Сибирцева Анна
БПИ206

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Вариант: **222**
Задание: **12**
Функция: **16**

Описание задачи:

Необходимо реализовать программу, в которой базовым классом является животное, а его альтернативы (наследники): рыбы (поле – место проживания – перечислимый тип: река, озеро, море, пруд, океан), птицы(параметр – отношение к перелёту – перелётные или остающиеся на зимовку), звери(поле – тип питания – хищники, травоядные, насекомоядные, всеядные).

Общим для все классов являются: имя животного/его название – строка символов, вес животного в граммах – целое число.

Общей для всех классов функцией является нахождение частного от деления суммы кодов незашифрованной строки на вес, являющиеся действительным числом.

Обработка данных является их упорядочивание по убыванию, с использованием сортировки методом деления по пополам (*Binary Search*).

Основные характеристики программы:

Число интерфейсных модулей: **5**
Число модулей реализация: **6**
Общий размер исходных текстов: **~15,05 Кб**
Размер исполняемого файла: **124 КБ**
Время работы на тестах:

TEST 1: **0.003 с**
TEST 2: **0.007 с**
TEST 3: **0.001 с** *(*программа завершилась преждевременно из-за некорректных входных данных)*
TEST 4: **0.004 с**
TEST 5: **0.001 с** *(*программа завершилась преждевременно из-за некорректных входных данных)*
TEST 6: **1.880 с**
TEST 7: **0.006 с**
TEST 8: **0.014 с**

* Шестой тест был заменен для данной и последующих реализаций, т.к. ранее он был бесполезен. Теперь данный тест будет проверять умение программы быстро генерировать большое количество случайных значений. В предыдущей реализации данный он выполнялся **0.561 с**

TYPE TABLE	
INT	4
DOUBLE	8
STD::STRING	32
CHAR[26]	26
STD::VECTOR<STD::UNIQUE_PTR<ANIMAL>>	24
enum Nutrition: **CARNIVORES,** **HERBIVORES,** **INSECTIVORES,** **OMNIVORES**	4[0]
enum Habitat: **RIVER,** **SEA,** **LAKE,** **OCEAN,** **POND**	4[0]
class Beast: **nutrition_: int**	**4** 4[0]
class Bird: **is_migratory_: int**	**4** 4[0]
class Fish: **habitat_: int**	**4** 4[0]
class Animal: **name_: std::string** **weight_: int**	**36** 32[0] 4[32]

* В таблице не присутствует класс “Commander” так как в нем нет переменных или полей.

PROGRAM MEMORY	
main (int argc, char* argv[]): **argv: int** **argc: char**** **size: int** **zoo: std::vector<std::unique_ptr<Animal>>** **commander: Commander** **input: ifstream** **general_output: ofstream** **sorted_output: ofstream**	**1604** 4[0] 8[4] 24[12] x[36] 520[] 512[] 512[]
double Animal::Quotient() const: **sum: double** **i: char**	**9** 8[0] 1[8]
Commander::InputRandom(std::vector<std::unique_ptr<Animal>> &zoo, int size): **i: int** **animal_tupe: int**	**8** 4[0] 4[4]
Commander::InputFromFile(std::vector<std::unique_ptr<Animal>> &zoo, std::ifstream &input, int size) **i: int** **weight: int** **animal_type: int** **additional: int** **name: std::string**	**44** 4[0] 4[4] 4[8] 4[12] 32[12]
std::string RandomName(const char letters[26]) **name_length: int** **name: std::string**	**36** 4[0] 32[4]

