

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ**  
**ОДЕССКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ**  
**ИМЕНИ И.И. МЕЧНИКОВА**

**Индивидуальная работа по теме «Программирование (Объектные технологии и языки)»**

**на тему:**

**Порт (Вариант В)**

Сделал:

студент 1 курса по специальности  
компьютерная инженерия

Набока Вячеслав Дмитриевич

Преподаватель:

Антоненко Александр Сергеевич

## Задание

Создать иерархию классов: корабль – базовый класс и пассажирский корабль – производный. Корабль имеет мощность двигателя, водоизмещение, название, порт приписки, экипаж. Пассажирский корабль имеет дополнительные поля: количество пассажиров, количество шлюпок, вместимость шлюпки. Реализовать метод проверки, что шлюпок хватает на пассажиров и членов экипажа и метод увеличения числа шлюпок до минимально необходимого, если их не хватает. Также реализовать класс грузовой корабль с дополнительным параметром – грузоподъемность. В основном проекте реализовать создание, модификацию и удаление кораблей, а также добавление шлюпок на пассажирские корабли.

## Описание программы

### Класс CrewMember

Класс CrewMember представляет собой члена экипажа в корабле. Он предоставляет свойства и методы для управления информацией о члене экипажа.

Класс CrewMember инкапсулирует следующие свойства:

`string FullName` - полное имя члена экипажа.

`int Age` - возраст члена экипажа.

`double Height` - рост члена экипажа в сантиметрах.

`string ShipName` - название корабля, назначенного члену экипажа.

### Конструктор:

`CrewMember(string fullName = "Unknown", int age = 0, double height = 0)`

Инициализирует новый экземпляр класса CrewMember с указанным полным именем, возрастом и ростом.

### Параметры:

`fullName`: Полное имя члена экипажа.

`age`: Возраст члена экипажа.

`height`: Рост члена экипажа в сантиметрах.

### Исключения:

`ArgumentException`: Если указанный возраст или рост меньше 0.

### Методы:

`AssignShip(string shipName)` - Назначает корабль члену экипажа.

### Параметры:

`shipName`: Название корабля, который будет назначен члену экипажа.

`Info()` – возвращает строку, содержащую информацию о члене экипажа.

## Класс BaseShip

Класс `BaseShip` представляет собой базовый класс для кораблей. Он предоставляет свойства и методы для управления информацией о корабле и его экипаже.

Класс `BaseShip` инкапсулирует следующие свойства:

`string Name` - получает или задает название корабля.

`string RegistrationPort` - получает или задает порт регистрации корабля.

`int EnginePower` - получает или задает мощность двигателя корабля.

### Исключения:

`ArgumentException`: если указанная мощность двигателя меньше 0.

`int Draught` - получает или задает осадку корабля.

### Исключения:

`ArgumentException`: Если указанная осадка меньше 0.

`List<CrewMember> Crew` - получает список членов экипажа корабля.

## Конструктор:

`BaseShip(string name = "Unknown", string registrationPort = "Unsigned", int enginePower = 0, int draught = 0)`

Инициализирует новый экземпляр класса `BaseShip` с указанным названием, портом регистрации, мощностью двигателя и осадкой.

Параметры:

`name`: Название корабля.

`registrationPort`: Порт регистрации корабля.

`enginePower`: Мощность двигателя корабля.

`draught`: Осадка корабля.

### Исключения:

`ArgumentException`: Если указанная мощность двигателя или осадка меньше 0.

## Методы:

`AddCrewMember(CrewMember human)` - добавляет члена экипажа на корабль.

**Параметры:**

human: Член экипажа для добавления.

AssignCrew(List<CrewMember> crew) - назначает экипаж на корабль.

**Параметры:**

crew: Список членов экипажа для назначения.

RemoveCrewMember(CrewMember human) - удаляет члена экипажа с корабля.

**Параметры:**

human: Член экипажа для удаления.

CrewInfo() - возвращает строку, содержащую информацию о членах экипажа.

