**Министерство науки и высшего образования**

**Федеральное государственное бюджетное образовательное учреждение высшего образования “Северо-Осетинский государственный университет им.К.Л.Хетагурова”**

**Факультет математики и компьютерных наук  
  
  
  
  
  
Курсовая работа  
Многопоточность на корабликах**

Студентки 1 курса

Cпециальность - 01.03.02 Прикладная математика и информатика

Квалификация - бакалавр

Форма обучения - очно

А. А. Мамсурова

**Владикавказ, 2022**

# Оглавление

[**Введение** 3](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.gjdgxs)

[**Глава 1. Теоретическая часть** 3](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.30j0zll)

Способ реализации

[**Глава 2. Практическая часть**  4](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.3znysh7)

[Словесное описание 4](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.2et92p0)

Структура 5

[**Глава 3. Документация**](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.3dy6vkm) 5

[Руководство пользователя](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.1t3h5sf) 5

[Руководство программиста](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.2s8eyo1) 6

[Заключение](https://docs.google.com/document/d/1K4AEnWXilv2ne5jHVevxp187wnirn_Yp/edit#heading=h.17dp8vu) 6

## **Введение**

**Многопоточность—**свойство платформы(например, операционной системы, виртуальной машины и т. д.) или приложения, состоящее в том, что процесс, порождённый в операционной системе, может состоять из нескольких потоков(поток представляет некоторую часть кода программы), выполняющихся «параллельно», то есть без предписанного порядка во времени.

При выполнении программы каждому потоку выделяется определенный квант времени. И при помощи многопоточности мы можем выделить в приложении несколько потоков, которые будут выполнять различные задачи одновременно. Представим что у нас есть приложение которое отправляет фотографии кому-либо.Отправка фотографий будет занимать какое то количество времени.С помощью многопоточности мы сможем дать доступ к UI пользователю во время отправки фотографий.

### Постановка задачи

Передо мной была поставлена следующая задача:

* ознакомиться с таким понятием как многопоточность.
* рассказать и продемонстрировать как плюсы так и минусы многопоточного программирования.
* разработать консольное и WinForm приложение для демонстрации работы многопоточности.
* правильно разбить задачу на параллельность.
* синхронизировать потоки.

## **Глава 1. Теоретическая часть**

### **Основные понятия многопоточности**

**Многозадачность (multitasking)** — свойство операционной системы или среды выполнения обеспечивать возможность параллельной (или псевдо параллельной) обработки нескольких задач.

**С точки зрения пользователя:**

**Процесс** — экземпляр программы во время выполнения.

**Потоки** — ветви кода, выполняющиеся «параллельно», то есть без предписанного порядка во времени.

**С точки зрения операционной системы:**

**Процесс** — это просто контейнер, в котором находятся ресурсы программы:

* адресное пространство
* потоки
* открытые файлы
* дочерние процессы
* и т.д.

**Поток** — это просто контейнер, в котором находятся:

* Счётчик команд
* Регистры
* Стек

**Отличие процесса от потока-**процесс рассматривается ОС, как заявка на все виды ресурсов (память, файлы и пр.), кроме одного — процессорного времени. Поток — это заявка на процессорное время. Процесс — это всего лишь способ группировать взаимосвязанные данные и ресурсы, а потоки — это единицы выполнения (unit of execution), которые выполняются на процессоре.

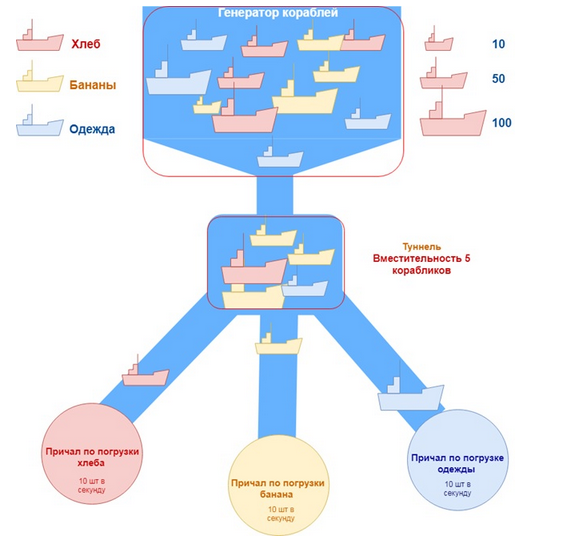
### Словесное описание

Есть транспортные корабли, которые подплывают к туннели и далее плывут к причалам для погрузки разного рода товара.Они проходят через узкий туннель где одновременно могут находиться только 5 кораблей. Необходимо правильным образом организовать работу кораблей,причалов, тоннеля.

Изначально свою работу начинают “Генератор кораблей” и “Причалы”.Они параллельно начинают свою работу.

Далее идет процесс отправки корабля в тоннель и далее к причалу

* Сгенерированный корабль отправляется в тоннель(если тоннель не заполнен).
* Каждый причал обращается к тоннелю и ищет корабль нужного типа(если причал не занят и корабль нужного типа имеется в тоннель корабль отправляется на погрузку).
* После завершения погрузки корабль отправляется в свободное плавание.