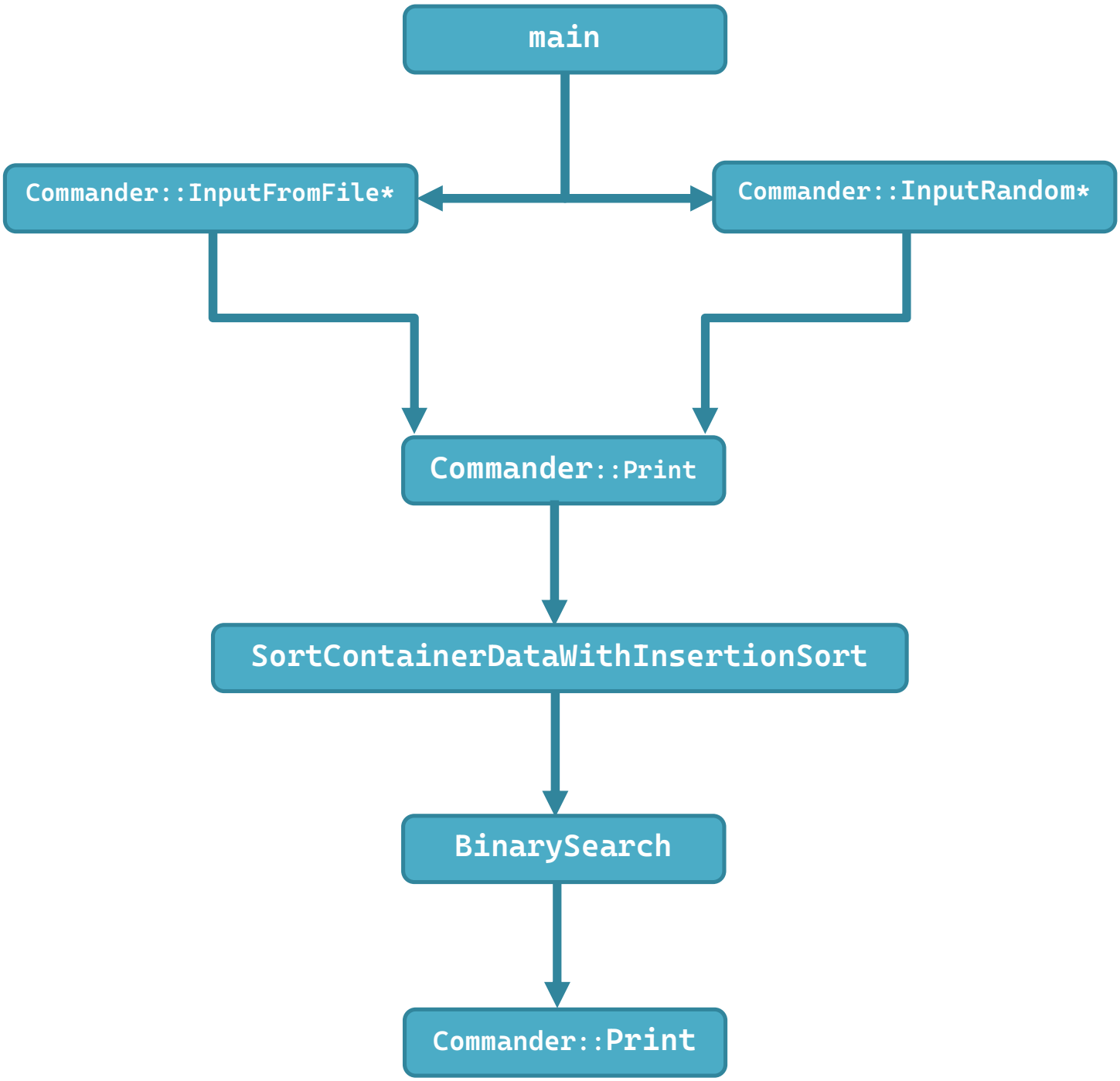
Дополнительная информация:

Программа была выполнена с использованием объектно-ориентированного подхода и статической типизации. Разработка осуществлялась на языке программирования C++.

Подробная информация о корректном использовании находится в файле README.txt

Ниже представлен ПРИБЛИЗИТЕЛЬНЫЙ стек вызовов:

Звездочками помечены методы, которые вызываются более одного раза в зависимости от параметров.



Сравнение программ:

Это вторая программа выполненная по данному заданию. В этой версии, вместо использования C++ в стиле C все было написано на «чистом» C++. Суть данного задания была в создании программы, используя ООП и статической

типизации, что и было выполнено. Так же были модифицированы некоторые фрагменты предыдущего кода, такие как:

1. Замена класса «Контейнер» на стандартный вектор.
2. Массивы, хранящие в себе char были заменены на строки.
3. Структуры были заменены на классы для упрощения дальнейшей работы.
4. Методы генерации случайных животных или чтения их параметров из файла теперь не являются отдельными методами для каждых альтернатив. Они были вынесены в новый класс «Командер» и выполняются аналогично для всех наследников.
5. Основная логика программы была перемещена из класса «Контроллер» в класс «Командер».
6. Поля классов теперь не являются публичными.

Из отличий, которые появились благодаря ООП (сравнив два отчета) можно выделить следующее:

Во-первых, .exe теперь занимает больше памяти, а вот сами файлы теперь занимают меньше памяти, однако, это может быть связано с совершенно новым подходом к решению (методы были выделены в отдельные файлы)

Во-вторых, программа гораздо дольше обрабатывает генерирование случайных животных (почти в два раза дольше), что можно заметить по шестому тесту со входными параметрами в 10 000 элементов.

В-третьих, в остальных тестах данная реализация (по времени) показала результаты лучше, чем предыдущая, так что можно сделать выводы, что шестой тест (который является исключением) показывает, что новая реализация генерации случайных животных занимает более длительный период, чем предыдущая.

В-четвертых, использование вектора вместо «самодельного» контейнера для животных сокращает израсходованную память, однако упорядочивание данных с ним крайне не удобно и затратно по временным ресурсам.

Делая выводы из всего перечисленного выше, можно сказать, что данная реализация определено является более эффективной.