

#### قنادي حاج محمود و شركا

حاج محمود که یکی از کسبه بازار است، در راسته شیرینی فروشها یک قنادی دارد و سالیان سال است که در همین دکان کار میکند. او به اندازهای کارش را دوست دارد که با وجود شیوع ویروس کرونا حاضر نشد درب دکان را ببندد بلکه تصمیم گرفت از این فرصت استفاده کند و برند خودش را جهانی کند. از طرفی عید نوروز فرصتی است که باید آن را غنیمت می شمرد، بنابراین از نوهاش خواست که یک سامانه فروش آنلاین برایش دست و پا کند تا به اهدافش برسد. به نوه حاج محمود کمک کنید تا این سامانه را پیادهسازی کند.

در این سوال، از شما میخواهیم که این برنامه را طبق UML Class Diagram داده شده، ییادهسازی کنید.

			Main			
		main (args	:string():void			
	_					
			ProgramController			
		confections	ary : Confectionary			
		run(): void	1			
	- 2	addCustom	er (matcher : Matcher) : void			
	- 0	chargeCust	omerBalance(matcher : Matcher) : void			
		add///areho	use (matcher : Matcher) : void			
	-1	increase\//a	rehouseMaterial (matcher : Matcher) : void			
	- 8	addSweet (	matcher : Matcher) : void			
	-1	increaseSw	eet (matcher : Matcher) : void			
	-8	addDiscoun	t (matcher : Matcher) : void			
	-8	addDiscoun	ntToCustomer (matcher : Matcher) : void			
	- 5	self9weet(n	natcher : Matcher) : void			
	-8	acceptTrans	sAction(matcher : Matcher) : void			
		printTransA	ctions(): void			
	1.5	printincome	(): void			
	-9	getComman	ndMatcher(input : String , regex : String) : Matcher			
		_				_
		_				
Customer	Transaction		Sweet		Warehouse	Confectionary
oustomers : ArrayList=Customer-	-idCounter:int.		- name : String		- amount : int	- balance: int
- customers : ArrayList+Customers - name: String	=idCounter_int. - id : int		- name : String - price: Int		- amount : int - materialName : String	
oustomers_: ArrayList: Customers name: String - lid; int	_idCounter.int. - id: int - customerid: int		- name : String - price: int - amount: int		- amount : int	- balance: int
_oustomers . ArrayLast-Customer: - name: String - id: int - balance: int	-idCounterLint id : int customend : int amount : int		- name : String - price: Int - emount: int - materials: HashMap <string,integer></string,integer>		- amount : int - materialName : String	- balance: int - discounts : HashMap <integer.integer></integer.integer>
oustomers_: ArrayList: Customers name: String - lid; int	LisCounterLinit - sticle - sticle - customend i let - amount linit - discountCode int		- name : String - price: int - amount: int		- arrount : int  - material Name : String  - watchouses : Array List-Warehouses	- balance: int - discounts: HashMap <integer,integer> + Confectionary () : &lt;<constructor>&gt;</constructor></integer,integer>
_oustomers . ArrayLast-Customer: - name: String - id: int - balance: int	_SCOunterCute.  -II: tet -outstreend: int -amount: let -decountCode int -decountCode int -decountCode int -decountCode int -decountCode int -decountPlace. int		-name: Sting -price; nt -amount id: -amoun	ing.Integer>): < <constructor>&gt;</constructor>	- amount : int  - materialName - String  - waterbouses : ArrayList-Warehouse:  - Wirehouse (materialName : String , amount : int): < <constructor>-</constructor>	- balance: int  -discounts. Healthlian-Integer.Integer:  - Confectionary (): < <constructor> - setBelance (balance: int): void  - getBalance(): int  - incresesBalance (int): void</constructor>
- Gustomers Arrand Sit - Customerc name String id. inti discount Code int discount Code int discount Code int	JOSOMERY ID:  14 Int Int  - Customend I let  - Amount I let  - Account Soil I let  - Acc		-name : String -price : Its -price : Its -manufal : -manufal : -manufal : NachMap-String Integer -creeks_Namblat-String Integer -creeks_Namblat-String Inter-int , materials : NachMap-Strin -politolarials of : NachMap-String Integer -politolarials of : NachMap-String Integer	ing.Integer>): < <constructor>&gt;</constructor>	- ambount; int  - materialName : String  - switchbouse 2. Ambolist-Villenbouse  - Warehouse (materialName : String , amount : int) - <-constructor  - increasolaterial (amount : int) - youd  - elef-mouth (amount : int) - youd  - guiderount () int  - guiderount (	- balance: int  - discounts. HashMas-integer.Integer discounts. HashMas-integer.Integer confectionary (): «constructor» - selfBalance(): int  - increaseBalance(): int  -
_oustomers_Ame_List*Countemers -name_String -ist int -basence int -decountCode_int -customer (name_String_id_int): < <constructor>&gt;</constructor>	-MCContext.int - 40 cet - outstanded int - amount int - amount int - decount/code int - d		-name String -notice sit -mount let -mount l	ing.Infeger>); < <constructor>&gt;</constructor>	-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
