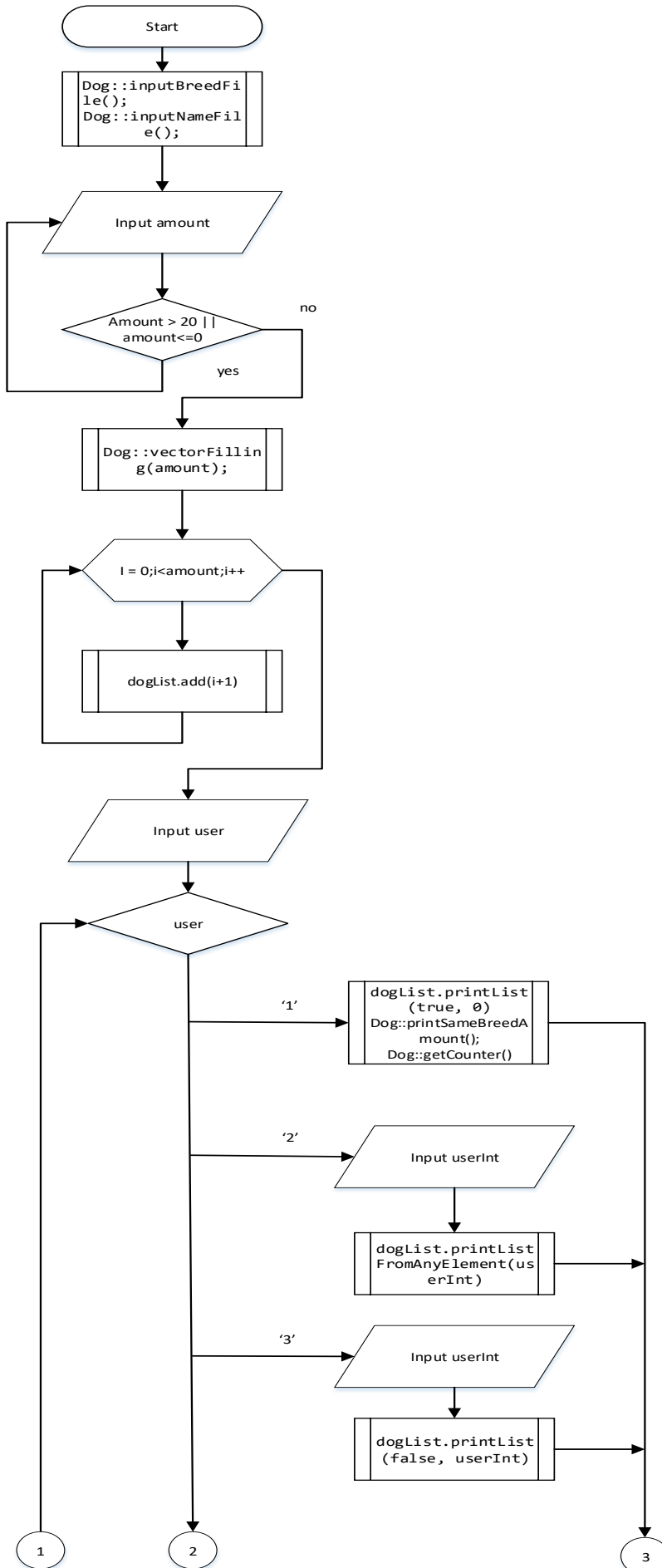


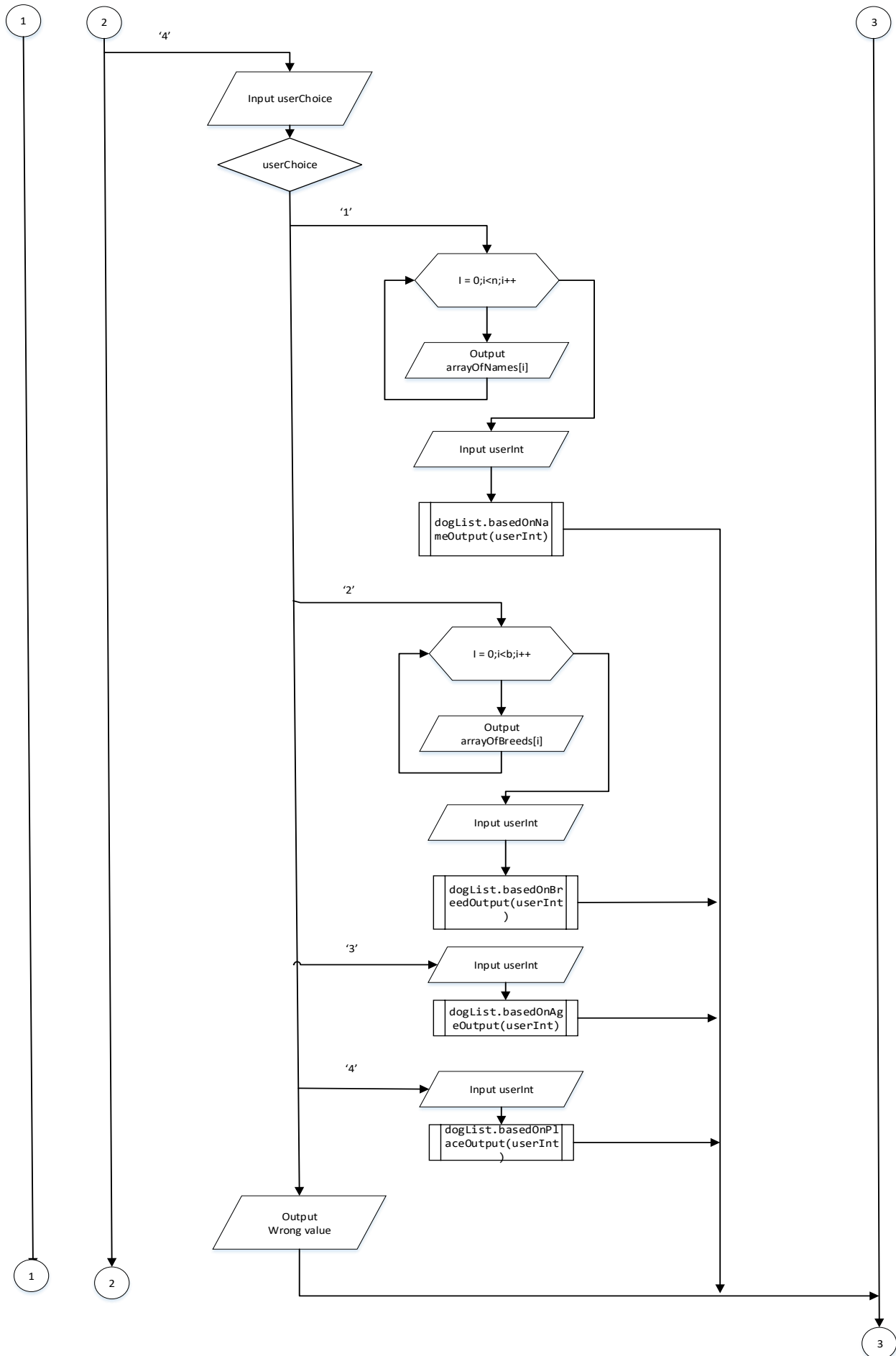


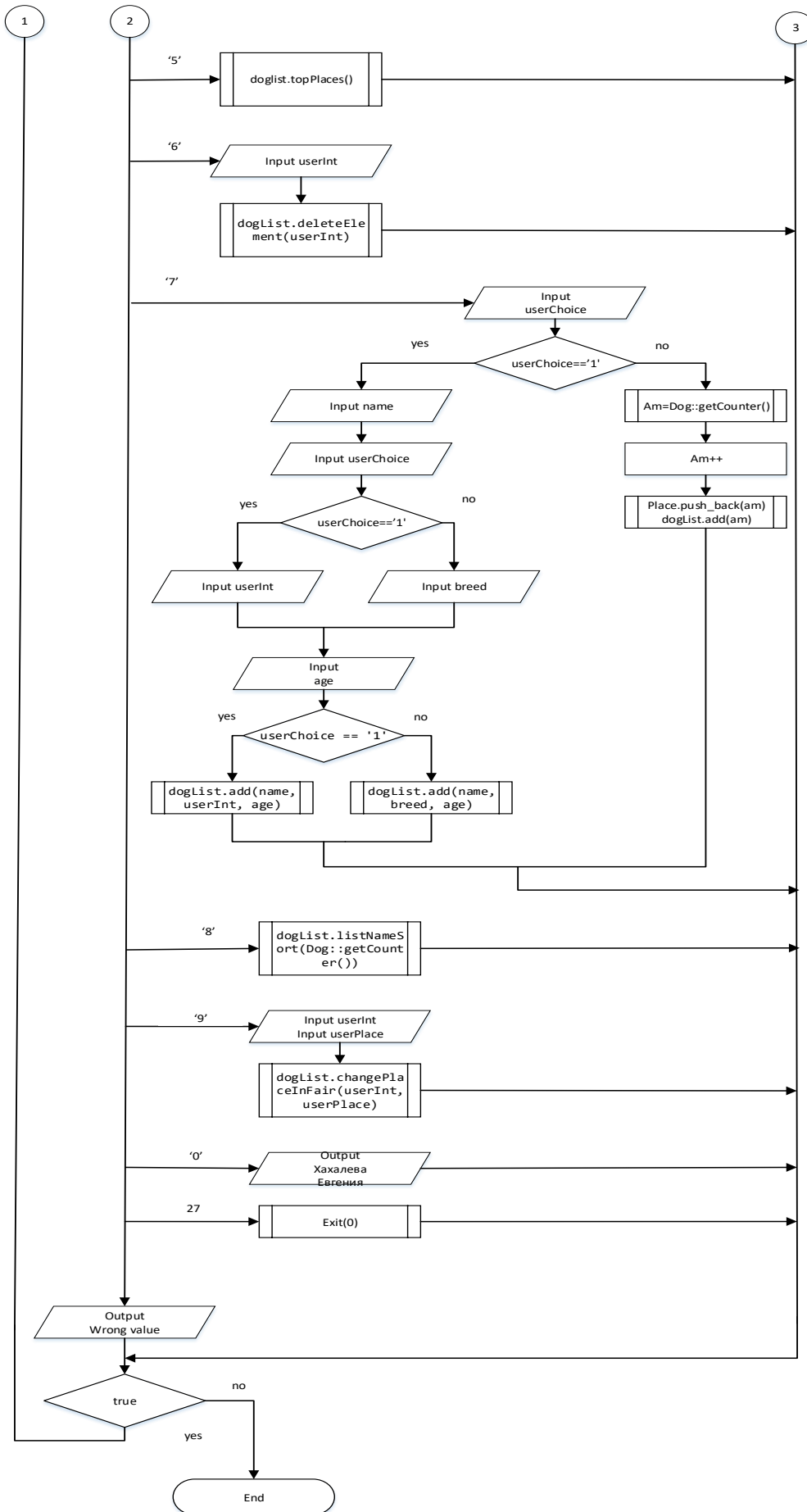












ДОДАТОК Б КОД ПРОГРАМИ

```

#include<iostream>
#include<string>
#include<ctime>
#include<fstream>
#include<vector>
#include<conio.h>
#include<iomanip>
#include<locale>
using namespace std;
const int n = 30;
const int b = 25;
string arrayOfNames[n], arrayOfBreeds[b];
vector<int>place;
class Dog {
    string breed, name;
    int number, age;
    int placeInFair;
    static int sameBreedAmount[b];
    static int counter;
public:
    Dog *next;
    Dog(int anumber, Dog *nxt) {
        age = rand() % 15 + 1;
        number = anumber;
        breed = arrayOfBreeds[rand()%b];
        name = arrayOfNames[rand()%n];
        compareBreeds(breed);
        placeInFair = randomPlaceInFair();
        next = nxt;
        counter++;
    }
    Dog( string n, string b, int a, Dog *nxt) {
        counter++;
        number = counter;
        name = n;
        breed = b;
        age = a;
        compareBreeds(breed);
        placeInFair = counter;
        next = nxt;
    }
    Dog(string n, int nb, int a, Dog *nxt) {
        counter++;
        number = counter;
        name = n;
        for (int i = 0; i < b; i++) {
            if (nb - 1 == i) {
                breed = arrayOfBreeds[i];
            }
        }
        age = a;
        compareBreeds(breed);
        placeInFair = counter;
        next = nxt;
    }
    Dog(){}
    ~Dog() {
        counter--;
    }
    void setNumber(int num) {

