

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ МОЛДОВА  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ МОЛДОВЫ  
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ, ИНФОРМАТИКИ И  
МИКРОЭЛЕКТРОНИКИ  
ДЕПАРТАМЕНТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И АВТОМАТИКИ

ОТЧЕТ

По лабораторной работе №2

по дисциплине: АРРОО

Тема: Single Responsibility и Interface Segregation.

Выполнил: студент группы TI-154

Сокол М.

Проверил:

Pecari M.

Кишинев 2018

## Цель лабораторной работы

Создать программу, которая содержит имплементацию двух принципов SOLID – Single Responsibility и Interface Segregation.

## Теория

SOLID - мнемонический акроним, введённый Майклом Фэзерсом для первых пяти принципов, названных Робертом Мартином в начале 2000-х, которые означали пять основных принципов объектно-ориентированного программирования и проектирования. Эти принципы, когда применяются вместе, предназначены для повышения вероятности того, что программист создаст систему, которую будет легко поддерживать и расширять в течение долгого времени

Расшифровка:

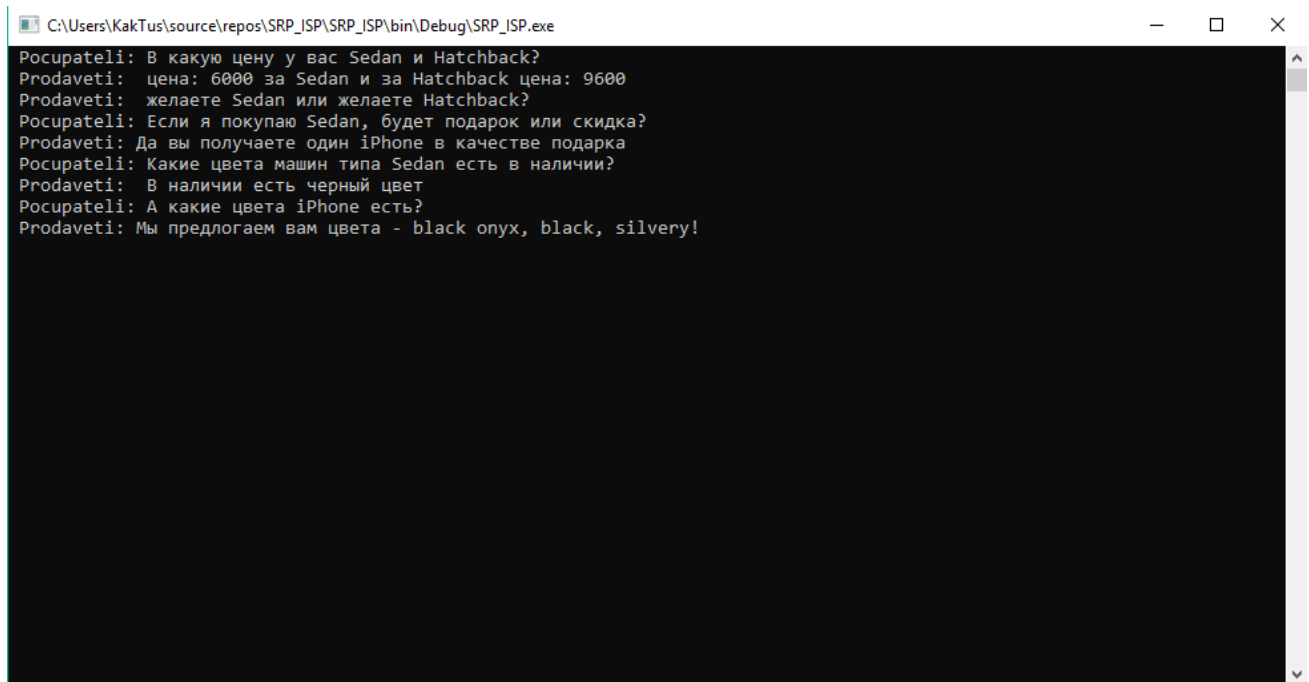
- Single responsibility (принцип единственной ответственности)
- Open-closed (принцип открытости / закрытости)
- Liskov substitution (принцип подстановки Барбары Лисков)
- Interface segregation (принцип разделения интерфейса)
- Dependency inversion (принцип инверсии зависимостей)

Принцип единственной ответственности (single responsibility principle) обозначает, что каждый объект должен иметь одну обязанность и эта обязанность должна быть полностью инкапсулирована в класс. Все его сервисы должны быть направлены исключительно на обеспечение этой обязанности.

Принцип разделения интерфейса (interface segregation) в формулировке Роберта Мартина: «клиенты не должны зависеть от методов, которые они не используют». Принцип разделения интерфейсов говорит о том, что слишком «толстые» интерфейсы необходимо разделять на более маленькие и специфические, чтобы клиенты маленьких интерфейсов знали только о методах, которые необходимы им в работе. В итоге, при изменении метода интерфейса не должны меняться клиенты, которые этот метод не используют.

## Описание выполнения лабораторной работы

Ссылка на репозиторий: <https://github.com/ghipermax/APPOO>



```
C:\Users\KakTus\source\repos\SRP_ISP\SRP_ISP\bin\Debug\SRP_ISP.exe
Pocupateli: В какую цену у вас Sedan и Hatchback?
Prodaveti: цена: 6000 за Sedan и за Hatchback цена: 9600
Prodaveti: желаете Sedan или желаете Hatchback?
Pocupateli: Если я покупаю Sedan, будет подарок или скидка?
Prodaveti: Да вы получаете один iPhone в качестве подарка
Pocupateli: Какие цвета машин типа Sedan есть в наличии?
Prodaveti: В наличии есть черный цвет
Pocupateli: А какие цвета iPhone есть?
Prodaveti: Мы предлагаем вам цвета - black опух, black, silvery!
```

Рисунок 1 – Результат работы программы

**Вывод:**

Во время выполнения данной лабораторной работы было произведено ознакомление с двумя из принципов SOLID, которые были реализованы в программе, а именно: Single Responsibility и Interface Segregation.. Параллельно с этим был изучен популярный паттерн под названием Factory Method, который задаёт интерфейс создания всех доступных типов продуктов, а каждая конкретная реализация фабрики порождает продукты одной из вариаций.