|  |  |
| --- | --- |
| Вариант 1 | Вариант 2 |
| 1)Написать интерфейс IComputer содержащий:   1. Тип процессора (Сделать Enum) 2. Название производителя (Сделать Enum) 3. Тип операционной системы (Сделать Enum) 4. Тактовая частота процессов (int) 5. Объём оперативной памяти (int) 6. Установленное ПО (List<string>) 7. Пользователи системы (List <string>)   Написать интерфейс IOverclock содержащий:   1. Метод OverclockTheComputer() (продумать его реализацию на основе типов процессора и проверку что он может быть вызван только один раз).   Написать класс Computer:   1. Реализующий интерфейсы IComputer и IOverclock. 2. Конструкторы по умолчании и с параметрами. 3. Статический метод Generate возвращающий экземпляр класса Computer со случайным набором параметров. 4. Статический метод Generate100 возвращающий 100 экземпляров класса Computer со случайным набором параметров. | 1)Написать интерфейс IConsole содержащий:   1. Тип процессора (Сделать Enum) 2. Название производителя (Сделать Enum) 3. Тип носителей для консоли (CD, DVD, Картриджи и т.д ) (Сделать Enum) 4. Модель (зависит от производителя) 5. Объём жесткого диска (int) 6. Установленные игры (List<string>) 7. Аккаунты системы (List <string>)   Написать интерфейс IHack содержащий:   1. Метод TtySoftwareHack() 2. Метод TtyHardwareHack()   Написать класс Console:   1. Реализующий интерфейсы IConsole и IHack. 2. Конструкторы по умолчании и с параметрами. 3. Статический метод Generate возвращающий экземпляр класса Console со случайным набором параметров.   Статический метод Generate100 возвращающий 100 экземпляров класса Console со случайным набором параметров. |
| 2)Создать коллекцию List<Computer >  Поместить в нее результат работы метода Generate100   1. Отфильтровать по Типу процессора 2. Отфильтровать по Типу процессора и названию производителя 3. Отфильтровать по коллекции пользователей и объёму оперативной памяти | 2)Создать коллекцию List<Console>  Поместить в нее результат работы метода Generate100   1. Отфильтровать по Типу процессора 2. Отфильтровать по Типу процессора и названию производителя   Отфильтровать по коллекции пользователей и объёму оперативной памяти |
| 3)Создать коллекцию List<Computer >  Поместить в нее результат работы метода Generate100   1. Отсортировать по Типу процессора 2. Отсортировать по Типу процессора и названию производителя | 3)Создать коллекцию List<Console>  Поместить в нее результат работы метода Generate100   1. Отсортировать по Типу процессора   Отсортировать по Типу процессора и названию производителя |
| 4)Создать коллекцию List< Computer >  Поместить в нее результат работы метода Generate100  Написать SELECT который возвращает коллекцию, содержащую следующие поля   1. Тактовая частота процессов (int) 2. Объём оперативной памяти (int) 3. Установленное ПО (List<string>) | 4)Создать коллекцию List<Console >  Поместить в нее результат работы метода Generate100  Написать SELECT который возвращает коллекцию, содержащую следующие поля   1. Тактовая частота процессов (int) 2. Объём жесткого диска (int) 3. Аккаунты системы (List <string>) |
| 5)Написать интерфейс IManufacturer содержащий данные о производителях:   1. Название 2. Страна (Сделать Enum) 3. Количество сотрудников   Написать класс Manufacturer   1. Реализующий интерфейс IManufacturer. 2. Конструкторы по умолчании и с параметрами.   Создать и заполнить данными коллекцию List<Manufacturer>  Показать, как работает:   1. Внутренние соединение | 5)Написать интерфейс IManufacturer содержащий данные о производителях:   1. Название 2. Страна (Сделать Enum) 3. Количество сотрудников   Написать класс Manufacturer   1. Реализующий интерфейс IManufacturer. 2. Конструкторы по умолчании и с параметрами.   Создать и заполнить данными коллекцию List<Manufacturer>  Показать, как работает:  Внутренние соединение |
| 6)Написать распирающий метод для типа string, удаляющий русские буквы. | 6)Написать распирающий метод для типа string, удаляющий латинские буквы. |
| 7)Переписать задания 2-5 используя расширяющие методы LINQ. | 7)Переписать задания 2-5 используя расширяющие методы LINQ. |