Архитектура вычислительных систем.

Задание 2. Вариант 12. Функция 19.

Выполнила: Словягина Анна Михайловна

Студентка 2-го курса программной инженерии

группа БПИ207

1 Описание задания

* Обобщенный артефакт, используемый в задании:
  + животные.
* Базовые альтернативы и их уникальные параметры:
  + рыбы:

место проживания - перечислимый тип (река, озеро, океан, море)

* + птицы:

отношение к перелету - булевская величина (перелетные, не перелетные)

* + звери:

тип зверя - перечислимый тип (травоядные, хищники, насекомоядные)

* Общие для всех альтернатив переменные:
  + Название - строка символов;
  + Вес в граммах - целое число.
* Общие для всех альтернатив функции:
  + Частное от деления суммы кодов названия животного на вес - действительное число.
* Функция - удаление из контейнера тех элементов, для которых значение, полученное с использованием функции, общей для всех альтернатив, меньше, чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции.

2 Структурная схема изучаемой архитектуры ВС

Таблица типов

|  |  |
| --- | --- |
| int | 4 |
| double | 8 |
| char | 1 |
| bool | 1 |
| string | 32 |
| Class Container  int: len  Animals\*: storage[len] | 80008  4[0]  8\*10000[4] = 80000[4] |
| Class Animals  string: name  int: weight | 40  32[0]  4[32] |
| Class Fish  enum habit: h | 8  4[0] |
| Class Bird  bool: migratory | 8  1[0] |
| Class Beast  enum type: t | 8  4[0] |

Память программы

|  |  |
| --- | --- |
| main(int argc, char\* argv[])  int: argc  char\*: argv:  Class Container: c  ifstream: ifst  auto: size  ifstream: ofst1  ifstream: ofst2 | 4[0]  1[4]  80008[5]  520[80013]  4[80533]  512[80537]  512[81049] |
| void Container::Delete()  Animals\* : an  int : i | 8[0]  4[8] |
| double Container::Division()  double : sum  int : i | 8[0]  4[8] |
| void Bird::InRnd()  int: name\_length  char: symbol  int: i | 4[0]  1[4]  4[5] |

3 Стек



4 Основные характеристики программы

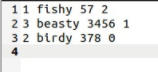
Число интерфейсных модулей: 5

Число модулей реализации: 6

Общий размер исходных текстов: 543

Формат ввода:

В каждой строке файла должно содержаться 4 переменные, разделенные пробелом:



Первая переменная отвечает за вид животного (1 – рыба, 2 – птица, 3 – зверь), вторая переменная – за название животного, третья переменная – за его вес в граммах, четвертая переменная – за уникальные параметры.

Уникальные параметры для рыбы – это место обитания:

1 – река

2 – озеро

3 – океан

4 – море

Уникальный параметр для птицы – переменная, которая отвечает за то, является ли птица перелетной

0 – нет

1 – да

Уникальные параметры для зверя – это тип животного

1 – травоядный

2 – хищный

3 – насекомоядный

Замечание:

При некорректном вводе типа животного, программа не добавляет данный элемент в контейнер.

При некорректном вводе места обитания рыбы, программа выставляет значение по умолчанию — река.

Вводя переменную отвечающую, является ли птица перелетной следует помнить, что 0 — птица не перелетная, любое другое число — перелетная.

При некорректном вводе типа зверя, программа выставляет значение по умолчанию — травоядный.

Расположение входных и выходных данных

Входные данные лежат в:

HomeWork\_1/cmake-build-debug/tests

Выходные для первого задания:

HomeWork\_1/cmake-build-debug/results1

Выходные для второго задания:

HomeWork\_1/cmake-build-debug/results2

5 Время выполнения программы для различных тестовых наборов данных

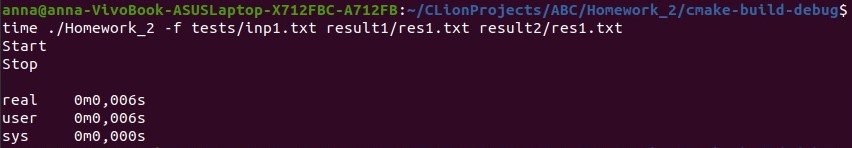
Ввод из файла:

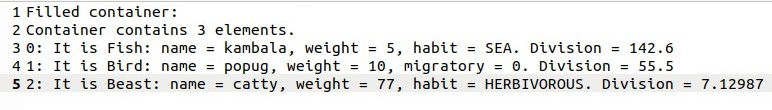
1 тест:

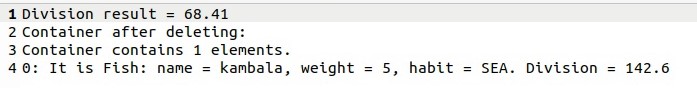
Ввод:



Запуск программы:

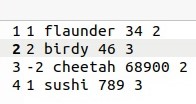


Результат:

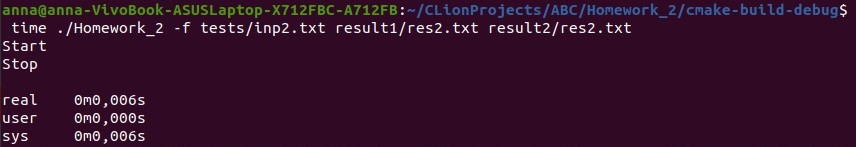


2 тест:

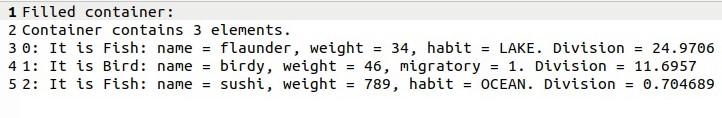
Ввод:

