Картина, която съдържа текст, графична колекция

Описанието е генерирано автоматичноТЕХНИЧЕСКИ УНИВЕРСИТЕТ – ВАРНА

Факултет по изчислителна техника и автоматизация

Катедра „Софтуерни и интернет технологии“

**СЕМЕСТРИАЛНА ДОМАШНА РАБОТА**

по дисциплината „Базово програмиране”

на тема: „Самолетни полети”

Вариант 31

|  |  |
| --- | --- |
| **Изготвил:**  Александър Станков | Проверил: |
| **Специалност**: СИТ |  |
| **Група: 2** |  |
| **Факултетен номер:** 21621501 |  |

2021

Съдържание

[I. ￼Задание на проекта 2](#_Toc1414568425)

[II. Анализ на решението 3](#_Toc739902258)

[1. Структура за данните в програмата 4](#_Toc975803039)

[2. Реализация на условие A 4](#_Toc1442588197)

[3. Реализация на условие B 4](#_Toc1913169044)

[4. Реализация на условие C 5](#_Toc1085714576)

[5. Реализация на условие D 6](#_Toc1650132496)

[6. Реализация на условие E 6](#_Toc1286823799)

[7. Реализация на условие F 7](#_Toc1355411359)

[8. Реализация на условие G 7](#_Toc417272584)

[9. Реализация на условие H (по желание) 8](#_Toc515297786)

[III. Упътване за употреба 8](#_Toc136992244)

[1. Впишете съответната част от проекта 9](#_Toc1275951274)

[2. Впишете съответната част от проекта 9](#_Toc1301671379)

[IV. Примерно действие на програмата 9](#_Toc1838932237)

[1. Условие A 10](#_Toc2014635896)

[2. Условие B 10](#_Toc1559189330)

[3. Условие C 10](#_Toc597706951)

[4. Условие D 10](#_Toc756462065)

[5. Условие E 10](#_Toc1177766447)

[6. Условие F 10](#_Toc1695348583)

[7. Условие G 11](#_Toc2106656535)

[8. Условие H (по желание) 11](#_Toc541278323)

Задание на проекта

**Базова задача – сложност** **ниска**

1. Меню за избор на функциите в програмата.
2. Добавяне на полети.
3. Добавяне по един полет
4. Добавяне на списък с полети. Въвежда се цяло число n и след него n на брой полети.
5. Извеждане на всички полети на екрана.
6. Извеждане на полети с най-ниска цена за първа класа
7. Извеждане на полети на даден пилот

**Допълнение първо – сложност средна (+ базова задача)**

1. Закупуване на билет за полет
2. При закупуване на билет, се избира полет и се въвеждат имената на пътника. Така всеки полет е попълнен с имена на пътници.
3. Въвежда се дестинация и дата на заминаване, извежда се номер на полет с цена за първа и втора класа. Ако до полета има повече от 20 дни, цената на билета е 50% от реалната цена, ако има по малко от 20 дни - цената е 75% от реалната цена, ако остават по малко от 5 дни, цената на билета е 100% от реалната цена, ако остават по малко от 24 часа цената на билета е с 20% по висока от реалната цена. (Приема се че един месец е 30 дни)

**Допълнение второ – сложност висока (+ базова задача + допълнение първо)**

Отказ от полет: По номер на полет и име на пътник, пътникът се отказва от полет, мястото му се освобождава и се възстановяват, 100% от цената ако билета е върнат 20 дни преди полета, 75% ако е върнат по малко от 20 дни преди полета, 50% ако е върнат 5 дни преди заминаване на полета (Извежда се сумата за връщане)

**Допълнение трето – сложност висока (+ базова задача + допълнение първо + допълнение второ)**

1. Справки за полетите с под меню
2. Извеждане на всички полети в подреден ред по:
3. Ден на заминаване на полетите;
4. Общ брой места от първа и втора класа.
5. Извеждане на полета с най-малко закупени билети;
6. Извеждане на полета с най-висока цена на билетите в първа класа.
7. Данните в програмата да могат да се запазват във файл между две стартирания на програмата.

**Допълнение четвърто – (за допълнителни точки)**

1. За точка B при добавянето на нов полет да се даде възможност за избор дали да се въведе нов или да се прекъсне въвеждането.
2. За точка C да се реализира извеждането на полетите по един на страница със списък от местата, за заетите да изписва името на пътника, за незаетите, да изписва че са в продажба.
3. За точка G да се записват данните в двоичен файл