```

[illegible]


```

        fnames >> arrayOfNames[i];
    }
    fnames.close();
}

void Dog::printSameBreedAmount() {
    bool check = false;
    cout << endl << "Amount of identical breeds: " ;
    for (int i = 0; i < b; i++) {
        if (sameBreedAmount[i] > 1) {
            check = true;
            cout << endl << arrayOfBreeds[i] << " " << sameBreedAmount[i];
        }
    }
    if (check == false) {
        cout << "-" << endl;
    }
}

void Dog::compareBreeds(string abreed) {
    for (int i = 0; i < b; i++) {
        if (abreed == arrayOfBreeds[i]) {
            sameBreedAmount[i]++;
        }
    }
}

class list {
    Dog *head, *current;
public:
    list() {
        head = 0;
    }
    void add(int n) {
        Dog *p = new Dog(n, 0);
        if (head == 0) {
            head = p;
        }
        else {
            current = head;
            while (current->next != 0) {
                current = current->next;
            }
            current->next = p;
        }
    }
    void add(string name, string breed, int age) {
        Dog *p = new Dog(name, breed, age, 0);

        if (head == 0) {
            head = p;
        }
        else {
            current = head;
            while (current->next != 0) {
                current = current->next;
            }
            current->next = p;
        }
    }
    void add(string name, int numBreed, int age) {
        Dog *p = new Dog(name, numBreed, age, 0);

        if (head == 0) {
            head = p;
        }
        else {
            current = head;

```

```

        while (current->next != 0) {
            current = current->next;
        }
        current->next = p;
    }
}

bool printList(bool check, int userN) {
    if (emptyList()) {
        if (check) {
            current = head;
            table();
            while (current) {
                outputAll();
                current = current->next;
            }
        }
        else {
            current = head;
            while (current) {
                if (userN == current->getNumber()) {
                    table();
                    outputAll();
                    return true;
                    break;
                }
                else {
                    current = current->next;
                }
            }
            return false;
        }
    }
    else {}
}

void printListFromAnyElement(int numOfel) {
    bool check;
    if (emptyList()) {
        check = false;
        table();
        for (int i = 0; i < 2; i++) {
            current = head;
            if (numOfel < current->getNumber()) {
                cout << "No such element!" << endl;
                break;
            }
            while (current) {
                if (i == 0 && numOfel <= current->getNumber()) {
                    outputAll();
                    check = true;
                }
                if (i == 1 && numOfel > current->getNumber() && check == true) {
                    outputAll();
                }
                current = current->next;
            }
            if (check == false) {
                cout << "No such element!" << endl;
                break;
            }
        }
    }
}

bool basedOnNameOutput(int numname) {

```

```

    bool check;
    if (emptyList()) {
        check = false;
        current = head;
        table();
        while (current) {
            if (arrayOfNames[numname-1] == current->getName()) {
                outputAll();
                check = true;
            }
            if (check == false && current->next == 0) {
                return true;
            }
            current = current->next;
        }
        return false;
    }
}

bool basedOnBreedOutput(int numbreed) {
    bool check;
    if (emptyList()) {
        check = false;
        current = head;
        table();
        while (current) {
            if (arrayOfBreeds[numbreed-1] == current->getBreed()) {
                outputAll();
                check = true;
            }
            if (check == false && current->next == 0) {
                return true;
            }
            current = current->next;
        }
        return false;
    }
}

bool basedOnAgeOutput(int age) {
    bool check;
    if (emptyList()) {
        check = false;
        current = head;
        table();
        while (current) {
            if (age == current->getAge()) {
                outputAll();
                check = true;
            }
            if (check == false && current->next == 0) {
                return true;
            }
            current = current->next;
        }
        return false;
    }
}

bool basedOnPlaceOutput(int place) {
    bool check;
    if (emptyList()) {
        check = false;
        current = head;
        while (current) {
            if (place == current->getPlaceInFair()) {
                table();
            }
            current = current->next;
        }
    }
}

