

№ 12 Kotlin. Классы. Интерфейсы. Объекты. Операции

Задание

- 1) Проектирование интерфейсов и класса
 - a) Создайте интерфейс для шахматной фигуры и реализуйте его в классе шахматной фигуры. У интерфейса должно быть абстрактное свойство и функция, а также функция с реализацией по умолчанию.
 - b) Создайте класс, который описывает шахматную фигуру на игровом поле. В свойствах должно храниться имя фигуры, например «конь», цвет фигуры (можно быть представлено перечислением), координаты фигуры на игровом поле (буква и цифра, например E3). Предусмотрите вариант, когда фигура не расположена на игровом поле (м.б. null значения).
 - c) Определите конструкторы основной и вспомогательный.
 - d) Создайте инициализатор (init), который проверяет корректность установки фигуры на игровом поле и возможность создания фигуры (например, может быть одна королева и король, две ладьи и т.д.).
 - e) Определите функции хода (выдает возможные координаты перехода).
 - f) Все переходы фигуры должны записываться (история перемещений по игровой доске)
 - g) Фигура должна обладать состоянием, которое определяет – находится ли она под боем, может ли она побить фигуру другого цвета, можно ли сделать ход. Состояние сделайте на основе внутреннего или вложенного класса, выбор обоснуйте.
 - h) Определите для класса функцию расширения, которая удаляет фигуру с доски (обнулить координаты)
 - i) Создайте пять произвольных фигур, продемонстрируйте их возможности.
- 1) Создайте object класс шахматная доска с расстановкой фигур ней.
- 2) Создайте data класс для Игрока (имя, разряд, количество побед, проигрышей и т.п.)
- 3) Поздняя инициализация
 - a) Измените класс 'A', чтобы не генерировалось исключение UninitializedPropertyAccessException.

```
class A {  
    private lateinit var prop: String  
  
    fun setUp() {  
        prop = "100 + 200 "  
    }  
  
    fun display() {  
        println(prop)  
    }  
}
```

```

    }
}

fun main(args: Array<String>) {
    val a = A()
    a.display()
}

```

- b) Сделайте в классе перегрузку нескольких операторов (например, == , < , > ...). При необходимости добавьте свойства, функции.
- c) Напишите функцию `converter`, которая принимает в качестве параметра строку, а возвращает функциональный тип:

```
fun converter (which: String) -> ((Double, Double..) -> Double )?
```

Она принимает строку содержащую операцию: “+” , “-“ , “/” , “*” и возвращает одну из четырех функций от двух аргументов. Предусмотрите случай, когда введена не верная операция – в этом случае должно возвращаться `null`.

Вопросы при защите:

1. Как определить интерфейс и что он может содержать?
2. Приведите пример реализации метода интерфейса по умолчанию.
3. Что такое функциональное программирование?
4. Когда используются модификаторы `open`, `abstract` и `final`?
5. Расскажите о модификаторах видимости.
6. В чем отличие внутреннего и вложенного класса?
7. Какие бывают конструкторы?
8. Что такое `init`? В какой последовательности происходит инициализация свойств в классе.
9. Что такое поздняя инициализация?
10. Что такое отложенная инициализация?
11. Как объявляются свойства класса? (`val` и `var`).
12. Если в классе не определять конструктор, то его не будет вообще?
13. Приведите пример вторичного конструктора. Можно и в нем объявить свойства?
14. Как сделать наследование классов?
15. Как реализуются свойства интерфейсов?
16. Что такое классы данных? Расскажите про автоматическую деструктуризацию. Какие есть ограничения для классов данных.
17. Как используется `object` в Kotlin? Поясните в каких ситуациях.
18. Как выполняется перегрузка операций для класса?
19. Как выполнить делегирование свойств?