

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Программирование мобильных информационных систем

Отчет
по лабораторной работе №4
на тему:
«Объектно ориентированное программирование»

Выполнил:
ст. гр. 214302
Короткая Д.О.

Проверил:
Усенко Ф.В

Минск 2024

Задание: Напишите систему отчетности для компании, включающую обобщенные классы `Report<T>` и `SummaryReport<T>`. Реализуйте методы для создания отчетов по различным аспектам бизнеса (например, продажи, расходы, производительность сотрудников) с поддержкой фильтрации и агрегации данных.

Листинг кода:

```
open class Report<T>(  
    private val data: List<T>  
) {  
    fun filter(predicate: (T) -> Boolean): List<T> {  
        return data.filter(predicate)  
    }  
    fun aggregate(selector: (T) -> Double): Double {  
        return data.sumOf(selector)  
    }  
    fun display() {  
        data.forEach { println(it) }  
    }  
}  
  
class SummaryReport<T>(  
    private val report: Report<T>  
) {  
    fun getTotal(selector: (T) -> Double): Double {  
        return report.aggregate(selector)  
    }  
    fun displaySummary(selector: (T) -> Double) {  
        println("Сводный отчет:")  
        report.display()  
        println("Общая сумма: ${getTotal(selector)}")  
    }  
}  
  
data class Sale(val product: String, val amount: Double)  
data class Expense(val category: String, val cost: Double)  
data class EmployeePerformance(val name: String, val hoursWorked: Int)  
  
fun main() {  
    val salesData = listOf(  
        Sale("Телефон", 100.0),  
        Sale("Компьютер", 150.0),  
        Sale("Телевизор", 200.0)  
    )  
  
    val salesReport = Report(salesData)  
    println("Отчет по продажам:")  
    salesReport.display()  
  
    val salesSummary = SummaryReport(salesReport)  
    salesSummary.displaySummary { sale -> sale.amount }  
  
    val expensesData = listOf(  
        Expense("Оплата аренды", 500.0),  
        Expense("Платеж за коммунальные услуги", 200.0),  
        Expense("Зарплата сотрудников", 1000.0)  
    )  
  
    val employeePerformance = Report(listOf(  
        EmployeePerformance("Иванов", 160),  
        EmployeePerformance("Петров", 140),  
        EmployeePerformance("Сидоров", 180)  
    ))  
  
    val employeeSummary = SummaryReport(employeePerformance)  
    employeeSummary.displaySummary { emp -> emp.hoursWorked }  
}
```

```

        Expense("Реклама", 100.0),
        Expense("Производство", 500.0),
        Expense("Доставка", 200.0)
    )

    val expensesReport = Report(expensesData)
    println("\nОтчет по расходам:")
    expensesReport.display()

    val expensesSummary = SummaryReport(expensesReport)
    expensesSummary.displaySummary { expense -> expense.cost }

    val performanceData = listOf(
        EmployeePerformance("Катя", 40),
        EmployeePerformance("Саша", 35),
        EmployeePerformance("Петя", 45)
    )

    val performanceReport = Report(performanceData)
    println("\nОтчет по производительности сотрудников:")
    performanceReport.display()

    val performanceSummary = SummaryReport(performanceReport)
    performanceSummary.displaySummary { performance ->
        performance.hoursWorked.toDouble() }
}

```

Контрольные вопросы:

1. Какие типы циклов существуют в Kotlin?

В Kotlin есть три типа циклов: `for`: для итерации по коллекциям, массивам или диапазонам. `while`: выполняется, пока условие истинно. `do...while`: выполняется хотя бы один раз, затем проверяется условие.

2. Как работает цикл `for` для итерации по элементам массива?

`val array = arrayOf(1, 2, 3) for (element in array) { println(element)}` Этот цикл проходит по каждому элементу массива `array` и выводит его.

3. Что такое диапазон в Kotlin, и как его создать?

Диапазон — это последовательность значений, созданная с помощью оператора `...`. Например: `val range = 1..5` // Диапазон от 1 до 5 включительно