Проект 12: Склад

*Изготвил:* Ивелин Илиев, I курс,

спец: Информационни системи, Факултетен № 71958

Съдържание

[Съдържание 2](#_Toc43060236)

[1. Увод 3](#_Toc43060237)

[1.1. Описание и идея на проекта 3](#_Toc43060238)

[1.2 Цел и задачи на разработката 3](#_Toc43060239)

[2. Преглед на предметната област 4](#_Toc43060240)

[2.1 Основни дефиниции, концепции и алгоритми, които ще бъдат използвани 4](#_Toc43060241)

[2.2 Дефиниране на проблеми и сложност на поставената задача 5](#_Toc43060242)

[2.3 Подходи, методи (евентуално модели и стандарти) за решаване на поставените проблемите 5](#_Toc43060243)

[2.4 Потребителски (функционални) изисквания (права, роли, статуси, диаграми, ...) и качествени (нефункционални) изисквания (скалируемост, поддръжка, ...) 6](#_Toc43060244)

[3. Проектиране 7](#_Toc43060247)

[3.1 Обща архитектура – ООП дизайн 7](#_Toc43060249)

[4. Реализация, тестване 8](#_Toc43060250)

[4.1 Реализация на продуктовите класове (включва важни моменти от реализацията на класовете и малки фрагменти от кода) 8](#_Toc43060252)

[4.2 Управление на паметта и алгоритми. Оптимизации. 8](#_Toc43060261)

[4.3 Планиране, описание и създаване на тестови сценарии (създаване на примери) 9](#_Toc43060262)

[5. Заключение 18](#_Toc43060263)

[5.1 Обобщение на изпълнението на началните цели 18](#_Toc43060265)

[5.2 Насоки за бъдещо развитие и усъвършенстване 19](#_Toc43060266)

[6. Използвана литература 19](#_Toc43060267)

# Увод

* 1. Описание и идея на проекта

Да се напише програма, която да реализира информационна система, обслужваща склад. Програмата съхранява и обработва данните за складовите наличности във файл.

**За всеки продукт се съхранява следната информация:**

* име (описание на продукта, символен низ с произволна дължина)
* срок на годност
* дата на постъпване в склада
* име на производител
* мерна единица (килограми, литри)
* налично количество
* местоположение (секция/рафт/номер)
* коментар (свободен текст)

**Бонуси:**

* по въведен продукт и неговата цена, за брой или съответно килограм, се пресмятат загубите за зададен от потребителя период (бракуваното количество продукт със изтекъл срок на годност)

1.2 Цел и задачи на разработката

Целта на програмата е да може да оперира с един или повече файлове и да може да извършва следните видове общи операции (open, close, save, save as, help и exit). След като отвори даден файл, програмата може да извършва посочените в *Таблица 1* операции:

|  |  |
| --- | --- |
| **Print** | Извежда информация за наличните продукти в склада |
| **Add** | Добавя нов продукт в склада в диалогов режим, като въвежда всички данни за продукта и ги проверява за коректност. При добавянето на продукта се спазват следните правила:   * ако нов продукт е с различен срок на годност от вече съществуващ едноименен продукт, той да бъде поставен на различно място * ако имате достатъчно място, еднакви продукти с един и същи срок на годност да бъдат поставени на едно и също място * при извеждане на списъка с налични продукти да се изведе общото количество на едноименните продукти, независимо от срока им на годност |
| **Remove** | Изважда продукт от склада в диалогов режим:   * по дадено име и количество изважда съответните продукти от склада и извежда информацията за продукта и къде се е намирал * при наличие на повече от една партида, първо намалява тази със най-скоро изтичащ срок на годност, тогава във информацията за извършеното действие се отбелязва количеството и мястото на всяка от партидите, които сме намалили * в случай на опит за изваждане на повече от наличното да се дава информация на потребителя за наличността на продукта и срока на годност на партидите и възможност да реши дали все пак не иска да извади това което е останало |
| **Log <from> <to>** | Извежда справка за всички промени в наличността в периода от дата <from> до дата <to>, включително зареждания и извеждания на стоки. |
| **Clean** | Разчиства склада от всички стоки, на които е изтекъл или предстои скоро да изтече срока на годност, като извежда информация за разчистените стоки |

