18 Практическая работа Классы и коллекции

Обзор приложения

Вы новый инженер-программист в команде приложения для отслеживания событий. Цель этого приложения - позволить пользователям отслеживать свои события. Ваша команда будет назначать вам задачи, чтобы помочь улучшить функциональность приложения.

В конце каждой задачи следует сравнить свое решение с предложенным. Существуют разные способы достижения желаемой функциональности, поэтому не беспокойтесь, если ваш код не совсем соответствует предоставленному коду решения.

Используйте код решения, предоставленный в предыдущей задаче, в качестве стартового кода для следующей задачи, чтобы начать с общей отправной точки.

Задача 1

Другой инженер-программист уже выполнил некоторую высокоуровневую работу над приложением, и вам поручено реализовать детали.

Вам нужно реализовать класс Event. Этот класс используется для хранения сведений о событии, введенных пользователем. (**Подсказка**: этому классу не нужно определять какие-либо методы или выполнять какие-либо действия.)

Для этой задачи вам необходимо создать класс данных с именем Event.

Экземпляр этого класса должен иметь возможность хранить:

* **Название** события в виде строки.
* **Описание** события в виде строки (может быть нулевым).
* [**Часть дня**](https://en.wikipedia.org/wiki/Dayparting) события в виде строки. Нам нужно только отслеживать, начинается ли событие утром, днём или вечером.
* **Продолжительность** события в минутах как целое число.

Прежде чем продолжить, попробуйте написать код самостоятельно.

Используя свой код, создайте экземпляр, используя следующую информацию:

* **Название** : Исследование Котлина
* **Описание** : Посвятите изучению Kotlin не менее 15 минут в день.
* **Время суток** : Вечер
* **Продолжительность** : 15

Попробуйте распечатать свой объект и убедитесь, что вы получили следующий результат:

Event(title=Study Kotlin, description=Commit to studying Kotlin at least 15 minutes per day., daypart=Evening, durationInMinutes=15)

Задача 2

Чтобы проект не прерывался, ваш менеджер решает использовать код, который мы предоставили для класса данных.

После того, как члены вашей команды Event какое-то время использовали класс, старший товарищ по команде понимает, что использование строки для части дня не идеально.

Некоторые разработчики сохранили значение «Утро», некоторые использовали «утро», а третьи — «УТРО».

Это вызвало множество проблем.

Ваша задача — исправить эту проблему, проведя [рефакторинг](https://en.wikipedia.org/wiki/Code_refactoring) . Рефакторинг — это процесс улучшения вашего кода без изменения его функциональности. Некоторые примеры — упрощение логики или перемещение повторяющегося кода в отдельные функции.

Какой тип класса можно использовать для моделирования ограниченного набора различных значений, чтобы помочь решить эту проблему?

Ваша команда хочет, чтобы вы изменили код части дня, чтобы использовать класс перечисления. Используя класс перечисления, ваши товарищи по команде вынуждены выбирать одно из предоставленных значений времени суток, что предотвращает проблемы такого типа.

Класс перечисления должен иметь имя Daypart. Он должен иметь три значения:

* MORNING
* AFTERNOON
* EVENING

Как бы вы создали этот класс перечисления?

Как бы вы реорганизовали свой Eventкласс, чтобы использовать его?

Прежде чем продолжить, попробуйте написать свое решение сейчас.

Нажмите **«Далее»** , чтобы увидеть, как мы это закодировали.

## Задача 3

Вашим коллегам нравится использовать рефакторинг Daypart, но у них есть другие проблемы.

