Шульмин Павел, БПИ207

Задание: 6; функция 20

Описание задания

Разработать программный продукт с использованием объектно-ориентированного подхода и статической типизацией. Программа должна содержать следующие структуры:

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
Пассажирский транспорт	1. Самолеты (дальность полета – целое, грузоподъемность – целое) 2. Поезда (количество вагонов – целое) 3. Корабли (водоизмещение – целое; вид судна – перечислимый тип = (лайнер, буксир, танкер)	1. Скорость – целое; 2. Расстояние между пунктами отправления и назначения – действительное	Идеальное время прохождения пути (действительное число)

Дополнительная функция: удалить из контейнера те элементы, для которых идеальное время прохождения пути больше чем среднее арифметическое идеальное время прохождения пути для всех элементов контейнера. Остальные элементы передвинуть к началу контейнера с сохранением порядка.

Структурная схема архитектуры ВС с программой

Таблица типов	
Название	**Размер (байты)**
int	4
double	8
FILE	216
class Container: len: int storage: *Transport[10000]	48024 4[0] 48000[4]
class Transport: speed: int distance: double Ссылки на виртуальные методы	24 4[0] 8[4] 8[8]
class Plain: range: int capacity: int	32 4[0] 4[4]
class Train: railway_carriages: int	32 4[0]

class Ship:
  displacement: int
  type: enum Type {
  liner: Type
  tow: Type
  tanker: Type
  }
	32
	4[0]
	4[20]
	4[24]
	4[28]
	4[32]

class Random:
  first: int
  last: int
	8
	4[0]
	4[4]

