САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССОЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

Факультет «инфокоммуникационных технологий»

Направление подготовки «11.03.02 Инфокоммуникационные технологии и системы связи»

**Домашняя работа**

**Реализация программной модели инфокоммуникационной системы**

Выполнил:

Гололобов Владимир Владимирович

Группа К3120

Проверил:

Марченко Елена Вадимовна

Санкт-Петербург

2019

**Задание:** реализовать систему для аренды памяти клиентом у арендатора

**Описание решения:**

Для создания данной системы потребовалось разделить сущности на два смысловых блока. Первый: модуль с арендуемыми устройствами памяти, а второй с людьми, арендующими и выставляющими на аренду эти устройства. Они были разнесены по разным файлам: Application\_system и Sources\_Consumers соответственно.

Для каждого модуля был написан абстрактный класс: Person и Storage Unit соответственно. Класс Person представляет человека и имеет общую информацию о нём такую как: имя, фамилия, отчество и дата рождения. Данный класс наследуется классами Consumer и Owner, чтобы использовать конструктор, переменные и методы родительского класса и не захламлять код переписыванием этого всего. Так как переменные класса Person являются private переменными (инкапсуляция), были написаны сеттеры и геттеры для получения и установки значений этих полей. Кроме того, данный класс имеет одну чистую виртуальную функцию, которая отражает количество какого-то ресурса у человека. У покупателя это деньги, а у владельца количество юнитов для последующей аренды. Для них были переопределен данный виртуальный метод (полиморфизм). Класс Storage\_Unit представляет юнит для хранения информации. Это может быть как дата центр, так и жёсткий диск. Класс Storage\_Unit имеет общую информацию о юните: название, общая память, оставшаяся память и цена за гигабайт. Данный класс имеет одну чисто виртуальную функцию, которая отображает тип юнита. Storage\_Unit наследуется классам DataCenter, который представляет конкретный дата центр. Он имеет в дополнение местоположение и глобальный параметр – список существующих дата центров. Это сделано для быстрого поиска клиентом нужного дата центра.

Принцип работы системы, следующий: создаётся владелец, также юниты памяти, которые создаются и формируют их список. Потом владельцу добавляется юнит. Дальше создаётся клиент. Он пытается арендовать память у юнита с помощью советующего метода, передавая в качестве параметров название юнита и желаемое количество памяти. Функция рассчитывает стоимость памяти в данном юните, проверяет осталась ли память там в принципе и проверяет есть ли достаточное количество денег у человека. Если условия соблюдены, функция совершает покупку, вычитая память из текущий у юнита и деньги у пользователя

**Блок-схема:**

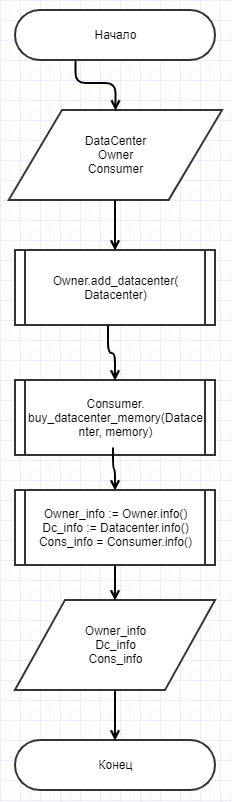


Рисунок 1 – Блок-схема

**Диаграмма классов:**

1. Классы связанные с людьми

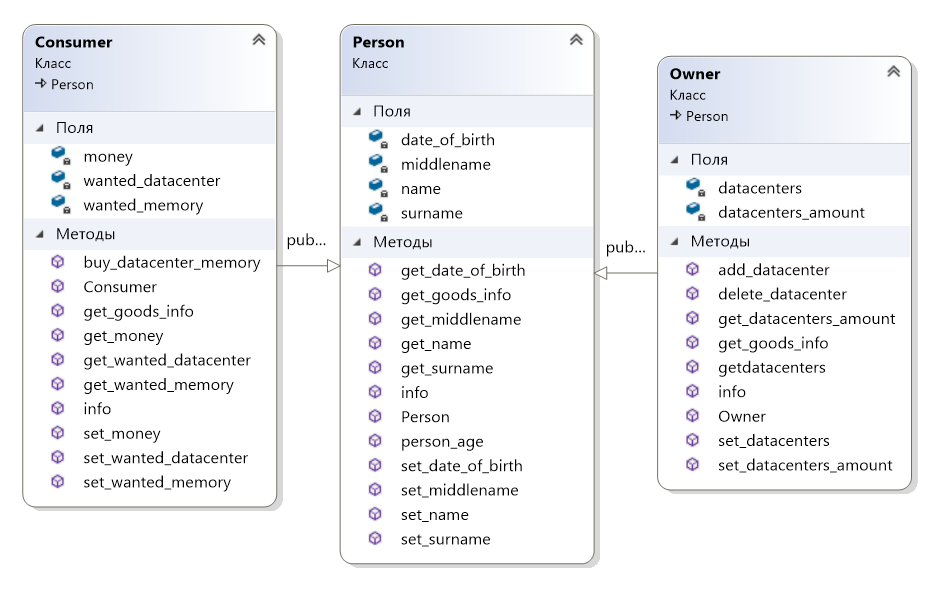


Рисунок 2 – Диаграмма классов

1. Классы связанные с юнитами хранения

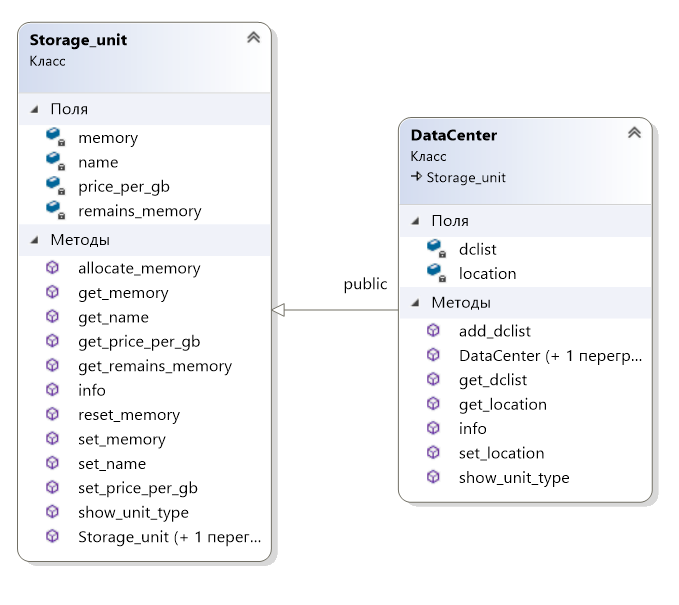


Рисунок 3 – Диаграмма классов

**Вывод:** в ходе выполнения домашнего задания, была реализована система аренды памяти. Из работы был сделан вывод, что полезно создавать абстрактные или шаблонные классы, которые в общих чертах описывают сущность, а потом строить производные классы, которые по необходимости дополняют родительский класс. Такой подход делает код компактным и логически понятным, также вводит определённую структуру в разработку дальнейшей системы, чем предотвращает хаос