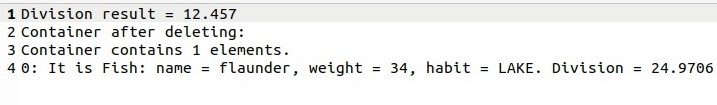


Запуск программы:



Результат:

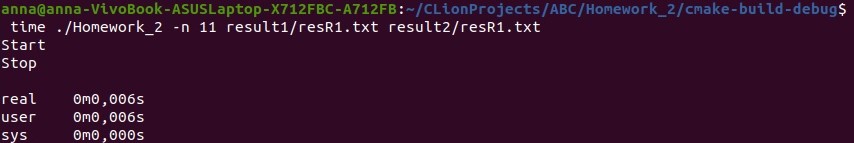




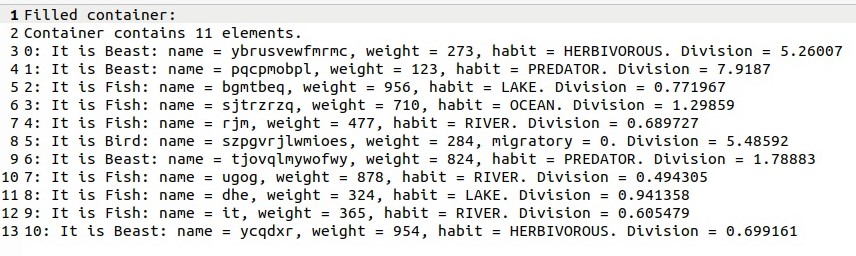
Рандомный ввод:

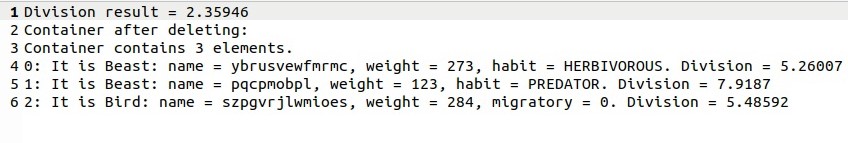
3 тест:

Запуск программы:



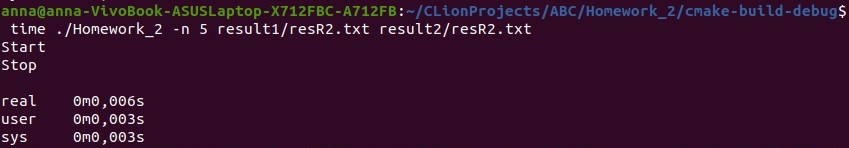
Результат:



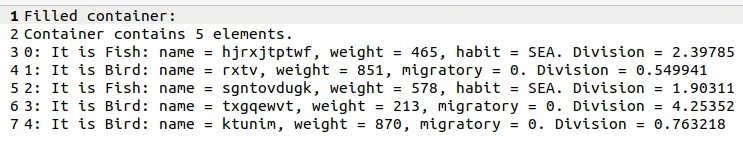


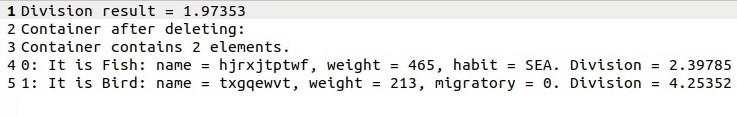
4 тест:

Запуск программы:



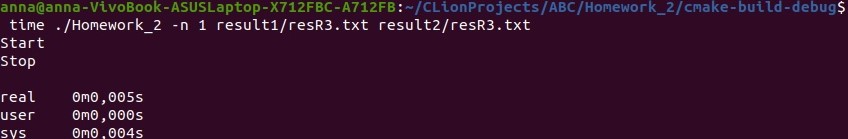
Результат:



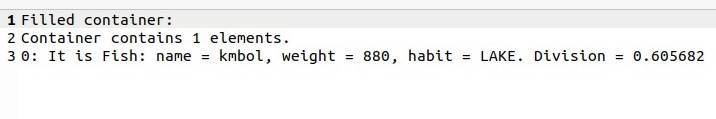


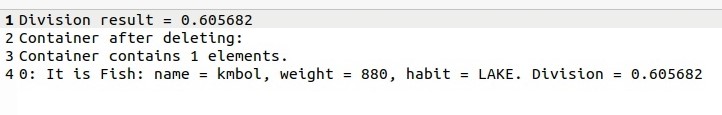
5 тест:

Запуск программы:



Результат:





6 Сравнительный анализ

Основные характеристики данной программы:

Число интерфейсных модулей: 5

Число модулей реализации: 6

Общий размер исходных текстов: 543

Основные характеристики предыдущей программы:

Число интерфейсных модулей: 5  
Число модулей реализации: 6  
Общий размер исходных текстов: 589

Сравнение времени исполнения программ:

|  |  |  |
| --- | --- | --- |
| № | Данная программа | Предыдущая программа |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| 5. |  |  |

Исходя из полученных данных, можно сделать вывод, что программа, написанная с использованием парадигмы объектно-ориентированного программирования превосходит по времени исполнения программу, написанную с помощью процедурного подхода.  
Однако, главным отличием является подход написания программы. Предыдущая программа написана с помощью процедурного подхода, именно поэтому все базовые альтернативы были объединены в обобщенную структуру и все взаимодействие с ними проходило именно через обобщенный артефакт, используемый в задании. В программе, написанной с помощью ООП подхода все базовые альтернативы объединял абстрактный класс, в котором содержались общие поля для каждого класса наследника, а также виртуальные методы, переопределенные в каждом наследнике. Взаимодействие между объектами в данной программе происходило “напрямую”, а не через обобщенную структуру. Это и является главным преимуществом ООП подхода по сравнению с процедурным.