Архитектура вычислительных систем.

Вариант 300

Задание 6, Функция 22

Выполнил:  
Лобанович Михаил Михайлович,  
группа БПИ-208

**Задача**

**Сущности:**

1. Самолеты (дальность полета – целое, грузоподъемность – целое)

2. Поезда (количество вагонов – целое)

3. Корабли (водоизмещение – целое; вид судна – перечислимый тип = (лайнер, буксир, танкер)

**Общие для всех альтернатив переменные:**

1. Скорость – целое;

2. Расстояние между пунктами отправления и назначения – действительное.

**Общая для всех альтернатив функция:**

Идеальное время прохождения пути (действительное число).

**Общая для все альтернатив структура:**

Транспорт.

### **Функционал**

1) Зафиксировать для отчета основные характеристики программы, такие как:

число интерфейсных модулей (заголовочных файлов) и

модулей реализации (файлов с определением программных объектов),

общий размер исходных текстов, полученный размер исполняемого кода (если он формируется),

время выполнения программы для различных тестовых наборов данных.

2) В соответствии с вариантом задания разработать программный продукт с

использованием процедурного подхода и статической типизацией.

Разработку осуществить на языке программирования C или на языке C++ в стиле C.

3) Провести отладку и тестирование разработанной программы на заранее

подготовленных тестовых наборах данных.

Количество тестовых наборов данных – не менее пяти.

Число уникальных элементов в тестовых наборах должно варьироваться от нуля до 10000.

При необходимости программа должна правильно обрабатывать переполнение по данным.

Тестовые наборы до 20 элементов должны вводиться из заранее подготовленных тестовых файлов. Тестовые данные с большим числом элементов должны порождаться программно с использованием генераторов случайных наборов данных. Данные, формируемые генератором случайных наборов, должны поддерживать допустимые значения. Управление вводом данных задается из командной строки.

4) Описать структуру используемой ВС с наложением на нее обобщенной схемы

разработанной программы.

--1--.

1. Интерфейсные модули:

transport.h

plane.h

train.h

rnd.h

ship.h

container.h

Всего: 6

2) Модули реализации

trandport.cpp

plane.cpp

train.cpp

rnd.cpp

ship.cpp

container.cpp

Всего: 6

3) Общий размер исходных текстов = 3 + 2 + 3 + 2 + 1 + 1 + 2 + 2 + 1 + 1 + 4 + 2 = 24КБ

4)Время выполнения программы для различных тестовых наборов данных: (тесты рассчитаны на значения в диапазоне от 1 до 20 вводимых элементов, начиная с 500 элементов программа тестировалась при помощи генерации случайных входных данных)

1 тест (1 элемент):

0.001 sec

2 тест (3 элемента):

0.001 sec

3 тест (5 элементов):

0.001 sec

4 тест (10 элементов):

0.001 sec

5 тест (15 элементов):

0.001 sec

6 тест (20 элементов):

0.001 sec

7 тест (500 элементов):

0.003 sec

8 тест(2500):

0.01099 sec

9 тест(5000 элементов):

0.021 sec

10 тест(10000 элементов):

0.04599 sec

--2--.

Программа загружена в систему Github.

--3--.

Тесты находятся в папке cmake-build-debug.

--4--.

|  |  |
| --- | --- |
| **Struct transport** | 32 |
| k: key | 4[0] |
| Speed: int | 4[4] |
| Distance: double | 8[8] |
| p: plane | 8[16] |
| t: train | 8[24] |
| s: ship | 8[32] |

|  |  |
| --- | --- |
| **Struct plane** | 8 |
| Range: int | 4[0] |
| Lifting\_capacity: int | 4[4] |
| **Struct train** | 4 |
| Wagons number: int | 4[0] |

|  |  |
| --- | --- |
| **Struct ship** | 8 |
| Dispacement: int | 4[0] |
| Type: enum | 4[4] |

|  |  |
| --- | --- |
| **Struct container** | 520008 |
| max\_len: enum | 4[0] |
| len: int | 4[4] |

|  |  |
| --- | --- |
| **Main(int argc, char \*argv[]):** | **281568** |
| 1)argv: int | 4[0] |
| 1. argc: char\* | 8[4] |
| 1. size: int | 4[12] |
| 1. c: container | 280000[16] |
| 1. ifst: ifstream | 520[280016] |
| 1. ofst1: ofstream | 512[280536] |
| 1. ofst2: ofstream | 512[281048] |
| **In(container& c,ifstream &ifst)** | **280520** |
| 1. c: container | 280000[0] |
| 1. ifst: ifstream | 520[280000] |
| **Init(container c)** | **280000** |
| 1. c : container | 280000 |
| **InRnd(container& c, int size)** | **280004** |
| 1. c : container | 280000[0] |
| 1. size: int | 4[280000] |
| **Out(container& c,ofstream& ofst)** | **280524** |
| 1. c: container | 280000[0] |
| 1. ifst: ifstream | 520[280000] |
| 1. i: int | 4[280520] |
| **Move to the end(container &c)** | **560012** |
| 1. c : container | 280000[0] |
| 1. average: double | 8[280000] |
| 1. new\_container: container | 280000[280008] |
| 1. index: int | 4[560008] |
|  |  |

**Глобальная память**

|  |  |
| --- | --- |
| **-** | **0** |

**Формат ввода:**

Для ввода одного транспорта следует ввести следующие данные:

Формат ввода меняется в зависимости от типа вводимого транспорта:

1. Самолет: X1 X2 X3 X4 X5,

Где X1 – 1, тип самолета

X2 – скорость самолета

X3 - расстояние между пунктами отправления и назначения

X4 – дальность полета

X5 - грузоподъемность

1. Поезд: X1 X2 X3 X4,

Где X1 – 2, тип поезда

X2 – скорость поезда

X3 - расстояние между пунктами отправления и назначения

X4 – количество вагонов

1. Корабль: X1 X2 X3 X4 X5,

Где X1 – 3, тип корабля

X2 – скорость корабля

X3 - расстояние между пунктами отправления и назначения

X4 – водоизмещение

X5 – тип корабля из перечисляемых

**Расположение входных и выходных данных**

Входные данные лежат в:

HW1/cmake-build-debug/tests/input

Выходные лежат в:

HW1/cmake-build-debug/tests/output

**Cхема**

1. HEAP (Куча) - int\*
2. Стек

Изображение выглядит как текст, доска

Автоматически созданное описание