**Практическая работа №9**

Задание:

1. **Разработать класс в соответствии с вариантом.**

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Контейнер | Класс | Свойства |
| 1 | Почта | Посылка | №, Адрес, Вес, Статус |
| 2 | Музей | Экспонат | №, Название, Эпоха, Описание |
| 3 | Парикмах. | Клиент | №, Фамилия, Имя, Вид стрижки |
| 4 | Провайдер | Подписчик | №, Фамилия, Имя, Тариф |
| 5 | Хостел | Гость | №, Фамилия, Имя, Номер комнаты |

Пример реализации класса «Медикамент»:

|  |
| --- |
| class Drug{  public:  int number;  string name;  double price;  int amount;    Drug() {  ...  }    Drug(int number, string тame, double price, int amount){  ...  }    void Display(){  ...  }  }; |

1. **Разработать класс-контейнер, содержащий выше написанный класс.**

Требуемый функционал класса-контейнера:

1. Создание контейнера
2. Добавление объекта в конец контейнера.
3. Удаление объекта с конца контейнера
4. Отображение всех объектов
5. Отображение количества объектов
6. Очистка контейнера

Пример реализации класса-контейнера «Аптека»:

|  |
| --- |
| class DrugShop {  private:  int counter; //счетчик элементов  int lenght; //длина массива  Drug \*mas; //указатель для динамического массива    public:  DrugShop(int lenght){  this->lenght = lenght;  mas = new Drug[lenght];  }    void add(Drug drug) //метод добавления объекта в контейнер  ...  }    void delete(int n){ //метод удаления объекта из контейнера по индексу  ...  }    void display(){ //метод отображения всех объектов  ...  }    void showCounter(){ //метод отображения количества объектов в контейнере  ...  }    void erase(){ //метод очистки контейнера  ...  }    ~DrugShop () {  ...  }  }; |

1. **Разработать меню, позволяющее удобно использовать класс-контейнер**

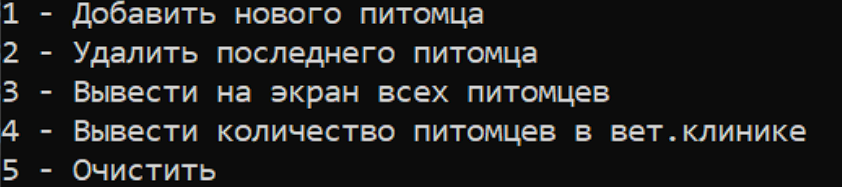


Рисунок 1 – Пример консольного меню