- Outlament - Amaulint-Customer: - name: Sing - Id Int - Balance Int - decount-Ook Int - Outlamer (name - Sting , Id Int) constructor-> - outCustomer(b) Did Int) - Customer - outCustomer(b) Did Int) - Customer	SIGNATE. ISC  - ISL INF  - colored I II  - annot I II  - description I II  - descripti		-name : thing  -name : thing  -name in  -name : thing  -name : name : na	mg_Infege(+>) : < <constructor>&gt;</constructor>	- ambount; int  - materialName : String  - switchbouse 2. Ambolist-Villenbouse  - Warehouse (materialName : String , amount : int) - <-constructor  - increasolaterial (amount : int) - youd  - elef-mouth (amount : int) - youd  - guiderount () int  - guiderount (	- balance: int  - discounts. HashMas-integer.Integer discounts. HashMas-integer.Integer confectionary (): «constructor» - selfBalance(): int  - increaseBalance(): int  -
_outcomer_Area/utcCustomer  - name String  - ist if  - bearing: if  - decountCode Int  - Customer (name String , id Int) constructor->  - paticustomer(string String , id Int) constructor->  - paticustomer(string String , id Int) constructor->  - increaseCustomer(string String , id Int) constructor->  - increaseCustomer(string String string Int) vidd	- accounter_int - are term of	clor>>	-aans-dring -noo-st it	ng Inleger>): «constructor»	-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
- pastoners _ Arepublic Customer - name tibrog - vial ret - - balance : - decount-one ret - - customer (name : Strey, ris : ret) - <-constructors - _ pastoner (name : Strey, ris : ret) - <-constructors - _ pastoner (name : Strey, ris : ret) - vial - _ recesses (Customerical Biolice, rist) - vial - - setBalance(balance aret) - vaid - - setBalance(balance aret) - vaid	SIGNATE. IR.  - Gallered II.  - annot II.  - descendible III.  - descendible Indiana  - descendible Indiana  - mandado (outsmed) III., amount IIII., descundible III.  - mandado (outsmed) III., amount IIII., descundible III.  - mandado (outsmed) III., amount IIII., descundible III.  - mandado (outsmed) III., amount III., descundible III.  - mandado (outsmed) III.	ctor>>	James Strong  Grotic Int  James Line Strong, Integra- James Line Strong  James Lin	ing.linlegel=> ) : < <constructor>&gt;</constructor>	-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continent Americal Continent - American Continent -	SACOUNTE LIST  SET OF THE PROPERTY OF THE PROP	ctor>>	-pains Story -price per -price pe		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continent Americal Continent - Item String - of et - Item String - of	SIGNATE. IR.  - Gastered : IR.  - castered : IR.  - descendible in .  - small between .  - termination . Ameliat finesation.  - descendible . descendible .  - descendible . descendible . descendible .  - descendible . descendible	ctor>>	James Strop  Grotic Int  James Line Strop  James Line  James		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continents Acres (Accidentes)	SACounter_IE   SACounter_IE   - extrament int  - extrament  - extra	ctor>>	-pains Story -price per -price pe		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continent Americal Continent - American Strong - Information Strong Strong Strong - Strong Strong Strong Strong Strong - Accordance Strong Strong Strong - Accordance Strong Strong - Accordance Strong Strong - Accordance Strong	SIGNAME. IR.  - Gustered: IR  - annote: IR  - descendibles int  - small printer: IR  - small printer: IR  - manacides: Ameulat-finesadore  - terradocio (outbreef) int, annote: IRI, descendible int)   - referendibles (outbreef) int, annote: IRI, descendible int)   - referendibles (outbreef) int, annote: IRI, descendible int)   - annotes (outbreef) int annote: IRI, descendible int)   - annotes (outbreef) int)   - annotes (outbreef) int)   - int interaction/coaped (o) telling  - interaction	ctor>>	James Strop  Grotic Int  James Line Strop  James Line  James		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continents Acres (Accidentes)	SACounter_IE   SACounter_IE   - extrament int  - extrament  - extra	ctor>>	James Strop  Grotic Int  James Line Strop  James Line  James		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continents Acres (Accidentes)	SIGNAME. IR.  - Gustered: IH - amount is - descentible in - descentible - descentib	ctor>>	James Strop  Grotic Int  James Line Strop  James Line  James		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continents Acres (Accidentes)	Sidewise Let  side let  customend let  let  customend let  let  let  let  let  let  let  let	ctor>>	James Strop  Grotic Int  James Line Strop  James Line  James		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void
Continents Acres (Accidentes)	SIGNATE LIST  All IRI  -submetted int -amount one	clar>>	James Strop  Grotic Int  James Line Strop  James Line  James		-amount: ret -materialmen : String -material	- balance: int  - siscouristHashMan-intendinistear: - Confectionary () - <-constructor> - Selfishince balance: int() - void - perBalance() int - increaseBalance (belience int) - void - indiscouristicate int) - bolean - indiscouristicate int) - bolean - indiscouristicate int) - void