```

```

        outputAll();
        check = true;
    }
    if (check == false && current->next == 0) {
        return true;
    }
    current = current->next;
}
return false;
}
}
void topPlaces() {
    if (emptyList()) {
        table();
        current = head;
        while (current){
            if (current->getPlaceInFair() < 4) {
                outputAll();
            }
            current = current->next;
        }
    }
}
void deleteElement(int numOfelement) {
    current = head;
    while (current) {
        if (numOfelement == current->getNumber()) {
            Dog::minusSameBreedAmount(current->getBreed());
        }
        current = current->next;
    }
    int n;
    current = head;
    for (int i = 1; i < numOfelement && current != NULL; i++) {
        current = current->next;
    }
    if (current != NULL && numOfelement > 0) {
        if (numOfelement == head->getNumber()) {
            head = head->next;
            delete current;
        }
        else {
            Dog *p = head;
            while (p->next != current) {
                p = p->next;
            }
            p->next = current->next;
            delete current;
        }
        current = head;
        while (current) {
            if (current->getNumber() > numOfelement) {
                n = current->getNumber();
                current->setNumber(n - 1);
            }
            current = current->next;
        }
    }
    else {
        cout << "No such element!" << endl;
    }
}
}
void changePlaceInFair(int numOfel, int newPlace) {
    int temp, temp2;
    current = head;

```

```

        while (current) {
            if (numOfel == current->getNumber()) {
                temp = current->getPlaceInFair();
                current->setPlaceInFair(newPlace);
            }
            current = current->next;
        }
        current = head;
        while (current) {
            if (newPlace == current->getPlaceInFair() && numOfel != current->getNumber()) {
                current->setPlaceInFair(temp);
            }
            current = current->next;
        }
    }
    void listNameSort(int am) {
        table();
        for (int i = 65; i <= 90; i++) {
            current = head;
            for (int j = 0; j < am; j++) {
                for (int k = 0; k < current->getName().length(); k++) {
                    if (current->getName()[k] == (char)(i)) {
                        outputAll();
                    }
                }
                current = current->next;
            }
        }
    }
    ~list() {
        while (head) {
            current = head;
            head = head->next;
            delete current;
        }
    }
private:
    void outputAll() {
        cout << current->getNumber() << setw(15);
        cout << current->getPlaceInFair() << setw(10) << '\t';
        cout << current->getName() << "\t\t";
        cout << current->getAge() << '\t';
        cout << current->getBreed() << endl;
    }
    bool emptyList() {
        if (head != 0) {
            return true;
        }
        else {
            cout << "List is empty!" << endl;
            return false;
        }
    }
    void table() {
        cout << "No.\t" << "Place in fair\t\t" << "Name\t\t" << "Age\t " << "Breed\t" << endl;
        cout << "===== " << endl;
    }
};

void main() {
    srand(time(0));
    setlocale(LC_ALL, "Russian");
    Dog::inputBreedFile();
    Dog::inputNameFile();
}