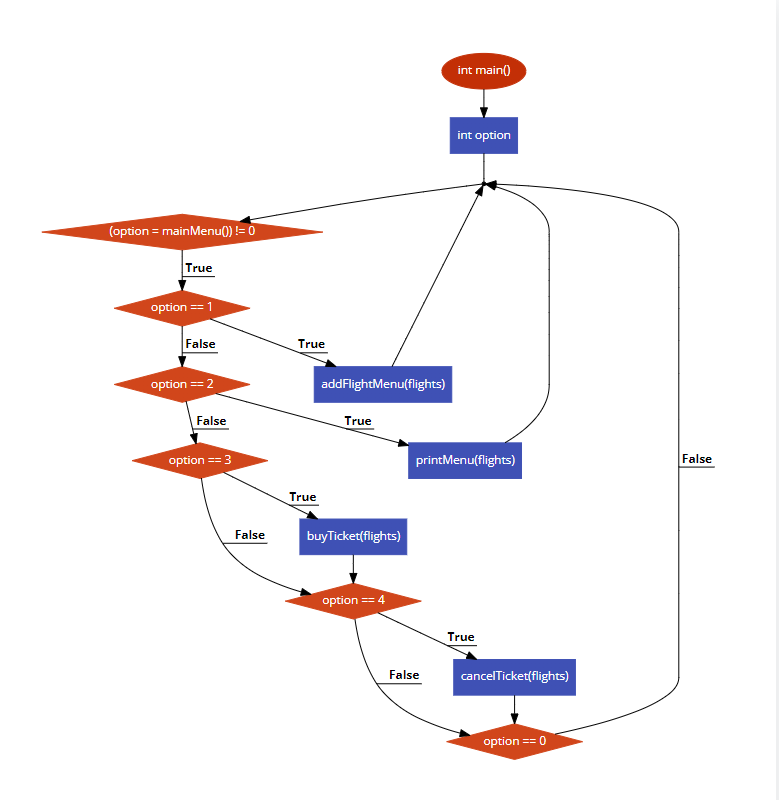
Анализ на решението

Структура за данните в програмата

Реализация на условие A (базова задача)

Анализ на алгоритъма, който трябва да се реализира

Меню за избор на влизане в подменюта на различни функции. Чрез въвеждане на число то 0 до 4 ще ви отведе в съответното подменю допълнителна функция

Блок схема на алгоритъма

Т

Функция с която е реализиран алгоритъма

int option;

while ((option = mainMenu()) != 0) {

if (option == 1) {

addFlightMenu(flights);

continue;

}

if (option == 2)

{

printMenu(flights);

continue;

}

if (option == 3)

{

buyTicket(flights);

}

if (option == 4) {

cancelTicket(flights);

}

if (option == 0)

{

break;

}

}

#### Входни данни на функцията

„1“ – Влизане в подменю където се въвеждат данни от структурата за запис на нов полет.

„2“ – Влизане в подменю където може да се достъпят функции за изписване на полети под определени критерии.

„3“ – Изпълнение на функция за закопуване на билет за специфичен полет.

„4“ – Отказване на билет и възвръщане на сума изчислена под определен критерии.

#### Изходни данни на функцията или данни, които се извеждат

Всички функции на програмата подпомагат потребителя с извеждане на инструкции с очакване на точно какви данни се изискват от него. При въвеждане на данни, които са от грешен тип или не точно попълнени, се изписва грешка на съответното неизпълнено действие.

Реализация на условие C (базова задача)

Анализ на алгоритъма, който трябва да се реализира

Алгоритъма трябва да реализира функцията за добавяне на полет в базата данни следвайки неговата структура. Потребителят трябва да въведе изискващите се от него данни в конзолата за да може да се постигне желания резултат.

Блок схема на алгоритъма

Функция с която е реализиран алгоритъма

Чрез навигиране през подменютата на програмата може да се изпълни функцията за добавяне на полет, която следва структурата на му. Изписват се съобщения от конзолата за това какви данни трябва да се въведат в нея с цел запаметяването им в базата данни.

#### Входни данни на функцията

Входните данни трябва да бъдат въведени в строга последователност:

1. Номер на полета
2. Дестинация на полета
3. Име на пилот
4. Цена на билет за първа класа
5. Цена на билет за втора класа
6. Ден на излитане на полета

#### Изходни данни на функцията или данни, които се извеждат

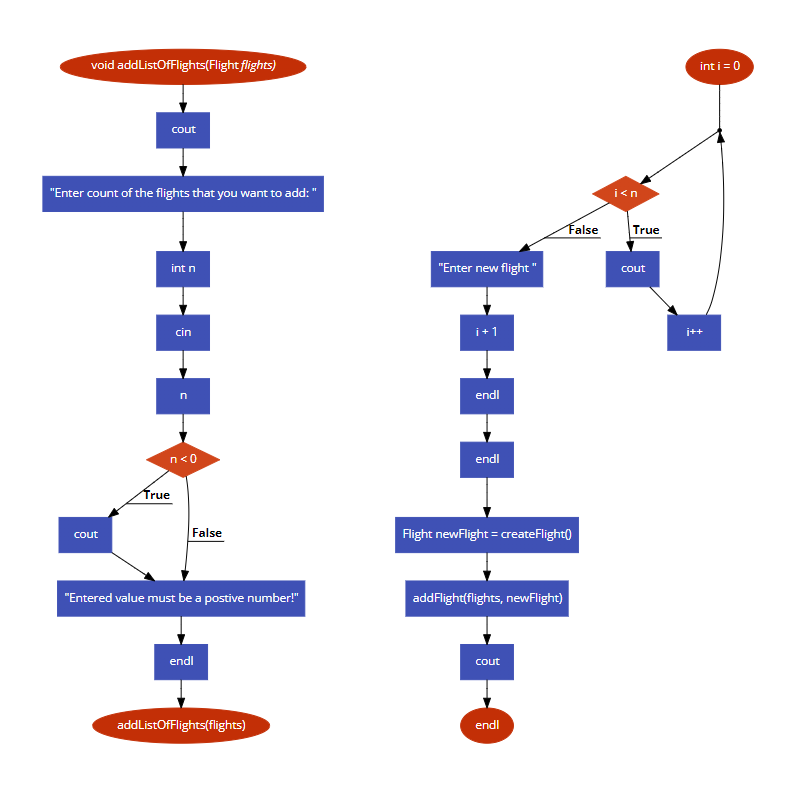
Тази функция не изписва никакви изходни данни. Тя вмъква нов полет с пълната му структура в базата данни.