#### قنادي

قنادی، مقداری شیرینی دارد و نیز انبارهایی از مواد اولیه که برای پخت شیرینی از آن مواد استفاده میکند. همچنین، یک کیف پول دارد که درآمد حاصل از فروش شیرینیها به آن واریز می شود. علاوه بر آنها، یک لیست کد تخفیف نیز دارد که می تواند به مشتری هایش کد تخفیف هدیه بدهد.

## مشتري

هر مشتری یک اسم و یک شناسه یکتا دارد که از طریق آن میتواند موجودی حسابش را افزایش دهد.

#### انبار

انبار، مکانی است که در آن، مواد اولیه ذخیره میشوند. هر انبار تنها یک نوع مواد اولیه میتواند ذخیره کند. توجه کنید که انبارها با نام جنسی که ذخیره میکنند، شناخته میشوند و در نتیجه، برای هر ماده ی اولیه، بیش از یک انبار وجود ندارد.

#### شيريني

هر شیرینی، یک اسم و یک قیمت دارد و برای پخت آن نیز به یک یا چند نوع مواد اولیه نیاز است.

#### تراكنش

هر تراکنش، یک <u>شناسه</u>، یک <u>شناسه مشتری</u> و مقداری پول دارد و میتواند <u>تخفیف</u> نیز داشته باشد. بعد از خرید موفق هر شیرینی توسط مشتری، یک <u>تراکنش</u> ساخته می شود که در صورت تأیید شدن این تراکنش، پول از مشتری کم شده و به حساب قنادی واریز می شود.



برنامەنويسى شىءگرا تمرین ۲

## انواع عمليات ممكن

دقت کنید در صورت برخورد با هر دستور نامعتبری عبارت زیر چاپ میشود:

invalid command

#### افزودن قنادي

در اولین دستور، دستور ساخت و راهاندازی قنادی داده می شود. دقت کنید که تمام دستورهای قبل از این دستور، نامعتبر هستند و این دستور تنها یک بار به برنامه داده میشود.

create confectionary

#### افزودن یک مشتری با نام NAME و شماره شناسایی ID

add customer id [ID] name [NAME]

دقت کنید که شماره شناسایی، یک عدد طبیعی است. نام مشتری میتواند یک یا چند بخشى باشد؛ به عنوان مثال، اسمها مىتوانند dutchman يا tir andaz khafan باشند. تضمین می شود که اسم مشتری تنها شامل حروف است.