```

```

bool check1, check2;
int amount, userInt, age, userPlace, user, am;
char userChoice;
string userStr, name, breed;
do {
    cout << "Please, input amount of elements in list (not more than 20): ";
    cin >> amount;
    if (amount > 20 || amount <= 0) {
        system("cls");
        cout << "Wrong amount! Try again..." << endl;
    }
} while (amount > 20 || amount <= 0);
Dog::vectorFilling(amount);
list dogList;
for (int i = 0; i < amount; i++) {
    dogList.add(i + 1);
}
do{
    system("cls");
    cout << "1-list output;" << endl << "2-list output starting from any element;" << endl << "3-
output certain element;" << endl << "4-based output;" << endl << "5-top places;" << endl;
    cout << "6-delete element;" << endl << "7-add element;" << endl << "8-alphabetical output;" <<
endl << "9-change place in fair;" << endl << "0-information about author;" << endl << "esc-exit" << endl;
    cout << "Please, input your choice: ";
    user = _getch();
    cout << endl;
    switch (user) {
        case '1':
            system("cls");
            dogList.printList(true, 0);
            Dog::printSameBreedAmount();
            cout << endl;
            cout << "Amount of instances: ";
            cout << Dog::getCounter() << endl;
            break;
        case '2':
            system("cls");
            cout << "Input number of element that will start the list: ";
            cin >> userInt;
            dogList.printListFromAnyElement(userInt);
            break;
        case '3':
            system("cls");
            cout << "Input number of element that you want to output: ";
            cin >> userInt;
            if (dogList.printList(false, userInt)) {}
            else {
                cout << "No such element!" << endl;
            }
            break;
        case '4':
            do{
                system("cls");
                cout << "Please, input on what output will based:" << endl << "1-name; 2-breed;
3-age; 4-place in fair;" << endl;
                cin >> userChoice;

                switch (userChoice) {
                    case '1':
                        system("cls");
                        for (int i = 0; i < n; i++) {
                            cout << i + 1 << '.' << " " << arrayOfNames[i] << endl;
                        }
                        cout << endl << "Input number of name: ";
                        cin >> userInt;

```

```

        if (dogList.basedOnNameOutput(userInt)) {
            cout << "No such name!" << endl;
        }
        break;
    case '2':
        system("cls");
        for (int i = 0; i < b; i++) {
            cout << i + 1 << '.' << " " << arrayOfBreeds[i] << endl;
        }
        cout << "Input breed: ";
        cin >> userInt;
        if (dogList.basedOnBreedOutput(userInt)) {
            cout << "No such breed!" << endl;
        }
        break;
    case '3':
        system("cls");
        cout << "Input age: ";
        cin >> userInt;
        if (dogList.basedOnAgeOutput(userInt)) {
            cout << "No such age!" << endl;
        }
        break;
    case '4':
        system("cls");
        cout << "Input place in fair: ";
        cin >> userInt;
        if (dogList.basedOnPlaceOutput(userInt)) {
            cout << "No such place in fair!" << endl;
        }
        break;
    default:
        cout << "You input wrong value!" << endl;
        system("pause");
        break;
    }

    } while (userChoice != '1' && userChoice != '2' && userChoice != '3' && userChoice !=
'4');

    break;
case '5':
    system("cls");
    cout << setw(43) << "Top places:" << endl;
    dogList.topPlaces();
    break;
case '6':
    cout << "Input number of element that you want to delete: ";
    cin >> userInt;
    dogList.deleteElement(userInt);
    break;
case '7':
    system("cls");
    do {
        cout << "1-input data;" << endl << "2-add automatically;" << endl;
        cin >> userChoice;
        if (userChoice != '1' && userChoice != '2') {
            cout << "You have to input 1 or 2! Try again..." << endl;
        }
    } while (userChoice != '1' && userChoice != '2');

    if (userChoice == '1') {
        system("cls");
        cout << "Input name: ";
        cin >> name;
    }