## **Глава 2. Практическая часть**

### **Способ реализации**

### **Структура консольного приложения**

Для решения поставленной задачи были созданы такие классы как: “Enums”, “Ship”, ”Tunnel”, “ShipGenerator”, “PierLoader”. “Enums”-содержит в себе все необходимые перечисления. “Ship”-описывает корабль. “Tunnel”-регулирует работу тоннеля. “Ship generator”- создает определенное количество кораблей и помещает их в тоннель. “PierLoader”-бесконечно пытается забрать корабли из тоннеля.

### **Структура winform приложения**

## **Глава 3. Документация**

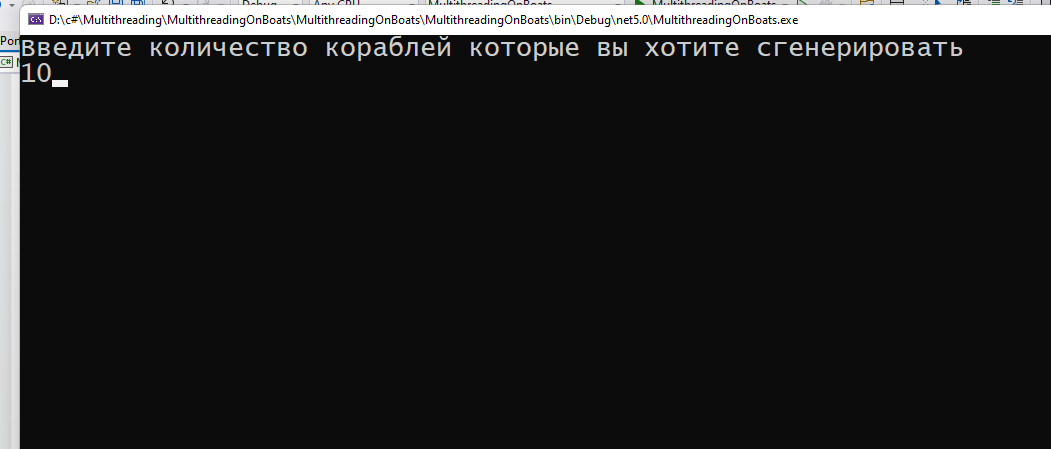
### Руководство системного администратора

Приложение реализовано в среде Visual Studio 2022 на языке C#. Для запуска приложения требуется Windows 7-10. Особых требований к объему памяти и производительности нет. Для работы приложения надо запустить файл MultithreadingOnBoats.exe

### Руководство пользователя

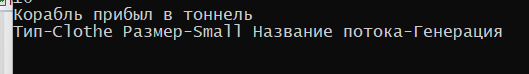
**Консольное приложение**

Для запуска консольного приложения необходимо запустить MultithreadingOnBoats.exe. Далее в консоле вам необходимо ввести количество кораблей которое вам необходимо заполнить товаром.

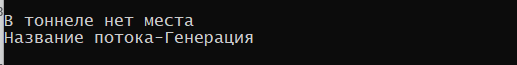


Далее программа начнет свое выполнение и в окне вывода вы сможете увидеть процесс ее работы.

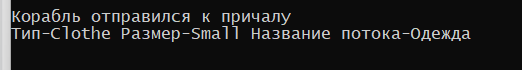
“Корабль прибыл в тоннель”-сгенерированный корабль смог попасть в тоннель



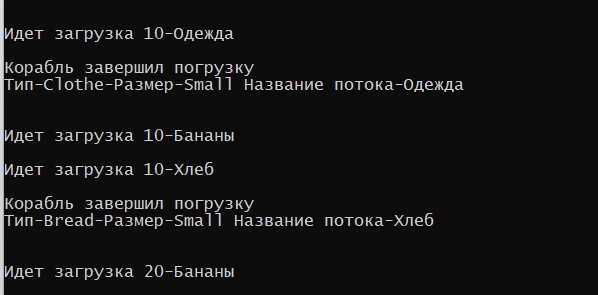
“Корабль прибыл в тоннель”-сгенерированный корабль не смог попасть в тоннель так как он был заполнен.



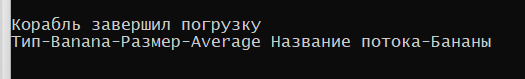
“Корабль отправился к причалу”-корабль отправился к нужному причалу для загрузки товаром.



“Идет загрузка”-корабль приплывший к причалу начал пополняться припасами



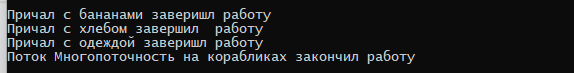
“Корабль завершил погрузку”-корабль полностью заполнен припасами и готов к отправке



“Генератор закончил работу”-необходимое количество кораблей было сгенерировано и генератор завершает свою работу.



Все корабли были заполнены и причалы завершают свою работу.



**Winform приложение**

### Руководство программиста

#### *Цикл работы программы*

#### *Библиотеки, используемые в проекте*

* using System.Collections.Generic
* using System.Threading
* using System

## Заключение