مثال:

add customer id 21 name ali asghar

اگر مشتریای با این شماره شناسایی، قبلا وجود داشت، باید خطای زیر چاپ شود:

customer with this id already exists





## شارژ کردن حساب مشتری (ریختن پول به مقدار AMOUNT به حساب مشتری با شناسه ID)

increase balance customer [ID] amount [AMOUNT]

تضمین می شود مقدار واریزی عددی طبیعی و مثبت است.

مثال:

increase balance customer 21 amount 25000

اگر مشتریای با این شماره شناسایی وجود نداشت، باید خطای زیر چاپ شود:

customer not found

تمرین ۲

#### افزودن انبار با مواد اوليه MATERIAL NAME و مقدار AMOUNT واحد

add warehouse material [MATERIAL NAME] amount [AMOUNT]

دقت كنيد كه نام مواد اوليه مىتواند يك يا چند بخشى باشد؛ به عنوان مثال، اسمها مىتوانند konjed يا roghan oila باشند. تضمين مىشود اسم مواد اوليه، تنها شامل حروف است.

مثال:

add warehouse material Sugar amount 50

اگر انباری با این مواد اولیه قبلا وجود داشت، باید خطای زیر چاپ شود:

warehouse having this material already exists





# افزایش مقدار مواد اولیه یک انبار با مواد اولیه MATERIAL NAME و مقدار AMOUNT

increase warehouse material [MATERIAL NAME] amount [AMOUNT]

مثال:

increase warehouse material Oil amount 10

اگر انباری با این مواد اولیه وجود نداشته باشد، باید خطای زیر چاپ شود:

warehouse not found

تمرین ۲

افزودن شیرینی (که تاکنون وجود نداشته است) با اسم NAME (که یکتاست)، قیمت PRICE، یک یا چند مواد اولیه MATERIAL NAME و مقدار AMOUNT مورد نیاز آن مواد اولیه برای ساخت شیرینی

به فرمت زیر دقت کنید (موارد مختلف با کاما جدا شدهاند):

add sweet name [NAME] price [PRICE] materials: [MATERIAL] [AMOUNT], [MATERIAL] [AMOUNT], ...

دقت کنید که نام شیرینی می تواند یک یا چند بخشی باشد؛ به عنوان مثال، اسمها می توانند khameE یا cake yazdi یا شند. تضمین می شود که اسم شیرینی ها تنها شامل حروف است، مقادیر مواد اولیه مورد نیاز اعدادی طبیعی هستند و برای هر شیرینی تنها یک بار این دستور داده می شود.

مثال:

add sweet name Cake price 2500 materials: khame 10, sugar 100

**توجه:** هر شیرینیای که با این دستور اضافه میشود، تعداد موجود آن در قنادی صفر است که برای اضافه کردن این مقدار، از دستور بعدی استفاده میشود.





#### خطاها هنگام ساخت شیرینی برای اولین بار

• اگر انبار مربوط به آن ماده (ها) وجود نداشت، خطای زیر نمایش داده می شود:

not found warehouse(s): [MATERIAL] [MATERIAL] ...

دقت کنید که هر کدام از از موادی که انبارش وجود نداشت، باید در این خطا به همان ترتیبی که در دستور اضافه کردن شیرینی وارد شده است، نمایش داده شود؛ به عنوان مثال، اگر انبارهای مواد اولیه برای ساخت کیک (خامه، شکر) وجود نداشت، خطای زیر چاپ می شود:

not found warehouse(s): khame sugar

اضافه کردن شیرینی با نام NAME به تعداد AMOUNT به موجودی فروشگاه (دقت کنید که هنگام افزایش شیرینی باید به همان مقدار AMOUNT از مواد اولیه مورد نیازش استفاده شود)

increase sweet [NAME] amount [AMOUNT]

با اجرای این دستور، تعداد شیرینیهای NAME به مقدار AMOUNT که عددی طبیعی است، افزایش می یابد.

اگر هنگام افزایش موجودی شیرینی، آن شیرینی از قبل وجود نداشته باشد، خطای زیر نمایش داده میشود:

sweet not found

تمرین ۲

اگر در انبارهای مربوطه، مقدار مورد نیاز موجود نبود، خطای زیر نمایش داده میشود:

insufficient material(s): [MATERIAL] [MATERIAL]  $\dots$ 





#### افزودن كد تخفيف به شماره CODE و مقدار قيمت PRICE به فروشگاه

add discount code [CODE] price [PRICE]

مثال:

add discount code 3223 price 10000

تمرین ۲

توجه کنید که شماره و مقدار کد تخفیف، اعدادی طبیعی هستند.