```

```

input;" << endl;

        cout << endl << "Do you want to choose breed or input yourself? 1-choose; 2-
do {
    cin >> userChoice;
    if (userChoice != '1' && userChoice != '2') {
        cout << "You have to input 1 or 2! Try again..." << endl;
    }
} while (userChoice != '1' && userChoice != '2');
if (userChoice == '1') {
    system("cls");
    for (int i = 0; i < b; i++) {
        cout << i + 1 << '.' << " " << arrayOfBreeds[i] << endl;
    }
    cout << "Input number of breed that you want to add: ";
    cin >> userInt;
}
else {
    cout << "Input breed: ";
    cin >> breed;
}
do {
    cout << endl << "Input age: ";
    cin >> age;
    if (age > 20 || age <= 0) {
        cout << "Error... Such age does not exist!" << endl;
    }
} while (age > 20 || age <= 0);
if (userChoice == '1') {
    dogList.add(name, userInt, age);
}
else {
    dogList.add(name, breed, age);
}
}
else {
    am = Dog::getCounter();
    am++;
    place.push_back(am);
    dogList.add(am);
}
break;
case '8':
    dogList.listNameSort(Dog::getCounter());
    break;
case '9':
    system("cls");
    check1 = false;
    check2 = false;
    cout << "Input number of element that you want to change: ";
    cin >> userInt;
    cout << "New place in fair: ";
    cin >> userPlace;
    for (int i = 1; i <= Dog::getCounter(); i++) {
        if (userInt == i) {
            check1 = true;
        }
    }
    for (int i = 1; i <= Dog::getCounter(); i++) {
        if (userPlace == i) {
            check2 = true;
        }
    }
    if (check1 == true && check2 == true) {
        dogList.changePlaceInFair(userInt, userPlace);
    }
}

```



```
        else {
            cout << "You input wrong number or place in fair!" << endl;
        }
        break;
    case '0':
        cout << endl << "Хахалева Евгения ОПК-217" << endl;
        break;
    case 27:
        exit(0);
        break;
    default:
        cout << "You input wrong value!" << endl;
    }
    system("pause");
} while (true);
}
```

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ КОМП'ЮТЕРНО-ТЕХНОЛОГІЧНИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Спеціальність: 5.05010101 "Обслуговування програмних систем і
комплексів"

ІЛЮСТРАТИВНІ МАТЕРІАЛИ
ДО КУРСОВОГО ПРОЕКТУ
на тему: "Розробка прикладної програми з використанням класу
«СОБАКА»"

Виконав: _____ Студентка групи ОПК-217
Хахалєва Є.В.

Перевірив: _____ Викладач ХКТК НТУ "ХПІ"
Коломієць П.Ю.

м. Харків 2019 р.

Курсовий проект: Клас «Собака»

Виконала
Студентка гр. ОПК-217
Хахалєва Є.В.



Використана
середина розробки



Microsoft®

Visual Studio®

Постановка завдання

Створити клас
«Собака» з кількістю
примірників обраною
користувачем.
Створити список з
відповідною кількістю
елементів. А також
меню для
користування
програмою.