# Преглед на предметната област

2.1 Основни дефиниции, концепции и алгоритми, които ще бъдат използвани

Програмата е реализирана на Visual Studio 2019 , C++. Концепцията, за реализацията на информационната система за склад е обвързана с основната цел и изисквания заложени в заданието. Информационната система извежда следните видове информация по критерии зададени от потребителя:

* информация за моментната складовата наличност
* информация за датите и количествата на постъпилите в склада стоки за предишни периоди от време
* изписването на стоки
* бракуването на стоки
* изчисляване на загубите от неизписани бракувани стоки поради изтекъл срок на годност

Складът се състои от 50 на брой секции разделени поравно за различните видове продукти с различни мерни единици (литри и кг). Всеки вид има по 25 секции. Всяка секция се състои от 6 на брой рафта, на които има по 4 отделени клетки за складиране на продукти. Всяка клетка разполага с до 50 кг/л товароносимост. Всички еднакви продукти разполагат с до 2 рафта максимално или до 400 кг/л максимално количество, което може да заеме в склада.

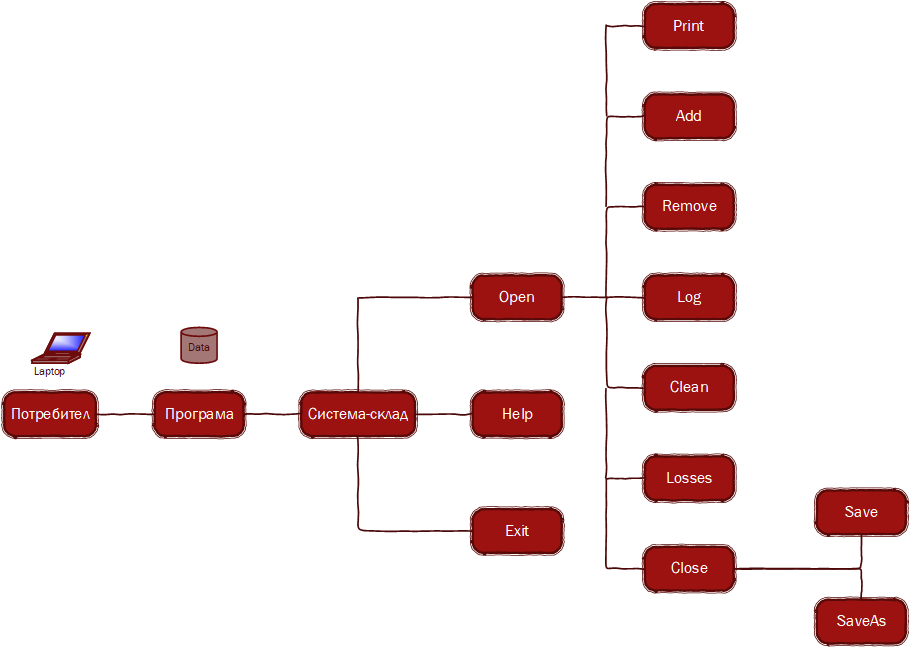
2.2 Дефиниране на проблеми и сложност на поставената задача

Сложността при създаването на програмата бе намирането на максимално най -добре работещото решение съгласно спецификацията на проекта и изграждането на концепция за реализацията на това решение. Срещнатите проблеми, бяха при уточняването на специфичните критерии, по които се извършва подредбата на продуктите с цел да не се допусне смесване на артикули от различен вид, различни партиди, дати на заприхождаване и срок на годност.