Реализация на условие D (базова задача)

Анализ на алгоритъма, който трябва да се реализира

Алгоритъма трябва да може да изпълнява пълната функционалност на условие C, както и да може да го прави толкова пъти, колкото потребителя изисква с нововъведени данни.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Функцията представлява, както следва, изчакване за вход на данни за това колко пъти иска потребителя да изпълни операцията. Функцията притежава if проверка за въвеждане на невалидни данни. Същността на фунцията е последния for цикъл, който извиква в себе си 2 функции за създаване на полети (createFlight, AddFlight) и се повтаря толкова пъти колкото потребителя е изписал в началото на функцията.

#### Входни данни на функцията

Първо се въвежда цяло число, което служи за обозначаване на това колко структури на полети искаме да въведем. След това данните са същите, както в подточка 2.3.3.1.

#### Изходни данни на функцията или данни, които се извеждат

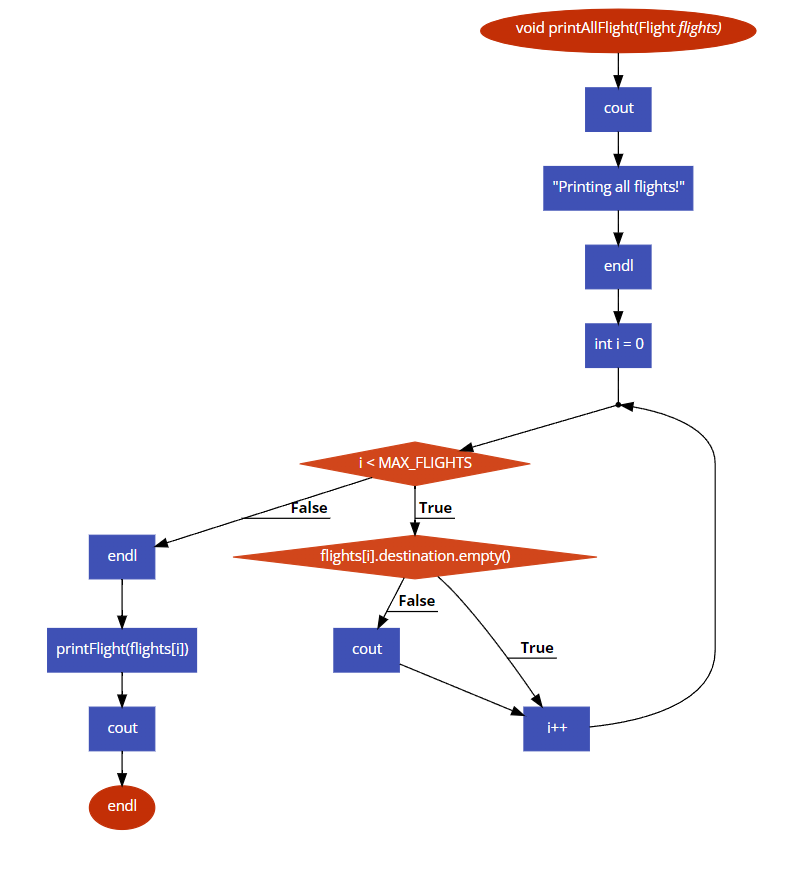
Тази функция не извежда данни. Тя записва всички нови структури на полети в базата данни.

Реализация на условие E (базова задача)

Анализ на алгоритъма, който трябва да се реализира

Трябва да се достъпят всички записи на структури на полети присъстващи в програмата и да бъдат изписани с наличните им параметри.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Използвани са 2 функции за реализиране на този алгоритъм. Една главна, която служи за изписване на всички параметри от структурата на полет и втора, която я извиква, толкова пъти, колкото структури са записани с пълните си параметри.

#### Входни данни на функцията

Тази функция няма точно определни входни данни, защото се достъпва от подменюта на програмата. Тя може да бъде изпълнена по следния начин:

1. Изпълнение на програмата
2. Въвеждане на цялото число „2“
3. Въвеждане на цялото число „1“

