| L logo 25 anos.png  **Universidade Luterana do Brasil**  **ULBRA – Campus Torres**  **Pró-Reitoria de Graduação** | | Tipo de atividade:  Prova ( ) Trabalho ( ) ..... ( )  Avaliação: G1 ( ) G2 ( )  Substituição de Grau: G1 ( ) G2 ( ) | |
| --- | --- | --- | --- |
| Curso: | Disciplina: | | Data: |
| Turma: | Professor(a): | | Valor da Avaliação:  Nota: |
| Acadêmico(a): n°: | | |

Usando o Projeto <https://github.com/ramonsl/TM_Atividades_Kotlin>

1. Notificações no dispositivo móvel

Normalmente, o smartphone oferece um resumo das notificações.

No código inicial fornecido escreva um programa que mostra a mensagem de resumo com base no número de notificações recebidas. A mensagem precisa incluir:

O número exato de notificações quando menor que 100.

99+ como o número de notificações quando houver 100 ou mais.

<https://github.com/ramonsl/TM_Atividades_Kotlin/blob/master/src/main/kotlin/NotificacoesNoDispositivoMovel.kt>

Complete a função printNotificationSummary() para que o programa mostre estas linhas:

Você tem 51 notificações.

Seu telefone possui 99+ notificações.

**fun main() {**

**val morningNotification = 51**

**val eveningNotification = 135**

**printNotificationSummary(morningNotification)**

**printNotificationSummary(eveningNotification)**

**}**

**fun printNotificationSummary(numberOfMessages: Int) {**

**if(numberOfMessages <= 99){**

**println("voce tem ${numberOfMessages} notificacoes")**

**}else {**

**println("Seu telefone possui 99+ notificações")**

**}**

**}**

2. Preço do ingresso do cinema

Geralmente, o preço dos ingressos difere dependendo da idade dos usuários.

No código inicial fornecido, escreva um programa que calcule estes preços de ingressos com base na idade:

https://github.com/ramonsl/TM\_Atividades\_Kotlin/blob/master/src/main/kotlin/PrecoDoIngressoDoCinema.kt

* O preço do ingresso é US$ 15 para pessoas de até 12 anos.
* O preço padrão do ingresso é US$ 30 para pessoas com idade entre 13 e 60 anos. Às segundas-feiras, você pode aplicar um desconto para que o ingresso padrão dessa faixa etária custe US$ 25.
* O preço de US$ 20 do ingresso para idosos é válido para pessoas com 61 anos ou mais. Suponha que a idade máxima de um frequentador de cinema seja de 100 anos.

Um valor -1 indica que o preço é inválido quando um usuário inserir uma idade fora das especificações.

Complete a função ticketPrice() para que o programa mostre estas linhas:

The movie ticket price for a person aged 5 is $15.

The movie ticket price for a person aged 28 is $25.

The movie ticket price for a person aged 87 is $20.

**fun main() {**

**var personOneAge = 5;**

**var personTwoAge = 28;**

**var personThreeAge = 87;**

**ticketPrice(age = personOneAge, isMonday = true)**

**ticketPrice(age = personTwoAge, isMonday = true)**

**ticketPrice(age = personThreeAge, isMonday = true)**

**ticketPrice(age = -1, isMonday = true)**

**}**

**fun ticketPrice(age: Int,isMonday: Boolean) {**

**if (age <= 12 && age > 0) {**

**print("O valor para idade de $age é: 15")**

**} else if (age >= 13 && age <= 60) {**

**if (isMonday == true) {**

**println("Hoje é segunda feira e o valor para idade de $age é: 25")**

**} else {**

**println("O valor para idade de $age é: 30")**

**}**

**} else if (age >= 61 && age <= 100) {**

**println("O valor para idade de $age é: 20")**

**} else {**

**println("A idade $age é inválida!")**

**}**

**}**

## 3[. Conversor de temperatura](https://developer.android.com/codelabs/basic-android-kotlin-compose-kotlin-fundamentals-practice-problems?hl=pt_br#3)

Existem três padrões principais de temperatura usados no mundo: Celsius, Fahrenheit e Kelvin.

No código inicial fornecido no abaixo, escreva um programa que converta uma temperatura de um padrão a outro usando estas fórmulas:

* Celsius para Fahrenheit: °*F* = 9/5 (°*C*) + 32
* Kelvin para Celsius: °*C* = *K* - 273,15
* Fahrenheit para Kelvin: *K* = 5/9 (°*F* - 32) + 273,15

O método String.format("%.2f", /\* measurement \*/ ) é usado para converter um número em um tipo String com duas casas decimais.

Complete a função main() para chamar a função printFinalTemperature() e mostrar as linhas abaixo. É necessário transmitir argumentos para a fórmula de conversão e a temperatura.

27.0 degrees Celsius is 80.60 degrees Fahrenheit.

350.0 degrees Kelvin is 76.85 degrees Celsius.

10.0 degrees Fahrenheit is 260.93 degrees Kelvin

**fun main(){**

**var valorCelcius = 25.0**

**var valorFahrenheit = 77.0**

**celciusForFahrenheit(valorCelcius)**

**fahrenheitForCelsius(valorFahrenheit)**

**fahrenheitForKelvin(valorFahrenheit)**

**}**

**fun celciusForFahrenheit(valor: Double){**

**println(9.0 \* valor / 5.0 + 32.0)**

**}**

**fun fahrenheitForCelsius(valor: Double){**

**println(5.0 \* (valor - 32.0) / 9.0)**

**}**

**fun fahrenheitForKelvin(valor: Double){**

**println(valor + 459.67 \* 5/9)**

**}**

## 4. Catálogo de músicas

Imagine que você precise criar um app de reprodução de música.