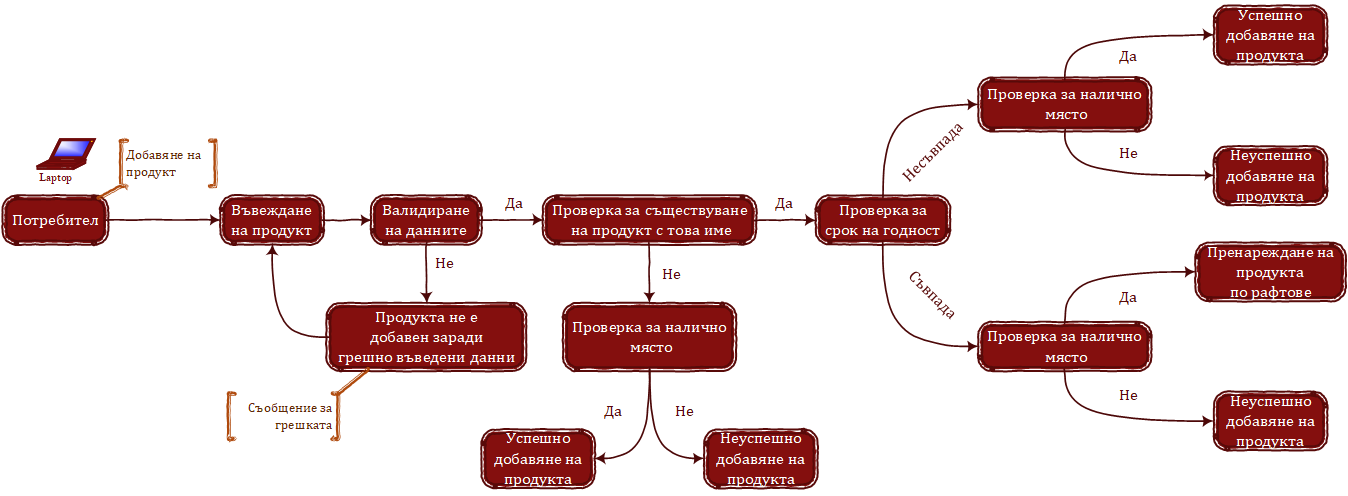
2.3 Подходи, методи (евентуално модели и стандарти) за решаване на поставените проблемите

Направен бе анализ на конкретната задача и бяха разгледани много различни подходи за намиране на решението ѝ. След взимане под внимание на всички плюсове и минуси на разгледаните различни подходи бе избран следният, а именно – създаване на модел на склад във следния вид – брой секции в склада, брой рафтове в секция и брой клетки в рафт и внедряването на даден продукт в клас. След като бе създадено по горе изброеното започна добавянето на функционалностите, като са обхванати всички потенциални сценарии за използване на всяка една от тях – добавяне и премахване на продукт в склада, бракуване и отстраняване на стоките с изтекъл срок на годност, както и създаването на записи при всяка една промяна в наличността. Представено е и конкретно решение на бонус задачата - пресмятане на загубите от бракуването на количество от даден продукт за зададен период от време.

2.4 Потребителски (функционални) изисквания (права, роли, статуси, диаграми, ...) и качествени (нефункционални) изисквания (скалируемост, поддръжка, ...)

Потребителските (функционални) изисквания към информационната система (*Фигура* *1*) са разработени максимално приятелски насочени с цел максимално улесняване на работата с приложението, както и пълното използване на функционалността му от потребителите.