#### Изходни данни на функцията или данни, които се извеждат

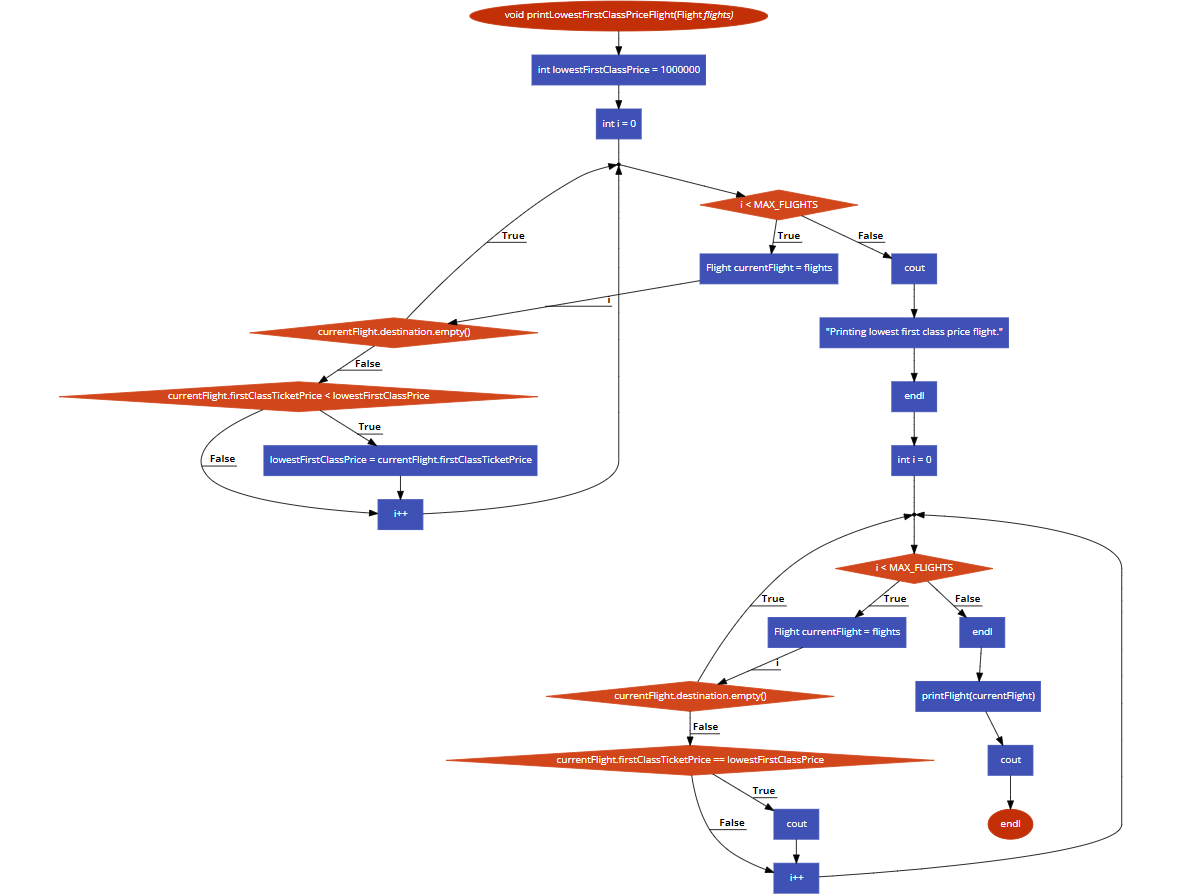
Изходните данни на функцията представляват изписване на всички параметри на структурата на полет, както и повтарянето на това действие за всеки един уникален полет записан в базата данни.

Реализация на условие F (базова задача)

Анализ на алгоритъма, който трябва да се реализира

Трябва да бъдат изписани всички полети, които са записани в базата данни на програмата по определен ред следващ критерии за най-ниска стойност на параметър от структурата на всеки един полет.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Функцията съдържа свой изолиран int, който служи за записване на най-ниската срещната стойност на **цената на билета за първа класа** при обхождане на базата данни. Следва цикъл, който обхожда всички структури и сравнява цените и записва най-ниската в излирания int. Следва втори цикъл, който обхожда втори път всички структури за да намери тази, която отговаря на стойността на изолирания int и се извиква още една функция за изписване на параметрите на полета.

#### Входни данни на функцията

Тази функция се достига с навигиране чрез подменюта от програмата. За да се достъпи следва потребителя да въведе следното:

1. Въвеждане на цялото число „2“
2. Въвеждане на цялото число „2“

#### Изходни данни на функцията или данни, които се извеждат

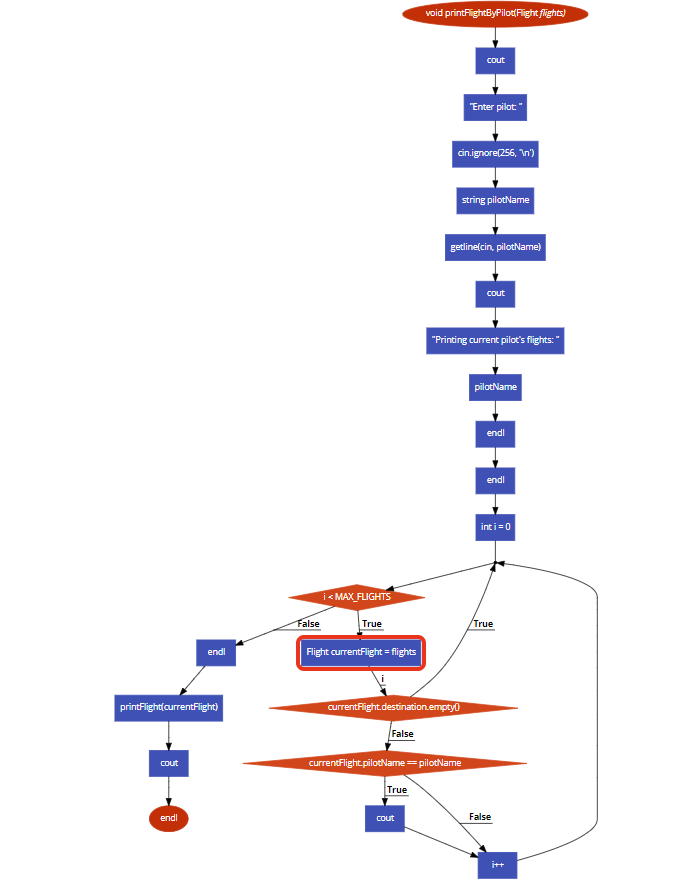
Изходните данни на фунцията са изписването на полета с най-ниска стойност на цената на билета за първа класа.

