학번 : 2021563033

이름 : 박준규

**사용한 개념**

**함수 사용**

* 각 클래스에 함수를 정의하여 기능을 수행하도록 작성

**객체지향 프로그래밍**

* CoffeeShop, PointSystem 등과 같은 클래스를 정의하여 OOP 개념 사용

**프로퍼티와 초기화**

* 클래스에서 주 생성자를 통해 프로퍼티를 초기화

**고차함수 사용**

* sumOf를 사용하여 포인트를 계산하였음

**가변(var), 불변(val)을 사용**

* 수정이 필요한 변수는 var수정이 필요 없는 변수는 val을 사용함

Coffee 클래스 설명

orderList

* 리스트를 사용하여 커피 메뉴와 주문 목록을 관리함

takeOrder 함수

* 사용자로부터 커피를 주문 받는 함수
* 메뉴를 선택하고 주문 수량을 입력하면, 해당 정보를 바탕으로 Order객체를 생성하고 orderList에 추가한다.
* 5번(주문 종료) 를 선택할 때까지 반복

completeOrder 함수

* 금액을 계산하는 함수
* 주문한 커피의 가격과 적립될 포인트를 계산함
* 전화번호를 받아 포인트 시스템에 등록함
* 주문한 내역을 안내하고 최종 금액을 출력함
* 마지막으로 금액 할인, 포인트 적립, 차감을 한다.

orderList.clear()

* 고객의 계산이 끝나면 다음 고객을 받아야 함
* 주문리스트르 초기화 시킨다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

PointSystem 클래스 설명

customerPorints

* mutavleMapOf 컬랙션을 사용하여 고객 포인트를 관리하도록 코드를 작성함
* 전화번호를 pk로 가지지는 map기만 변수
* key : 전화번호
* value : 고객 정보

registerCustomer 함수

* 고객의 번호를 pk로 받고 map에 전화번호가 있는지 확인함.
* 만약 전화번호가 있다면 고객 정보를 반환
* 없다면 map에 고객 정보를 추가한다.

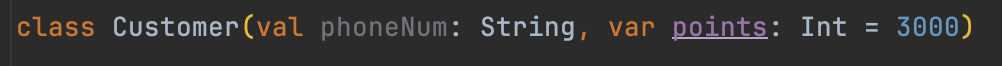
텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

프로퍼티와 초기화

Custoemr 클래스에서 주 생성자를 통해 프로퍼티를 초기화함

* Val phoneNum
  + 불변 프로퍼티로 생성함
* Var points
  + 가변 프로퍼티로 기본값을 3000으로 설정함



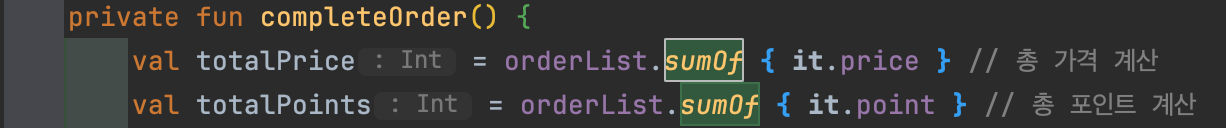
SumOf 고차함수

orderList.SumOf{it.price}

* orderList의 각 항목(it) 에서 price를 가져와 모두 더하는 작업을 함
* orderList에 있는 모든 주문의 가격을 계산한다.

totalPoints

* orderList의 주문한 상품의 모든 적립포인트를 계산하다.



가변 불변 변수

고객의 전화번호는 등록 후 변경되지 않으므로 val 변수를 사용한다.

그러나 고객의 포인트는 주기적으로 바뀌므로 var 를 사용함

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

**최종 코드(Main.kt)**

// 커피 클래스 정의함  
// 이름, 가격, 포인트를 속성으로 가짐  
class Coffee(val name: String, val price: Int, val point: Int)  
  
// 메뉴판 구성  
// 커피 이름, 가격, 포인트  
val *menupan* = *listOf*(  
 Coffee("아메리카노", 2000, 20),  
 Coffee("카페라떼", 2800, 28),  
 Coffee("카푸치노", 3200, 32),  
 Coffee("카페모카", 3400, 34)  
)  
  
// 주문 클래스 정의  
class Order(val coffee: Coffee, val quantity: Int) {  
 val name: String = coffee.name // 주문한 커피 이름  
 val price: Int = coffee.price \* quantity // 주문한 커피 총 가격  
 val point: Int = coffee.point \* quantity // 주문한 커피 총 포인트  
 // 커피 주문을 출력하는 함수  
 fun makeCoffee() = "$name $quantity 잔 $price 원입니다."  
}  
  
// 고객 클래스 정의  
// 초기 포인트 3000점이다.  
class Customer(val phoneNum: String, var points: Int = 3000)  
  
// 포인트 시스템 클래스 정의  
class PointSystem {  
 private val customerPoints = *mutableMapOf*<String, Customer>() // 고객 전화번호와 고객 객체를 맵으로 저장  
  
 // 고객 등록 함수, 신규 고객이면 새로 등록하고 포인트 3000점 부여  
 fun registerCustomer(phoneNum: String): Customer {  
 return customerPoints.*getOrPut*(phoneNum) **{** *println*("신규 고객이십니다. 3000포인트가 적립되었습니다.")  
 Customer(phoneNum)  
 **}** }  
}  
  
// 커피숍 클래스 정의  
class CoffeeShop {  
 private val orderList = *mutableListOf*<Order>() // 주문 리스트  
 private val pointSystem = PointSystem() // 포인트 시스템 객체  
  