Фигура

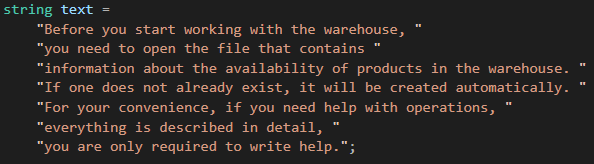
На следващата схема (*Фигура 2*) е показана детайлно функционалността на операция „add”, което представлява добавяне на продукт в склада:



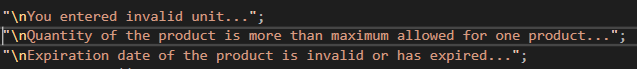


# Проектиране

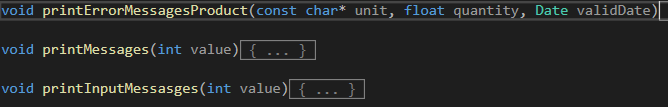
1. 1. Обща архитектура – ООП дизайн

Архитектурата на програмата е внимателно създадена, като е обърнато специално внимание на всеки детайл. Създадени са необходимите продуктови класове за поддържането на функционалността, а изгледа е създаден така че да улесни максимално работата на потребителя. (*Фрагмент 1*)

Фрагмент

Предвидени са и ситуациите, в които е възможно да бъде допусната потребителска грешка от неволно въведени некоректни данни или друга неволна грешка, в такива случаи, програмата извежда на екран подходящото съобщение и от съображение за удобството на потребителя, въвеждането на данните не започва от самото начало, а от мястото на сгрешено въведената компонента. (*Фрагмент 2*)

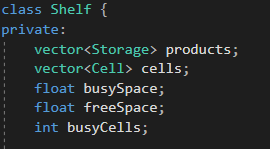
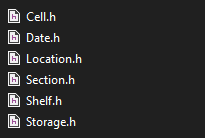
Фрагмент 2

Взети и са в предвид и грешките, които евентуално биха възникнали в информационната система при работа с файловете, тъй като информацията трябва да бъде защитена и да се записва винаги точно и на правилното място в тях. (*Фрагмент 3*)

Фрагмент 3

# Реализация, тестване

1. 1. Реализация на продуктовите класове (включва важни моменти от реализацията на класовете и малки фрагменти от кода)

Създадените класове, както се разбира от техните имена, са частите и компонентите от тези части, които задължително един склад трябва да притежава – секции, рафтове, клетки и продуктите разположени в тях, като продуктите са най-важните сред тях тъй като склада трябва винаги да е заредени с необходимите стоки с цел улесняване работата му с неговите клиенти. Останалите класове – дата и локация, чрез тях информационната система определя датата и локацията за намиране на даден продукт в склада.

Клас Shelf

Класове

Това е малък фрагмент от клас Рафт представящ член данните му. Използван е STL контейнера вектор, в който се записват продуктите намиращи се на този рафт. По количеството на съдържащите се продукти във вектора се определя заетото място на този рафт и останалото свободно място на него. Чрез направените изчисления се използва вектор от клетки фиксирани за един рафт, записващ вътре в себе си заетото и свободно място в клетката, спомагащи за правилното определяне на локацията на даден продукт.

5. 2. Управление на паметта и алгоритми. Оптимизации.

Всеки обект използващ динамична памет се контролира от информационната система и се следи каква част от паметта заема и се освобождава при приключване на работа с дадения обект (вектори, масиви или собствен обект). Има възможност за оптимизация във връзка с управлението на паметта.

* 1. Планиране, описание и създаване на тестови сценарии (създаване на примери)

За успешното внедряване на системата и въвеждането и в експлоатация от потребителя е необходимо да се планират стъпките и дейностите в етапите на тестване и внедряване. Крайният сорс код ще се окомплектова заедно с помощните материали и публикува в GitHub, като същевременно е спазвано и качването на междинните разработки.