اگر کد تخفیف از قبل وجود داشت خطای زیر چاپ میشود:

discount with this code already exists

#### افزودن کد تخفیف به شماره CODE برای مشتری به شماره شناسایی ID

add discount code code [CODE] to customer id [ID]

تضمین می شود که هر مشتری در لحظه فقط می تواند یک کد تخفیف داشته باشد.

اگر کد تخفیفی با این شماره در لیست کدهای تخفیف فروشگاه وجود نداشت، خطای زیر نمایش داده شود:

discount code not found

اگر مشتریای با این شماره شناسایی وجود نداشت، خطای زیر نمایش داده شود:

customer not found

دقت کنید که خطاها به همین ترتیب چک شوند. در صورت برخورد با یک خطا، نیازی به چک کردن بقیهی خطاها نیست.

## ${f ID}$ فروش شیرینی با نام ${f NAME}$ به مقدار ${f AMOUNT}$ به مثاسه

sell sweet [NAME] amount [AMOUNT] to customer [ID]



مثال:

sell sweet cake amount 3 to customer 2

اگر شیرینیای با این نام وجود نداشت، خطای زیر نمایش داده شود:

sweet not found

اگر مقدار این شیرینی در فروشگاه کافی نبود، خطای زیر نمایش داده شود:

insufficient sweet

اگر مشتریای با این شماره شناسایی وجود نداشت، خطای زیر نمایش داده شود:

customer not found

اگر مشتری یول کافی برای خرید این مقدار شیرینی نداشت، خطای زیر نمایش داده شود:

customer has not enough money

دقت کنید که خطاها به همین ترتیب چک شوند. در صورت برخورد با یک خطا، نازی به چک کردن بقیهی خطاها نیست.

در صورتی که هیچ خطایی وجود نداشته باشد، یک تراکنش ایجاد و پیام زیر نمایش داده میشود:

transaction [ID] successfully created

دقت کنید که اولین تراکنش ساخته شده، با شماره ۱ ساخته می شود و تراکنش های بعدی نسبت به آن شمارهگذاری میشوند؛ مثلا، شماره چهارمین تراکنش ساخته شده، ۴ . است. همچنین، اگر مشتری کد تخفیف داشته باشد، این کد در تراکنش لحاظ شده و کد تخفیف مشتری باطل می شود. اگر مشتری کد تخفیف نداشته باشد، به جای آن مقدار ۱ \_ ثبت ميشود.



تمرین ۲

#### قبول کردن تراکنش با شماره ID

accept transaction [ID]

اگر تراکنشی با این شماره وجود نداشته باشد یا قبلاً قبول شده باشد، خطای زیر نمایش داده شود:

no waiting transaction with this id was found

بعد از اجرای موفق تراکنش، پول از حساب مشتری کم شده و به حساب قنادی ریخته می شود. اگر مشتری کد تخفیف داشته باشد، در این عملیات، مقدار آن تخفیف از کل یول کم شده و سیس مقدار باقی مانده بعد از تخفیف از حساب مشتری کم شده و به حساب قنادي رىخته مىشود.

#### چاپ کردن لیست تراکنش ها

print transactions list

با اجرای این دستور، تمام تراکنشهایی که قبول شده اند به ترتیب از اولین تراکنش ساخته شده، با ترتیب زیر در چاپ مشخصاتشان، چاپ می شوند:

شماره تراكنش (ID)، شماره شناسايي مشتري (customer id)، مقدار كل پول حاصل از فروش آن تعداد شيريني خريده شده (payment amount = sweet price \* amount)، كد تخفيف (discount code) و مقدار نهايي بعد از اعمال تخفيف (discount code).

transaction [ID]: [customer id] [payment amount] [discount code] [final price]

#### چاپ کردن پول فروشگاه

print income

با اجرای این دستور، مقدار یولی که در قنادی از فروش شیرینیها به دست آمده است، چاپ می شود.



## خروج از برنامه

در انتها، با وارد کردن عبارت زیر، اجرای برنامه پایان می یابد:

end