Info() - возвращает строку, содержащую информацию о корабле.

## **Класс PassengerShip**

Класс PassengerShip является производным классом от класса BaseShip и представляет пассажирский корабль. Он расширяет функциональность базового корабля, добавляя свойства и методы, специфичные для пассажирских судов.

Класс PassengerShip наследует все свойства и методы класса BaseShip и добавляет следующие свойства:

int PassengerAmount - получает или задает количество пассажиров на корабле.

int RawboatAmount - получает или задает количество шлюпок на корабле.

int RawboatCapacity - получает вместимость одной шлюпки на корабле

**Конструктор:**

```
PassengerShip(string name = "Unknown", string registrationPort = "Unsigned", int enginePower = 0, int draught = 0, int passengerAmount = 0, int rawboatAmount = 0, int rawboatCapacity = 0)
```

Инициализирует новый экземпляр класса PassengerShip с указанными параметрами.

**Параметры:**

name: Название корабля.

registrationPort: Порт регистрации корабля.

**enginePower:** Мощность двигателя корабля.  
**draught:** Осадка корабля.  
**passengerAmount:** Количество пассажиров на корабле.  
**rawboatAmount:** Количество шлюпок на корабле.  
**rawboatCapacity:** Вместимость одной шлюпки на корабле.

## Методы:

**HasEnoughRawboats()** - проверяет, есть ли достаточное количество шлюпок для всех членов экипажа и пассажиров на корабле, возвращает значение типа **bool**.

**TryIncreaseRawboatAmount()** - пытается увеличить количество шлюпок на корабле, если текущее количество шлюпок недостаточно для всех членов экипажа и пассажиров. Возвращает **true**, если количество шлюпок было увеличено успешно, и **false**, если уже есть достаточное количество шлюпок.

**Info()** - возвращает информацию о пассажирском корабле, включая его базовую информацию, количество пассажиров, количество шлюпок и вместимость шлюпки.

## Класс CargoShip

Класс **CargoShip** является производным классом от класса **BaseShip** и представляет грузовой корабль. Он расширяет функциональность базового корабля, добавляя свойства и методы, специфичные для грузовых судов.

Класс **CargoShip** наследует все свойства и методы класса **BaseShip** и добавляет следующие свойства:

**int CarryingCapacity** - получает или задает грузоподъемность корабля, т.е. максимальный вес груза, который корабль способен перевозить.

## Конструктор:

```
CargoShip(string name = "Unknown", string registrationPort =  
"Unsigned", int enginePower = 0, int draught = 0, int  
carryingCapacity = 0)
```

Инициализирует новый экземпляр класса **CargoShip** с указанными параметрами.

## Параметры:

**name:** Название корабля.  
**registrationPort:** Порт регистрации корабля.

enginePower: Мощность двигателя корабля.  
draught: Осадка корабля.  
carryingCapacity: Грузоподъемность корабля.

### **Методы:**

Info() - возвращает информацию о грузовом корабле, включая его базовую информацию и грузоподъемность.

## **Класс PortSystem**

Класс PortSystem представляет систему порта, которая управляет списком кораблей, находящихся в порту. Он предоставляет методы для добавления, удаления и поиска кораблей, а также получения информации о всех кораблях в порту.

Класс PortSystem инкапсулирует следующие свойства:

List<BaseShip> ShipList - получает список кораблей в порту.

### **Конструктор:**

**PortSystem()**

Инициализирует новый экземпляр класса PortSystem.

### **Методы:**

AddShip(BaseShip baseShip) - Добавляет корабль в порт.

**Параметры:**

baseShip: Корабль для добавления.

RemoveShip(BaseShip baseShip) – удаляет корабль из порта.

**Параметры:**

baseShip: Корабль для удаления.

FindShipByName(string name) - ищет корабль по имени, возвращает экземпляр класса BaseShip или null

**Параметры:**

name: Название корабля для поиска.

GetAllShipsInfo() – возвращает строку, содержащую информацию о всех кораблях в порту.