За преглед - [*https://github.com/iiliev02/Univercity/tree/master/Project/Storage%20Project*](https://github.com/iiliev02/Univercity/tree/master/Project/Storage%20Project)

С цел проверка на постигнатата реализация на приложението с критериите по задание, е необходимо да се изпълняват минимум следните функционални проби посочени в (*Таблица 2*).Още по времето на създаване на всяка една функционалност са направени нужните тестове, за да се потвърди тяхната коректна работа. По време на извършването на тези частични функционални тестове възникнаха още нови и нови идеи, които с въвеждането им спомогнаха за изпълнение и максимално най-добре работещото решение.



|  |  |  |
| --- | --- | --- |
| ***Функция*** | ***Описание*** | ***Резултат (Да / Не)*** |
| Print | Извежда информация за наличните продукти в склада. | Да |
| Add | Добавяне на продукт:   * На нов, не съществуващ в склада. * След проверка за съществуващ вече но с различен срок на годност едноименен продукт. * След проверка за съществуващ вече но с еднакъв срок на годност , пренареждане на рафтовете за добавяне на еднаквите продукти един до друг. | Да |
| Remove | Извеждане на продукт:   * По дадено име и количество премахва продукта, но при наличие на повече от една партида премахва даденото количество от потребителя от продукта с най – малък срок на годност. * При опит за премахване на повече от наличното количество на продукта, извежда запитване до потребителя, дали иска да премахне цялото налично количество на продукта или продуктите. | Да |
| Log | По зададен от потребителя период от време извежда всички направени промени в склада. | Да |
| Clean | Изчиства всички продукти с изтекъл срок на годност или предстои да изтече. | Да |
| Losses | По зададено име, цена за кг или литър на продукта и период от време от потребителя извежда всички загуби за този продукт. | Да |

По-долу са представени няколко примера за работата на приложението, покривайки основните функционалности, които са изисквани от заданието. Входа от потребителя е означен с долар ($) в примерите.

***Пример 1:***

*Before you start working with the warehouse, you need to open the file that*

*contains information about the availability of products in the warehouse. If*

*one does not already exist, it will be created automatically. For your*

*convenience, if you need help with operations, everything is described in*

*detail, you are only required to write help.*

*----- Menu ▼ -----*

*• Open <File Path>*

*• Help*

*• Exit*

*Your choice: $ Open C:\Storage.txt*

*Successfully opened Storage.txt*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Add*

*Еnter the product name: $ Jack Daniels*

*Enter the expiration date: $ 2020-15-15*

*Enter the producer: $ USA*

*Enter the unit: (kg, litre) $ Litre*

*Enter the available quantity: $ 200*

*Enter the comment of product: $ Whisky*

*Expiration date of the product is invalid or has expired...*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Print*

*\*\*\*\*\*\* Litre products \*\*\*\*\*\**

*------- Product 1 -------*

*Product Name: cola*

*Unit: litre*

*Available Quantity: 266*

*Location: 1 - [1-2] - [1-6]*

*------- Product 2 -------*

*Product Name: sprite*

*Unit: litre*

*Available Quantity: 381*

*Location: 1 - [3-4] - [1-8]*

*------- Product 3 -------*

*Product Name: fanta*

*Unit: litre*

*Available Quantity: 299*

*Location: 1 - [5-6] - [1-6]*

*------- Product 4 -------*

*Product Name: jack daniels*

*Unit: litre*

*Available Quantity: 300*

*Location: 2 - [1-2] - [1-6]*

*\*\*\*\*\*\* Kg products \*\*\*\*\*\**

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Add*

*Еnter the product name: $ Pepsi*

*Enter the expiration date: $ 2021-10-10*

*Enter the producer: $ Pepsi*

*Enter the unit: (kg, litre) $ Litre*

*Enter the available quantity: $ 242*

*Enter the comment of product: $ Drink*

*Successful added product*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: & Log*

*-------------------------*

*From: $ 2020-6-12*

*To: $ 2020-6-14*

*-------------------------*

*2020-6-12->add->cola->litre->200.000000*

*2020-6-12->add->sprite->litre->381.000000*

*2020-6-13->add->fanta->litre->299.000000*

*2020-6-13->add->jack daniels->litre->400.000000*

*2020-6-13->remove->jack daniels->litre->100.000000*

*2020-6-14->add->pepsi->litre->242.000000*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Close*