Реализация на условие G (Базова задача)

Анализ на алгоритъма, който трябва да се реализира

Фунцията трябва да изписва всички полети записани с един и същ пилот.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Фунцията очаква да и бъде въведен низ с име на пилот за да може да работи цикъла, който обхожда всички записи на структурата на полет и проверява дали параметъра ”pilotName” съвпада с въведеният от потребителя низ. При съответствие се извиква функцията за изписване на структурата на съответния полет, цикъла се повтаря по цялата база данни и извежда всички полети съответстващи на низа.

#### Входни данни на функцията

Тази функция приема само низ въведен от потребителя за да бъде зададен критерия, по който да търси цикъла във функцията.

#### Изходни данни на функцията или данни, които се извеждат

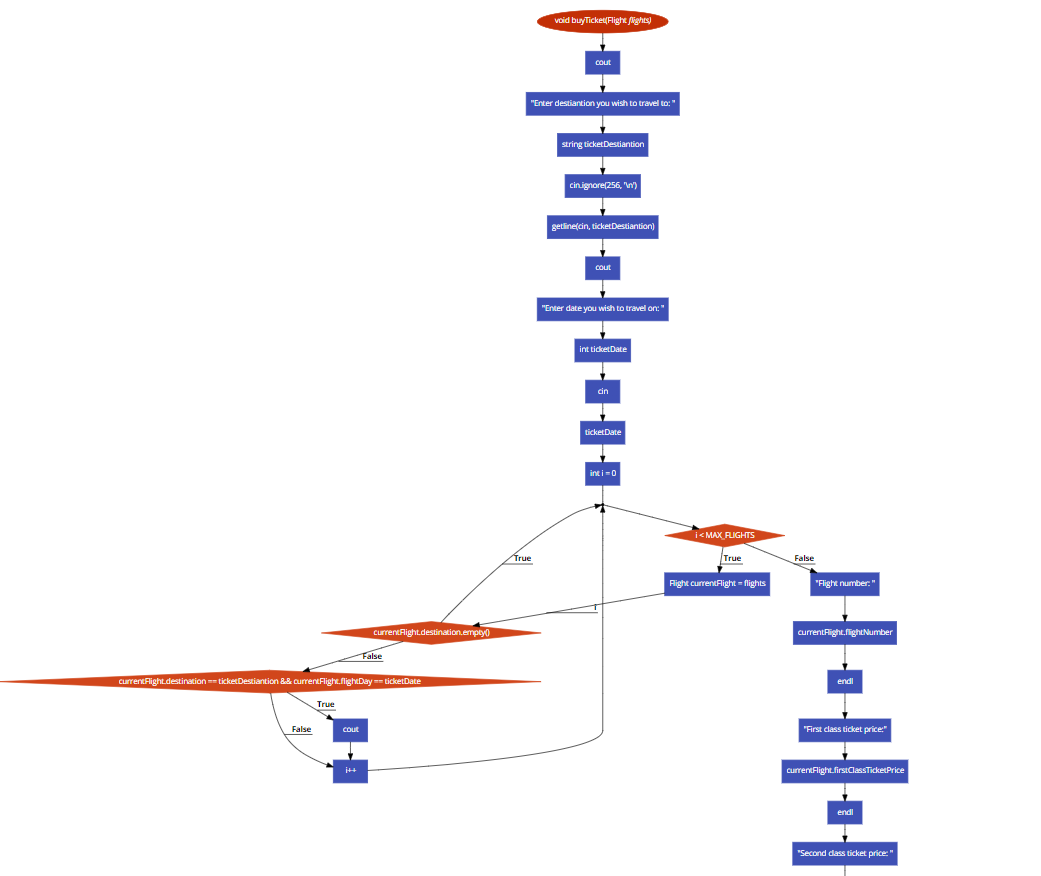
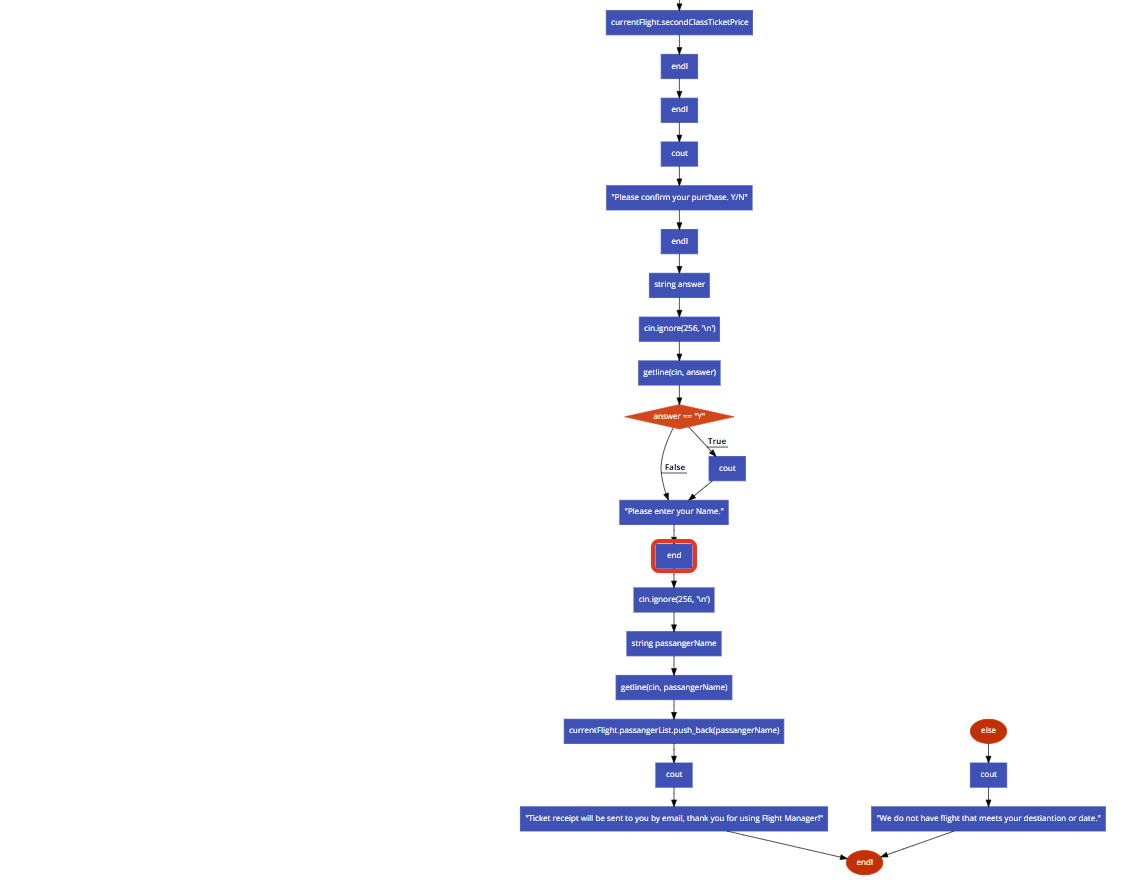
Изходните данни на фунцията са изписването на всички полети които отговарят на низа въведен от потребителя.