## Класс Extentions

Класс Extentions является статическим классом и содержит методы-расширения для различных типов данных.

### Методы-расширения:

**IsArrayMinSize<T>(this T[] array, int minSize)** - Проверяет, является ли длина массива array не меньшей, чем заданное минимальное значение minSize, возвращает true, если длина массива array не меньше, чем minSize, иначе false.

#### Параметры:

**array:** Массив, который нужно проверить.

**minSize:** Минимальный размер массива.

**RandomElement<T>(this T[] array)** – Возвращает случайный элемент типа T из массива array. Если массив пуст или равен null, возвращает значение по умолчанию для типа T.

#### Параметры:

**array:** Массив, из которого нужно получить случайный элемент.

## Класс RandomPortElements

Класс RandomPortElements содержит статические методы для генерации случайных данных, связанных с портовыми элементами.

### Методы:

**GeneratePersonName()** – возвращает случайно сгенерированное полное имя человека.

**GenerateShipName()** - генерирует случайное название корабля.

**GenerateShipPort()** - генерирует случайное название порта.

**GenerateRandomCrewMember()** - генерирует случайного члена экипажа, с случайно заданными параметрами.

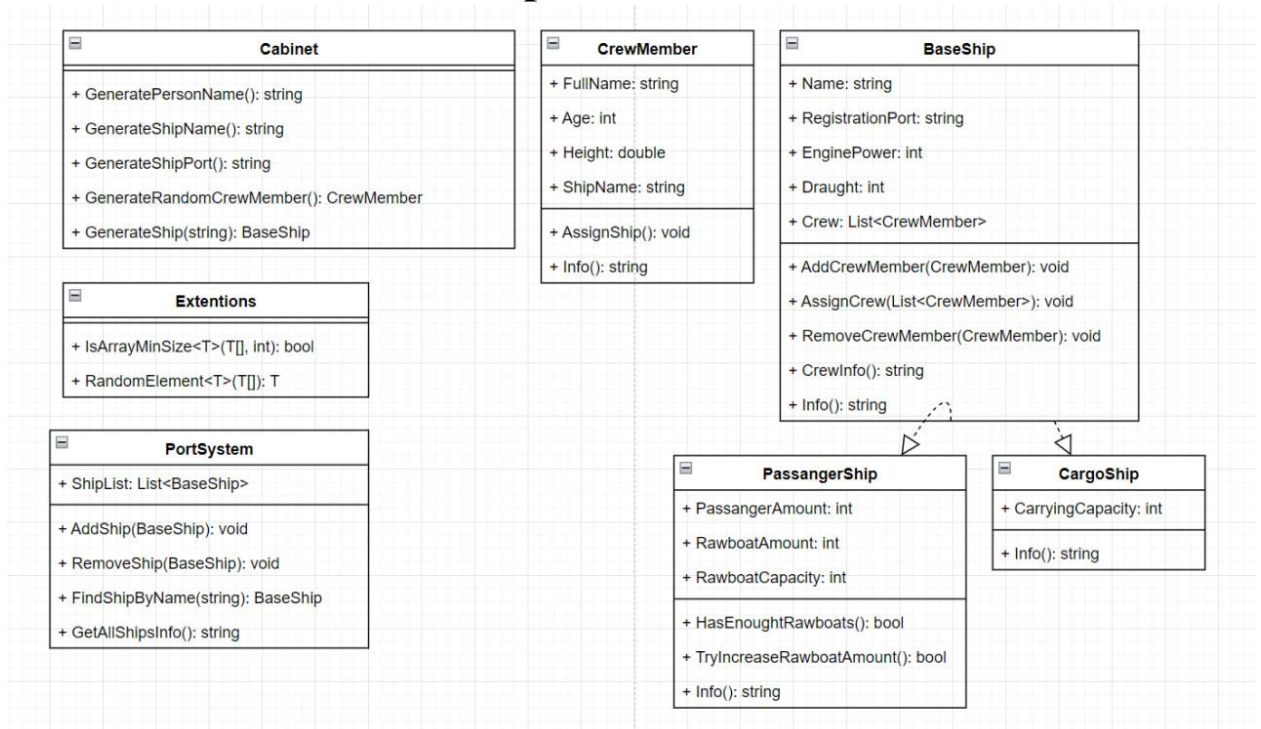
**GenerateShip(string shipType)** - Генерирует случайный корабль в соответствии с заданным типом.

