Технически Университет София

Курсова работа по СПр

Едуард Кирилов Кънчев

Група-91а, ФКСТ

Фак:381222013

Тема: „Каса на магазин“

1. Задача: “Продавач в магазин разполага с всяка сутрин с дадена сума пари - N, разделена на Nx0,1 лв., Nx0,2 лв., Nx0,5 лв., Nx1 лв., Nx2 лв., Nx5 лв., Nx10 лв., Nx20 лв., Nx50 лв., Nx100 лв. През деня клиентите пазаруват и оставят пари в касата, като не винаги имат точната сума и съответно продавачът връща ресто. При приключване на работния ден, продавачът отчита по колко броя от кои пари има. Напишете програма, която реализира горните действия.“
2. Анализ на задачата: Трябва да създадем приложение, което симулира работа на касов апарат в магазин. В началото на деня има определена сума пари разпределена на различни банкноти. През дена хората пазаруват като получават ресто при наличието на банкноти. В края на дена се прави отчет на парите в касата.
3. Функционално описание на приложението:
4. Стартиране на касата: при стартирането на програмата, касата трябва да бъде заредена с начални пари(което правим ние).
5. -Плащане на клиенти-: клиентите могат да плащат суми, които не винаги са точни, така че програмата трябва да изчислява дали има достатъчно пари в касата, за да върне ресто. Това изисква алгоритъм, който проверява какви видове да използва за връщане на остатъка.
6. -Връщане на ресто: ако клиента не плаща с точка сума, продавача трябва да върне ресто. Програмата трябва да намери начин да даде ресто с наличните монети и банкноти. Ако не може да върне точната сума, транзакцията трябва да се прекрати.
7. -Отчет на касата: в края на деня програмата трябва да може да отчете колко пари има в касата и колкото банкноти и монети са останали.
8. Изпълнение на функционалностите:
9. Инициализация на касата:

-Метод: def \_\_init\_\_(self, initial\_data):

-Описание: Инициализира касата с начални стойности за банкнотите и монетите от подадени данни.

-Параметри: Низ с данни за стойностите на банкнотите и монетите.

-Връщана стойност: Няма.

1. Добавяне на пари в касата:

-Метод: def add\_cash(self, amount):

-Описание: Добавя пари в касата и актуализира броя на съответните

Банкноти и монети.

-Параметри: Речник с банкноти и монети за добавяне.

-Връщана стойност: Няма.

1. Изчисляване на ресто:

-Метод:def give\_change(self, change\_amount):

-Описание: Връща ресто на клиента, като използва наличните банкноти и монети. Ако не може да върне ресто връща None;

-Параметри: Стойността на рестото.

- Връщана стойност: Речник с банкнотите и монетите за ресто или None

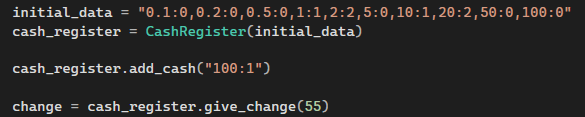
1. Отчитане на състоянието на касата

-Метод: def report(self):

-Описание: Връща текущото състояние на касата с броя на всяка банкнота и монета.

-Параметри: Няма.

-Връщана стойност: Речник с броя на всяка банкнота и монета в касата.

1. Експериментални данни:
2. Сценарии 1:

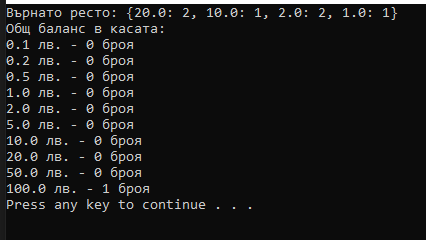
Стойности:

Начална стойност на касата: "1:1, 2:2, 10:1, 20:2, 50:0"

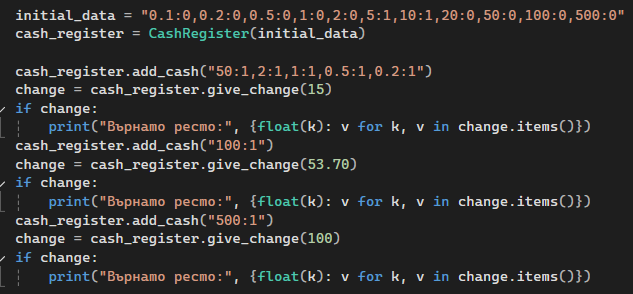
Добавяне на пари: "100:1"

Изпълнена операция за ресто: 55 лв.

Очакван резултат: Програмата трябва да върне точно ресто от 55 лв, като в касата трябва после да останат само 100лв.



1. Сценарии 2:



Начална стойност на касата:“0.1:0,0.2:0,0.5:0,1:0,2:0,5:1,

10:1,20:0,50:0,100:0,500:0“

Добавяне на пари: „50:1,2:1,1:1,0.5:1,0.2:1“

Изпълнена операция за ресто: 15 лв.

Остава в касата: „53.70“

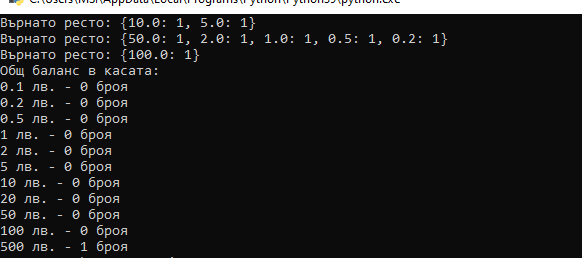
Добавяне на пари: „100:1“

Изпълнена операция за ресто: 53.70 лв.

Остава в касата: „100:1“

Добавяне на пари: „500:1“

Изпълнена операция за ресто: 100 лв.

 Остава в касата: „500:1“

1. Приложение:

https://github.com/edi9631/SPr/tree/main