Реализация на условие А, B (**сложност средна**)

Анализ на алгоритъма, който трябва да се реализира

Потребителят трябва да може да закупи билет за полет, който съществува, има свободни места и съвпада с изискванията му за дестинация и ден на излитане.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Фунцкията приема данни за дестинация и заминаване , по които да търси съответстващ полет. Използва се цикъл за да се обходи базата даннии и се изписват цените и номера на полета. Прави се проверка за потвърждение на потребителя за закупуване на билет и след трябва да се въведът данни за името му. Следва вмъкване на имената му в лист за самия полет и изпращащо съобщение за успешна покупка. Ако не се намери полет за потребителя, програмата изписва съобщение и се връща в менюто.

#### Входни данни на функцията

Данните които са нужни на функцията първоначално са въведени от потребителя. Пътрво низ, който представлява дестинацията и след това цяло число, определящо желания ден за заминаване. Потвърждава се покупката с отговор, който е отново под форма на низ и последните данни са имената на този за когото е билета. Те са нужни за да може да се вмъкнат в листа с пасажери за определения полет.

#### Изходни данни на функцията или данни, които се извеждат

Функцията извежда съобщения навигиращи потребителя какъв тип данни очаква да получи и след като всички стъпки биват завършени, се изписва потвърждаващо съобщение за успешна поръчка на билет.

Реализация на условие C (**сложност средна**)

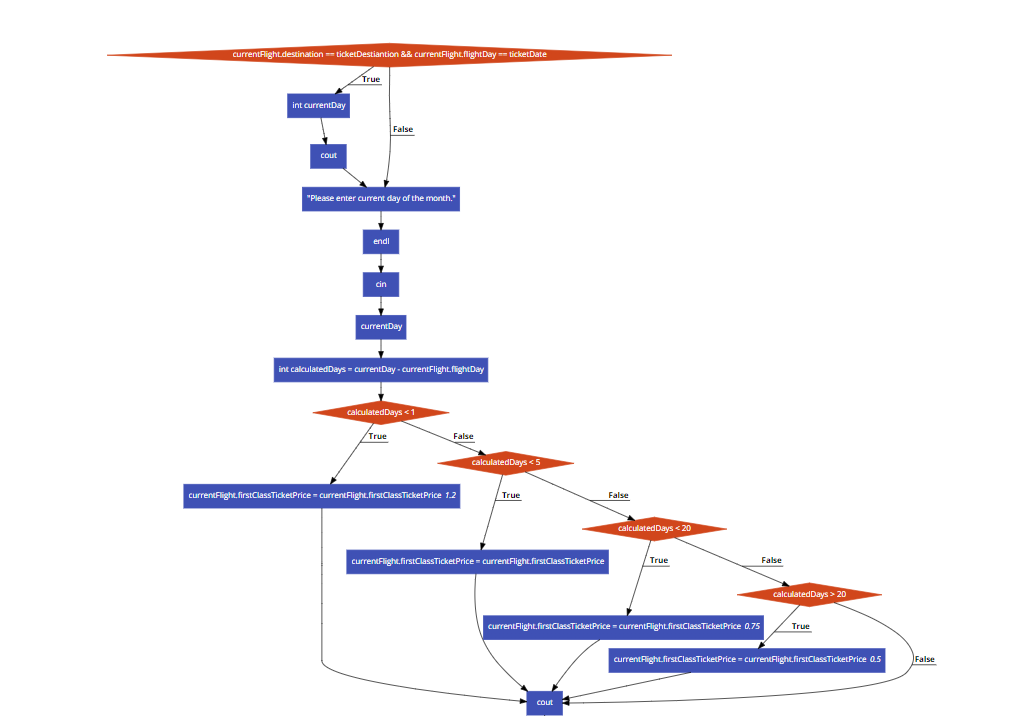
Анализ на алгоритъма, който трябва да се реализира