#### Параметры:

**shipType:** Тип корабля ("Base", "Passenger", "Cargo").

Возвращает случайно сгенерированный объект BaseShip или его производного класса, в зависимости от заданного типа корабля.

## Диаграмма классов



## Пример работы программы

Программа начинается с приветствия и напоминания о том что есть команда «Help» позволяющая увидеть список всех доступных команд.

```
Welcome to Port System Simulator
Type "Help" for available comand list

Help
Help - show all available comands
AddRandomShip {Ship Type} {Ship Count} - adding random ships of selected type
AddShip {Ship Type} {Ship Parametr} - manually add ship, parametr vary of Ship Type
ModifyShip {Ship Type} {Ship Name} {New Value} - modify ship parametr by ship name and type
RemoveShip {Ship Name} - removes ship by name
TryAddRawboats {Ship Name} - increases rawboats amount in passanger ship
Info {Type} - shows info about: "AllShips", "Crew {Ship Name}", "Ship {Ship Name}"
Available Ship Types: Base, Passenger, Cargo
```

Для начала можно добавить корабли к портам, например можно сделать случайные корабли по разным типам, а после вывести информацию о них.

```
AddRandomShip Cargo 3
3 random ships added to port

Info AllShips
Name: Edge, Registration Port: Rotterdam, EnginePower: 714, Draught: 81 Carrying capacity: 5183
Name: Albatros, Registration Port: Singapoure, EnginePower: 502, Draught: 64 Carrying capacity: 4326
Name: Titanik, Registration Port: Singapoure, EnginePower: 630, Draught: 69 Carrying capacity: 8132
```

Если данные корабли нам не подходят то можно удалить их.



```
RemoveShip Titanik
Ship Titanik Removed
Info AllShips
Name: Edge, Registration Port: Rotterdam, EnginePower: 714, Draught: 81 Carring capacity: 5183
Name: Albatros, Registration Port: Singapoure, EnginePower: 502, Draught: 64 Carring capacity: 4326
```

Также любые данные можно модифицировать вручную.

```
Info AllShips
Name: Titanik, Registration Port: Shanghai , EnginePower: 261, Draught: 98 Carring capacity: 6448
Name: Derge, Registration Port: Rotterdam, EnginePower: 222, Draught: 34 Carring capacity: 4551
Name: Albatros, Registration Port: Singapoure, EnginePower: 544, Draught: 77 Carring capacity: 8027

ModifyShip Cargo Derge EnginePower 1000
Ship successfully modified

Info AllShips
Name: Titanik, Registration Port: Shanghai , EnginePower: 261, Draught: 98 Carring capacity: 6448
Name: Derge, Registration Port: Rotterdam, EnginePower: 1000, Draught: 34 Carring capacity: 4551
Name: Albatros, Registration Port: Singapoure, EnginePower: 544, Draught: 77 Carring capacity: 8027
```

## Вывод

Программа имитирует работу корабельных портов. Она имеет иерархию классов: корабль – базовый класс и производные пассажирский и грузовой корабль. Корабли имеют мощность двигателя, водоизмещение, название, порт приписки, экипаж. Пассажирский корабль имеет дополнительные поля: количество пассажиров, количество шлюпок, вместимость шлюпки. Грузовой корабль имеет дополнительное поле – грузоподъемность.

В пассажирском корабле реализован метод проверки, что шлюпок хватает на пассажиров и членов экипажа и метод увеличения числа шлюпок до минимально необходимого, если их не хватает.

В основном проекте реализованы как случайное так и пользовательское создание, модификация и удаление кораблей, а также добавление шлюпок на пассажирские корабли.