Следующий код показывает, как в настоящее время создаются и сохраняются события пользователя.

val event1 = Event(title = "Wake up", description = "Time to get up", daypart = Daypart.MORNING, durationInMinutes = 0)  
val event2 = Event(title = "Eat breakfast", daypart = Daypart.MORNING, durationInMinutes = 15)  
val event3 = Event(title = "Learn about Kotlin", daypart = Daypart.AFTERNOON, durationInMinutes = 30)  
val event4 = Event(title = "Practice Compose", daypart = Daypart.AFTERNOON, durationInMinutes = 60)  
val event5 = Event(title = "Watch latest DevBytes video", daypart = Daypart.AFTERNOON, durationInMinutes = 10)  
val event6 = Event(title = "Check out latest Android Jetpack library", daypart = Daypart.EVENING, durationInMinutes = 45)

Они создали множество событий, и для каждого события сейчас требуется своя переменная. Чем больше событий создается, тем сложнее отслеживать их все. Насколько сложно будет при таком подходе определить, сколько событий запланировал пользователь?

Можете ли вы придумать лучший способ организовать хранение этих событий?

Каким образом можно хранить все события в одной переменной? ( **Примечание** : он должен быть гибким, так как можно добавлять больше событий. Он также должен эффективно возвращать количество событий, хранящихся в переменной.)

Какой класс или тип данных вы бы использовали? Как можно добавить больше событий?

Теперь ваша очередь реализовать эту функцию.

Задание 4.

Вашему менеджеру нравится, как развивается приложение, но он решает, что пользователь должен иметь возможность видеть сводку своих **коротких** событий в зависимости от продолжительности события. Например: «У вас 5 коротких событий».

«Короткое» мероприятие — это мероприятие продолжительностью менее 60 минут.

Как бы вы достигли этого результата, используя eventsпеременный код из решения предыдущей задачи?

**Примечание.** Это может помочь решить эту проблему в несколько этапов. Как бы вы отфильтровали события по их продолжительности? После того как вы отфильтруете нужные события, как вы определите их количество?

## Задание 5.

Вашим товарищам по команде нравится, как развивается приложение, но они хотят, чтобы пользователи могли видеть сводку всех событий и времени их дня.

Вывод должен быть похож на:

Morning: 3 events

Afternoon: 4 events

Evening: 2 events

eventsКак можно добиться такого результата, используя код переменной из предыдущего шага?

**Примечание.** Это может помочь решить эту проблему в несколько этапов. Это похоже на предыдущую задачу, за исключением того, что вместо разделения событий на две группы вам необходимо разделить их на несколько групп. Как бы вы сгруппировали события по времени суток? Как только вы их сгруппируете, как вы подсчитаете события в каждую часть дня?

## Задание 6.

В настоящее время ваш коллега находит и печатает последний элемент, используя его индекс. Используемый код: println("Last event of the day: ${events[events.size - 1].title}").

Ваш менеджер предлагает проверить [документацию Kotlin](https://kotlinlang.org/docs/collection-elements.html#retrieve-by-position) на предмет функции, которая могла бы упростить этот код.

Какую функцию вы нашли?

Попробуйте использовать его, чтобы убедиться, что вы получаете те же результаты при печати.

Нажмите **«Далее»** , чтобы увидеть решение.

## Задание 7.

Вашей команде нравится разработанный вами класс данных, но она считает утомительным писать код каждый раз, когда им нужна длительность события в виде строки:

val durationOfEvent = if (events[0].durationInMinutes < 60) {  
        "short"  
    } else {  
        "long"  
    }  
println("Duration of first event of the day: $durationOfEvent")

Хотя вы могли бы исправить это повторение, добавив метод непосредственно в класс, это не идеально, поскольку другие команды начали использовать ваш класс событий в своих приложениях. Если класс изменится, им придется повторно протестировать весь свой код, чтобы убедиться, что из-за вашего изменения ничего не сломалось.

Как можно написать свойство расширения, возвращающее те же значения, что и код выше, без непосредственного изменения класса данных?

При правильной реализации вы сможете использовать следующий код, и он выведет то же сообщение, что и код, показанный в начале этой задачи.

println("Duration of first event of the day: ${events[0].durationOfEvent}")