Алгоритъма трябва да пресмята цена на билетите според следните критерии:

1. Ако до полета има повече от 20 дни, цената на билета е 50% от реалната цена,
2. Ако има по малко от 20 дни - цената е 75% от реалната цена,
3. Ако остават по малко от 5 дни, цената на билета е 100% от реалната цена,
4. Ако остават по малко от 24 часа цената на билета е с 20% по висока от реалната цена.

Той може да се реализира, когато потребителя въведе днешната дата и с помощтта на функция се пресмята оставащите дни до излитане на полета и извеждането на коректни данни.

Блок схема на алгоритъма



Функция с която е реализиран алгоритъма

Операцията за пресмятане на правилните цени е вградена във функцията за купуване на билет от предната подточка. Данните които тя използва са въведен от потребителя ден, отговарящ на днешната дата и датата на излитане на полета за сравнение. След това се използват пресмятания след проверки със зададените критерии от подточка 2.9.1 и се изписват нужните резултати на цените за първа и втора класа.

#### Входни данни на функцията

Единственото, което използва тази операция е цяло число изписано от потребителя за да се запише днешна дата.

#### Изходни данни на функцията или данни, които се извеждат

Изходните данни са пресметнатите цени на полетите след пресмятането им в проверките за оставащи дни до полета.

Упътване за употреба

При стартиране на програмата на потребителят ще му бъде представено лесно за навигиране меню, което приема само цели числа за избирането на действия и влизане в подменюта или прекратяване на програмата.

Очаквани входни данни

Всеки път когато от потребителя бъде изискано да се впишат някакъв вид данни, програмата ще изпише точно и ясно какво очаква да приеме с улеснения на места където, ако се въведе невалиден тип или неправилно изписване на очаквания тип, ще се изпише съобщение с предположение каква е грешката.