*-------------------------*

*Do you want to save: [Yes / No] $ Yes*

*-------------------------*

*Choose one of the two options:*

*1) Save*

*2) SaveAs <File Path>*

*Your choice: $ Save*

*Successfully saved Storage.txt*

***Пример 2:***

*----- Menu ▼ -----*

*• Open <File Path>*

*• Help*

*• Exit*

*Your choice: $ Open*

*You must enter a file path!*

*Not successfully opened storage!*

*----- Menu ▼ -----*

*• Open <File Path>*

*• Help*

*• Exit*

*Your choice: $ Open C:\Storage.txt*

*Successfully opened Storage.txt*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Add*

*Еnter the product name: $ Tomatoes*

*Enter the expiration date: $ 2020-12-21*

*Enter the producer: $ Bulgaria*

*Enter the unit: (kg, litre) $ Kg*

*Enter the available quantity: $ 350*

*Enter the comment of product: $ Vegetables*

*Successful added product*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Add*

*Еnter the product name: $ Cucumber*

*Enter the expiration date: $ 2020-6-19*

*Enter the producer: $ Bulgaria*

*Enter the unit: (kg, litre) $ Kg*

*Enter the available quantity: $ 166*

*Enter the comment of product: $ Vegetables*

*Successful added product*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Remove*

*-------------------------*

*Еnter the product name: $ Tomatoes*

*Enter the quantity you want to remove: $ 150*

*-------------------------*

*Part of the product batch tomatoes with quantity 150 has been removed with an expiration date 2020-12-21 located on 1 - [1-2] - [1-7]*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Clean*

*\*\*\*\*\*\* Litre products \*\*\*\*\*\**

*There are no products with measure Litre for cleaning!*

*\*\*\*\*\*\* Kg products \*\*\*\*\*\**

*------- Product 1 -------*

*Product Name: cucumber*

*Unit: kg*

*Available Quantity: 166*

*Location: 1 - [3] - [1-4]*

*Choose one of the seven options:*

*1) Print*

*2) Add*

*3) Remove*

*4) Log*

*5) Clean*

*6) Losses*

*7) Close*

*Your choice: $ Close*

*-------------------------*

*Do you want to save: [Yes / No] $ Yes*

*-------------------------*

*Choose one of the two options:*

*1) Save*

*2) SaveAs <File Path>*

*Your choice: $ SaveAs C:\Another Storage.txt*

*Successfully saved Another Storage.txt*

# Заключение

1. 1. Обобщение на изпълнението на началните цели

Като цяло, проекта беше интересен за мен. Трябваше да се запозная с подобни типове програми за складове как работят, задачите които изпълняват и дали това удовлетворява самите потребители, който експлоатират такъв тип програмни продукти. Това ми помогна много да съставя своя версия и да я реализирам с помощта на досегашните ми знания. Лично аз съм удовлетворен от своята работа, работих върху различни свои и чужди идеи, но в крайна сметка в представената от мен информационна система успях да реализирам най-добрите от тях, постигайки стабилна работа и коректно изпълняване на поставените цели.

* 1. Насоки за бъдещо развитие и усъвършенстване

Има възможност за бъдещи оптимизации за усъвършенстване по кодовото изпълнение, за съкращаване на дадени функции с цел по оптимално решение за тяхната функционалност, както и за бъдещата дигитална визуализация на информационната система.

# Използвана литература

*Използван алгоритъм за подравняване двустранно на началния текст и взет код за клас TextAling.h :*

* [*https://gist.github.com/Niceblack/3ccc3b212ccf61629c664e7f45e39e26*](https://gist.github.com/Niceblack/3ccc3b212ccf61629c664e7f45e39e26)

*Използван софтуер за изработване на диаграмите:*

* *Microsoft Visio 2013 Pro*