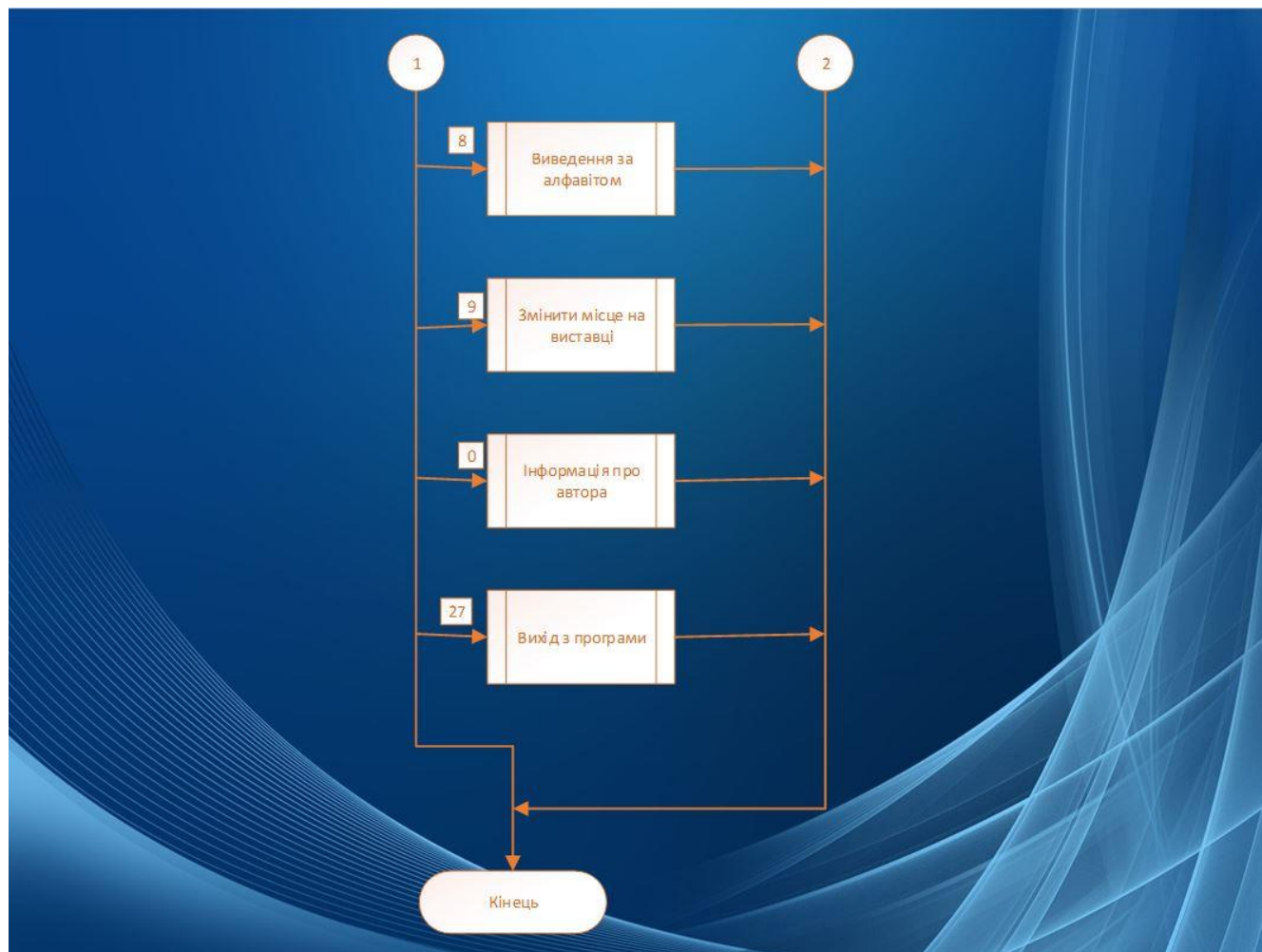
Детальний опис завдання

Список повинен виводитися:

- Повністю за порядком
- По колу починаючи з будь-якого елемента
- За алфавітом
- За ознакою, яку введе користувач (порода, рік, тощо)

Можливості програми:

- Додавання елемента
- Видалення елемента
- Зміна зайнятого собакою місця на виставці
- Виведення елемента з визначеним порядковим номером
- Виведення призових місць



Алгоритм вирішення поставленої задачі

- Створити клас «Собака» та необхідні поля і методи;
- Створити список примірників класу;
- Створити меню для роботи з примірниками класу;

Керівництво користувача

```
C:\Users\sbuluga\source\repos\CourseWork\Debug\project.exe
Please, input amount of elements in list <not more than 20>:
```

кількість елементів

```
C:\Users\sbuluga\source\repos\CourseWork\Debug\project.exe
1-list output;
2-list output starting from any element;
3-output certain element;
4-based output;
5-top places;
6-delete element;
7-add element;
8-alphabetical output;
9-change place in fair;
0-information about author;
esc-exit
Please, input your choice:
```

меню користувача

```
C:\Users\sbuluga\source\repos\CourseWork\Debug\project.exe
```

No.	Place in fair	Name	Age	Breed
1	5	Donald	15	Rottweiler
2	1	Sophie	1	BullTerrier
3	2	Max	15	Husky
4	4	Sunny	7	Dachshund
5	3	Molly	4	Shepherd

```

Amount of identical breeds: -
Amount of instances: 5
Для продолжения нажмите любую клавишу . . .
```

виведення всіх елементів