Crie uma classe que represente a estrutura de uma música. A classe Song precisa incluir estes elementos de código:

* Propriedades do título, artista, ano de lançamento e contagem de reprodução.
* Uma propriedade que indica se a música é famosa. Se o número for menor que 1.000, considere que não gostam muito dela.
* Um método que mostra uma descrição de música neste formato:

"[Título], de [artista], lançado em [ano de lançamento]."

**fun main() {**

**val results = Song("Cowboys Tears","Oliver Tree", 2022 )**

**results.printing()**

**}**

**class Song(**

**var titulo: String,**

**var artista: String,**

**var anoLancamento: Int,**

**var reproducoes: Int = 1200,**

**var famosa: String = "Gostaram",**

**){**

**fun printing(){**

**if(reproducoes >= 1000){**

**println("$titulo do ${artista}, lançado em ${anoLancamento}")**

**println("A musica possui ${reproducoes} reproducões, todos ${famosa} ")**

**}else {**

**println("$titulo do ${artista}, lançado em ${anoLancamento}")**

**println("A musica possui ${reproducoes} reproducões, nem todos ${famosa}")**

**}**

**}**

**}**

## 5. Perfil da Internet

Muitas vezes, é necessário preencher perfis para sites on-line que contêm campos obrigatórios e não obrigatórios. Por exemplo, você pode adicionar suas informações pessoais e enviar um link para as pessoas que indicaram que você criasse o perfil.

No código inicial fornecido no abaixo, escreva um programa que mostre os detalhes do perfil de uma pessoa.

Complete a função showProfile() para que o programa mostre estas linhas:

Name: Amanda

Age: 33

Likes to play tennis. Doesn't have a referrer.

Name: Atiqah

Age: 28

Likes to climb. Has a referrer named Amanda, who likes to play tennis.

**fun main() {**

**val amanda = Person("Amanda", 33, "play tennis", null)**

**val atiqah = Person("Atiqah", 28, "climb", amanda)**

**amanda.showProfile()**

**atiqah.showProfile()**

**}**

**class Person(val name: String,**

**val age: Int,**

**val hobby: String?,**

**val referrer: Person?**

**) {**

**fun showProfile() {**

**if(referrer == null){**

**println("Name: ${name}, Age: ${age}, Likes to: ${hobby}, Doesnt have a referrer")**

**}else{**

**println("Name: ${name}, Age: ${age}, Likes to: ${hobby}, Has referrer named ${referrer.name}, who likes to ${referrer.hobby}")**

**}**

**}**

**}**

## 6. Smartphones dobráveis

Normalmente, pressionar o botão liga/desliga do smartphone ativa ou desativa a tela dele. Por outro lado, se um smartphone dobrável estiver dobrado, a tela interna principal dele não vai ser ativada quando o botão liga/desliga for pressionado.

No código inicial fornecido no snippet abaixo, escreva uma classe FoldablePhone herdada da classe Phone. Ela vai conter o seguinte:

* Uma propriedade que indica se o smartphone está dobrado.
* Um comportamento da função switchOn() diferente da classe Phone para que ela só ative a tela quando o telefone não estiver dobrado.
* Métodos para mudar o estado da dobra.

**fun main(){**

**val foldablePhone = FoldablePhone()**

**// off**

**foldablePhone.switchOn()**

**foldablePhone.unfold()**

**foldablePhone.checkPhoneScreenLight()**

**// on**

**foldablePhone.switchOn()**

**foldablePhone.fold()**

**foldablePhone.checkPhoneScreenLight()**

**}**

**open class Phone(var isScreenLightOn: Boolean = false){**

**open fun switchOn() { //has to be open cus i override inside of FoldablePhone**

**isScreenLightOn = true**

**}**

**fun switchOff() {**

**isScreenLightOn = false**

**}**

**fun checkPhoneScreenLight() {**

**val phoneScreenLight = if(isScreenLightOn) "on" else "off"**

**println("The phone screen's light is $phoneScreenLight.")**

**}**

**}**

**class FoldablePhone(var isFolded: Boolean = true) : Phone() {**

**override fun switchOn(){**

**if(!isFolded){**

**isScreenLightOn = true**

**}**

**}**

**fun fold() {**

**isFolded = true**

**}**

**fun unfold() {**

**isFolded = false**

**}**

**}**

## 7. Leilão especial

Normalmente, em um leilão, o maior lance determina o preço de um item. Nesse leilão especial, se não houver um lance para um item, ele vai ser vendido automaticamente para o sistema de leilão pelo preço mínimo.

No código inicial fornecido , você tem uma função auctionPrice() que aceita um tipo Bid? anulável como argumento:

Complete a função auctionPrice() para que o programa mostre estas linhas:

Item A is sold at 5000.

Item B is sold at 3000.

**fun main() {**

**val winningBid = Bid(5000, "Private Collector")**

**auctionPrice(winningBid, 5000)**

**auctionPrice(null, 3000)**

**}**

**fun auctionPrice(bid: Bid?, minimumPrice: Int){**

**if(bid == null){**

**println("Item B is sold at ${minimumPrice}.")**

**}else {**

**println("Item A is sold at ${bid.amount}")**

**}**

**}**

**class Bid(val amount: Int, val bidder: String)**