Примерно действие на програмата

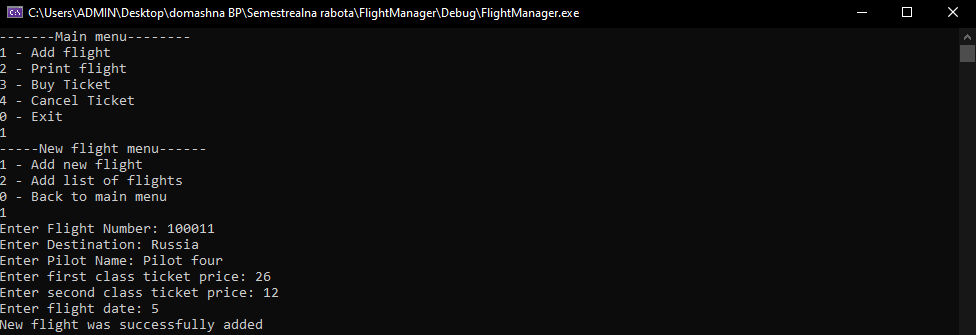
Условие A

Снимка на изгледа с примерни входни данни



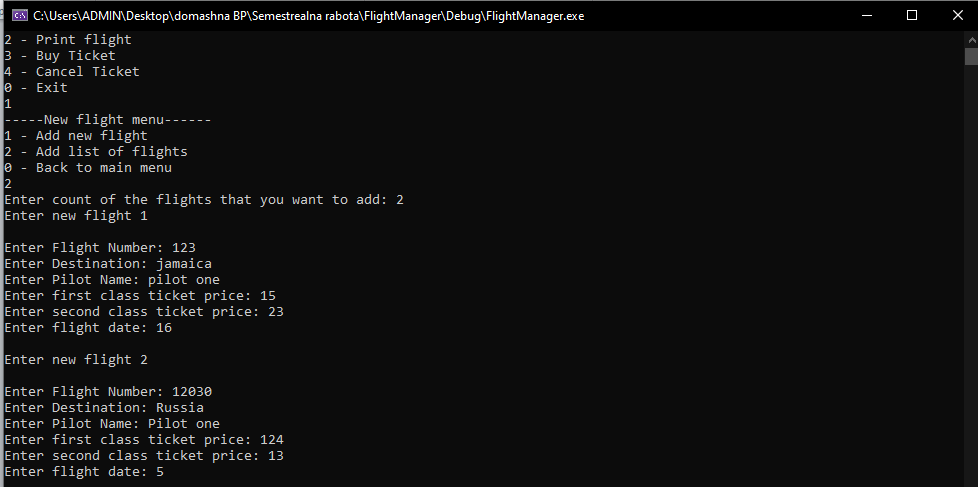
Условие B

Снимка на изгледа с примерни входни данни



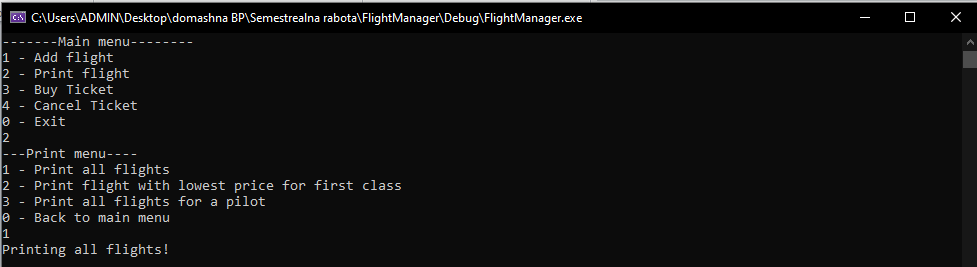
Условие C

Снимка на изгледа с примерни входни данни

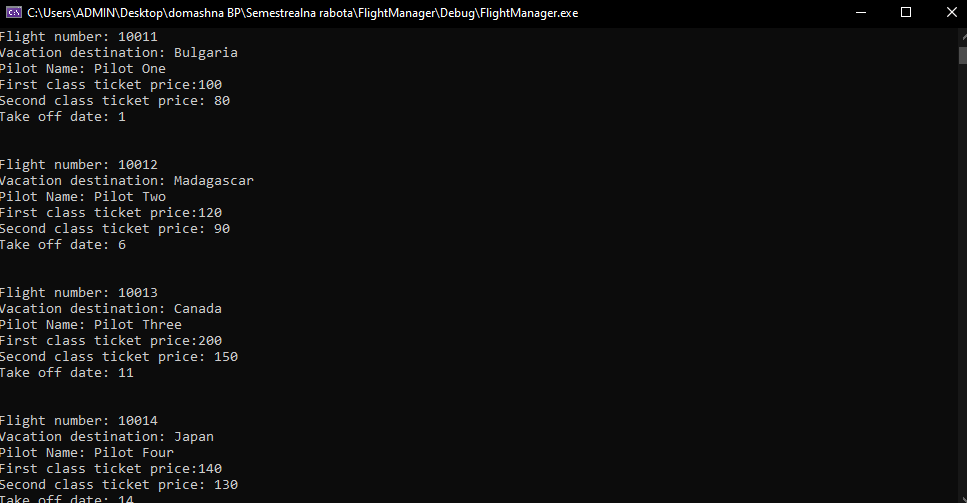


Условие D

Снимка на изгледа с примерни входни данни

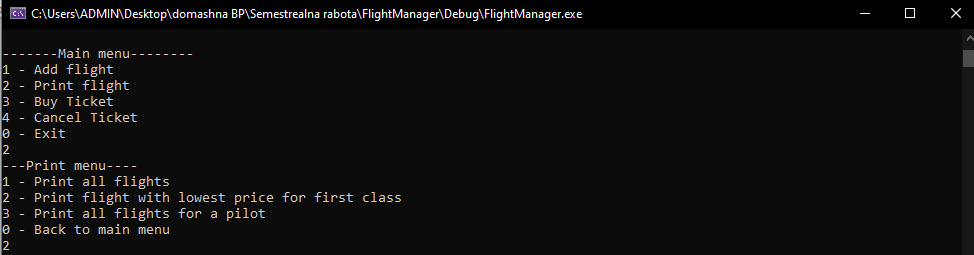


Снимка на изгледа с примерни изходни данни



Условие E

Снимка на изгледа с примерни входни данни

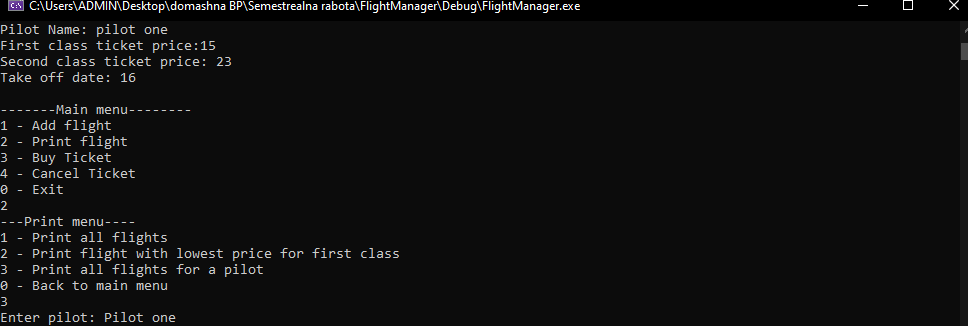


Снимка на изгледа с примерни изходни данни

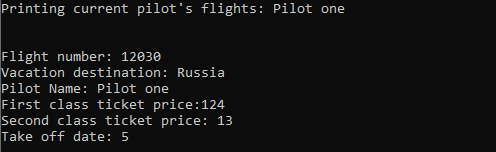


Условие F

Снимка на изгледа с примерни входни данни



Снимка на изгледа с примерни изходни данни



Условие A,B,C (**сложност средна**)

Снимка на изгледа с примерни изходни данни