Схема работы функции main

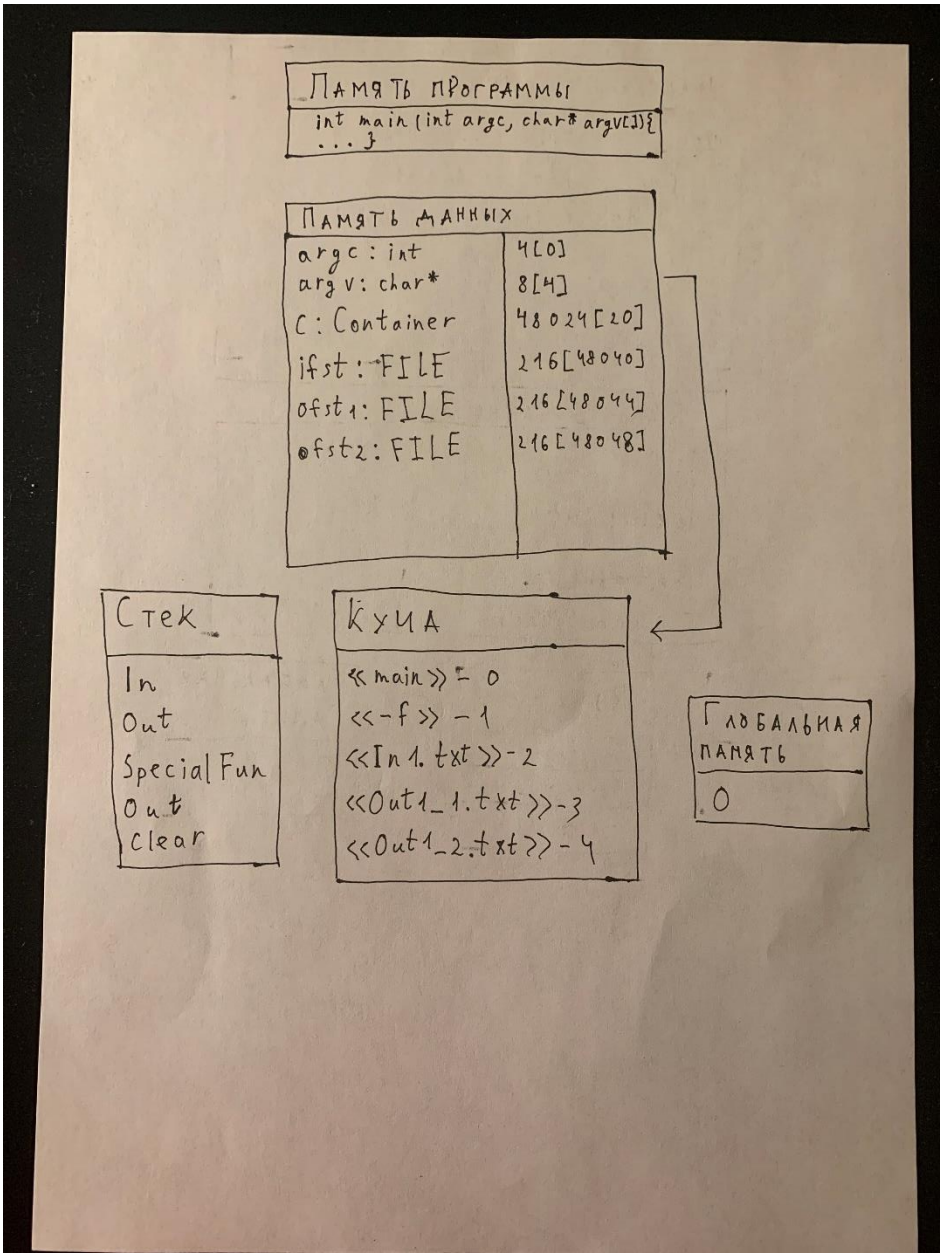
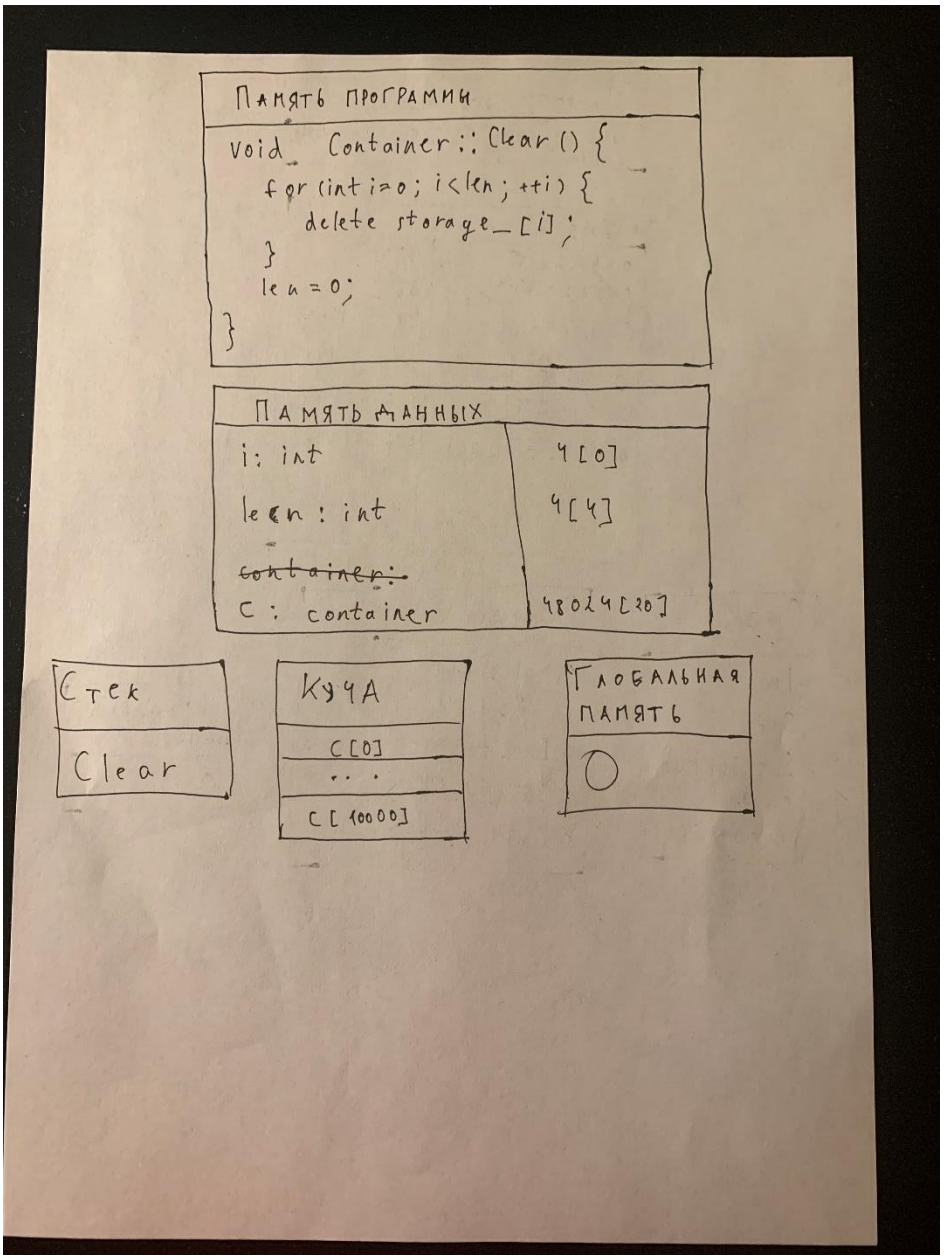


Схема работы функции clear



Характеристики программы

1. 6 заголовочных файлов
2. 6 модулей реализации
3. Общий размер исходных текстов: 470
4. Размер исполняемого кода: 18,9 Кб
5. Таблица с номером теста и временем

№	t (мс)
1	6
2	5
3	25
4	4
5	12

Отзыв и сравнение с процедурной парадигмой

ООП широко применяется на практике. Принципы объектно-ориентированного программирования помимо обработки событий – это инкапсуляция, наследование и полиморфизм. Они особенно полезны и необходимы при разработке тиражируемых и простых в сопровождении приложений.
Объект объединяет в себе методы и свойства, которые не могут существовать отдельно от него. Поэтому если объект удаляется, то удаляются его свойства и связанные с ним методы. При копировании происходит то же самое: объект копируется как единое целое.
Мне кажется процедурным программированием удобно пользоваться для написания небольших скриптов, которые будет поддерживать один разработчик, а не команда. ООП в свою очередь удобно использовать для крупных проектов, разрабатываемых и поддерживаемых группой разработчиков.