 // 주문을 받는 함수  
 fun takeOrder() {  
 *println*("================================================")  
 *println*("SUNBUCKS CAFE : 메뉴를 고르세요.")  
 while (true) {  
 *println*("1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료")  
  
 val menu = *readLine*()!!.*toInt*() // 메뉴 선택  
 if (menu in 1..4) {  
 *println*("몇 잔 필요하신가요? ")  
 val num = *readLine*()!!.*toInt*() // 주문 수량  
 val coffee = *menupan*[menu - 1] // 선택한 커피  
 val order = Order(coffee, num) // 주문 객체 생성  
 orderList.add(order) // 주문 리스트에 추가  
 } else if (menu == 5) {  
 completeOrder() // 주문 완료 처리  
 break  
 } else {  
 *println*("없는 메뉴 입니다.")  
 }  
 }  
 }  
  
 // 주문 완료 처리 함수  
 private fun completeOrder() {  
 val totalPrice = orderList.*sumOf* **{ it**.price **}** // 총 가격 계산  
 val totalPoints = orderList.*sumOf* **{ it**.point **}** // 총 포인트 계산  
  
 *println*("포인트 적립을 위해 전화번호를 입력해주세요.")  
 val phoneNum = *readLine*()!! // 전화번호 입력  
 val customer = pointSystem.registerCustomer(phoneNum) // 고객 등록  
  
 val currentPoints = customer.points // 현재 포인트  
 val discountPrice = if (currentPoints >= totalPrice) totalPrice else currentPoints // 할인 금액 계산  
  
 // 주문 내역 출력  
 for (order in orderList) {  
 *println*(order.makeCoffee())  
 }  
  
 *println*("포인트로 $discountPrice 원 할인되었습니다.")  
 *println*("총 금액은 ${totalPrice - discountPrice} 원 입니다.")  
  
 // 포인트 차감 및 적립  
 customer.points = currentPoints - discountPrice + totalPoints  
 *println*("현재 포인트는 ${customer.points}점 입니다.")  
  
 orderList.clear() // 주문 리스트 초기화  
 *println*("카페를 이용해주셔서 감사합니다.")  
 }  
}  
  
// 메인 함수, 커피숍 객체 생성 후 주문 받기 반복  
fun main() {  
 val coffeeShop = CoffeeShop()  
 while (true) {  
 coffeeShop.takeOrder()  
 }  
}

**코드 실행 결과**

* 010-0000-0000 고객의 첫 주문

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 010-0000-0000 고객의 두번쨰 주문

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 새로운 고객(010-1111-1111)의 주문

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* 010-1111-111 고객의 두번째 주문

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

* 010-0000-0000 고객의 세번째 주문

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

**콘솔 출력 결과**

================================================

SUNBUCKS CAFE : 메뉴를 고르세요.

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

1

몇 잔 필요하신가요?

2

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

4

몇 잔 필요하신가요?

4

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

5

포인트 적립을 위해 전화번호를 입력해주세요.

010-0000-0000

신규 고객이십니다. 3000포인트가 적립되었습니다.

아메리카노 2 잔 4000 원입니다.

카페모카 4 잔 13600 원입니다.

포인트로 3000 원 할인되었습니다.

총 금액은 14600 원 입니다.

현재 포인트는 176점 입니다.

카페를 이용해주셔서 감사합니다.

================================================

SUNBUCKS CAFE : 메뉴를 고르세요.

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

2

몇 잔 필요하신가요?

1

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

5

포인트 적립을 위해 전화번호를 입력해주세요.

010-0000-0000

카페라떼 1 잔 2800 원입니다.

포인트로 176 원 할인되었습니다.

총 금액은 2624 원 입니다.

현재 포인트는 28점 입니다.

카페를 이용해주셔서 감사합니다.

================================================

SUNBUCKS CAFE : 메뉴를 고르세요.

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

1

몇 잔 필요하신가요?

1

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

5

포인트 적립을 위해 전화번호를 입력해주세요.

010-1111-1111

신규 고객이십니다. 3000포인트가 적립되었습니다.

아메리카노 1 잔 2000 원입니다.

포인트로 2000 원 할인되었습니다.

총 금액은 0 원 입니다.

현재 포인트는 1020점 입니다.

카페를 이용해주셔서 감사합니다.

================================================

SUNBUCKS CAFE : 메뉴를 고르세요.

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

2

몇 잔 필요하신가요?

1

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

4

몇 잔 필요하신가요?

2

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

1

몇 잔 필요하신가요?

1

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

5

포인트 적립을 위해 전화번호를 입력해주세요.

010-1111-1111

카페라떼 1 잔 2800 원입니다.

카페모카 2 잔 6800 원입니다.

아메리카노 1 잔 2000 원입니다.

포인트로 1020 원 할인되었습니다.

총 금액은 10580 원 입니다.

현재 포인트는 116점 입니다.

카페를 이용해주셔서 감사합니다.

================================================

SUNBUCKS CAFE : 메뉴를 고르세요.

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

1

몇 잔 필요하신가요?

2

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료

5

포인트 적립을 위해 전화번호를 입력해주세요.

010-0000-0000

아메리카노 2 잔 4000 원입니다.

포인트로 28 원 할인되었습니다.

총 금액은 3972 원 입니다.

현재 포인트는 40점 입니다.

카페를 이용해주셔서 감사합니다.

================================================

SUNBUCKS CAFE : 메뉴를 고르세요.

1.아메리카노 2.카페라떼 3.카푸치노 4.카페모카 5.주문 종료