Факультатив
Язык Kotlin
II

План занятия

I.Орг. Вопросы
II.Принципы ООП очень кратко
III.ООП в Kotlin
   IV.Классы, конструкторы
   V. Поля, свойства
   VI.Интерфейсы
   VII.Объекты
   VIII.Классы-данные
   IX.Наследование, уровни доступа
   X.Вложенные и внутренние классы
   XI.Функции расширения
XII.Перерыв
XIII. Самостоятельное решение задач и защита выполненных дома
XIV. Домашнее задание

Орг. вопросы

I. Конкурс вместо ДЗ. Да/Нет?
II. Не все из задач, предлагаемых для самостоятельного решения, обязательно делать. Ориентируемся на то, чтобы набрать примерно 8-12 баллов в неделю. Если больше, хорошо.
III.Публикация ведомости. Да/нет?
IV.Возможный перенос занятий: 24 фев (праздник), 3 мар (я в отъезде)
V.Набор на стажировки – примерно в мае.

Обновленная версия

План по занятиям

3 фев | Введение. Процедурное программирование. Базовый синтаксис. Основные управляющие конструкции. Основы системы типов. Первые слова о коллекциях. Исключения и некоторые другие темы. Строки.
10 фев | ООП. Классы, интерфейсы, объекты. Наследование, свойства. Функции расширения.
17 фев | Функциональное программирование. Лямбда-выражения, функции высших порядков, встроенные функции. Работа с коллекциями.
24 фев | Комбинируем ООП и функциональный стиль. Приведения типов и др. Операторы, их перегрузка.
3 мар | Обобщения и работа с ними. Система типов языка.
10 мар | Предметно-ориентированные языки (DSL).
17 мар | Использование Java вместе с Kotlin
24 мар | Заключительные замечания. Другие интересные темы, которые успеем. Например: файлы, UI, тесты (jUnit).
? мар | Экзамен

Основная часть:)

Основы ООП

Объектно-ориентированное программирование. Программа — набор объектов разных типов (классов), взаимодействующих между собой (посредством методов, описывающих их поведение).

Ключевые слова:
- Класс
- Объект
- Абстрагирование
- Инкапсуляция
- Наследование
- Полиморфизм

Эквивалентность

`a === b` – ссылочная

`a == b` - структурная

Модификаторы

	К чему применяется
Final	Не может быть переопределен (по умолчанию)
Open	Может быть переопределен
Abstract	Должен быть переопределен
Override	Указывает, что это – переопределение (может быть переопределен далее, если не final)

Видимость:

	Член класса	Декларация верхн.уровня
Public	везде	везде
Internal	модуль	модуль
Protected	подклассы	не может применяться
Private	класс	файл

Вопросы?


JET
BRAINS
HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

Перерыв – 10 минут!

Практика

Сложность: низкая

Балл: 4

Задача 1

Объектно-ориентированный Hello World (устроить иерархию наследования, в которой для разных наследников печатаются разные сообщения).

Задача для тех, кто не видит сил выполнить сложную задачу *-2.

Сложность: средняя

Балл: 3

Задача 2

Спроектировать и реализовать класс с данным «Книга».

Класс должен уметь хранить библиографическую информацию: Название, имя автора (м.б. несколько), издательство, год издания, город, количество страниц, тип бумаги, тип обложки, ISBN, формат.

Книги могут отличаться по жанру, тематике.

Подумайте над тем, каким образом разумнее всего будет организовать иерархию классов, если вы пишете ПО для библиотеки.

Данный класс будет использоваться на следующем занятии.

Сложность: средняя

Балл: 3

Задача 2-2

Спроектировать и реализовать класс с данным «Книжный шкаф».

Шкаф содержит в себе полки, на которых могут размещаться книги.
Свойства шкафа: конфигурация полок, свойства полок: размеры, количество помещаемых книг.

Данный класс будет использоваться на следующем занятии.

Сложность: средняя

Балл: 5

Задача 3

Одно из заданий к первому занятию заключалось в необходимости реализовать простейший калькулятор. Допустим, что Вы хотите использовать эту реализацию в ОО программах.

Спроектируйте интерфейс Calculator, который бы позволял использовать различные реализации калькулятора для выражений, заданных строкой.

Оберните код, реализованный вами при решении задачи первого занятия в класс CalculatorImpl, реализующий интерфейс Calculator.

Разработайте функцию main для тестирования.

Сложность: низкая

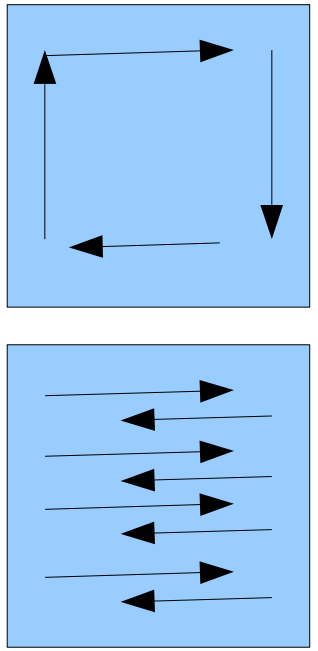
Балл: 2

Задача 4

На вход программе подается квадратная матрица NxN.

Реализуйте объект, содержащий функции для

- поворота матрицы
- отражения
- транспонирования
- вычисления определителя матрицы (+2)
  Обращайте внимание на эффективность выбранного метода!



Подготовьте тесты, демонстрирующие работоспособность кода

Большая задача

Сложность: высокая

Задача *

Балл: 25

Смоделировать лес, в котором на деревьях живут звери: белочки, бурундучки, барсучки, летяги и дятлы.

Некоторые сведения о модели леса (типы деревьев) у вас уже есть в выданном материале (виды деревьев: ель, сосна, дуб, береза, клен, орех).

Теперь деревья делятся на крону, ствол и корни. Ствол может содержать дупла, а корни – норы. Белки, летяги и дятлы живут в дуплах, бурундучки и барсучки – в норах. Звери могут перемещаться между частями дерева, а также между любыми деревьями в лесу. Они это делают в поисках еды и друзей.
Белочки питаются орешками и шишками (есть в кронах елей и сосен, ореха), бурундучки тоже едят орешки и шишки, но опавшие (есть в корнях тех же деревьев), летяги не едят шишки, но едят кленовые листья (есть в кронах кленов), дятлы едят червячков, которые живут в стволах. Барсуки едят корнеплоды, которые встречаются в некоторых корнях. Ресурсы периодически возобновляются. Если звери долго не едят, они умирают:(Если две зверя разного пола встречаются на одной части одного дерева, у них может появиться потомство.

Задача: спроектировать и реализовать модели леса с разными исходными наборами зверей/деревьев/распределением еды. Когда экосистема будет устойчивой?

Сложность: высокая

Задача *-2

Балл: 20

На дом:

Усложняем модель леса. Теперь задаётся топология леса. Представляем лес в виде графа. В узлах — деревья, ребра — соседство деревьев в лесу. Звери могут перемещаться только между соседними деревьями.

Также в лесу появляются два типа хищников: коршун и волк. Коршун может поймать в кроне дерева и съесть любого зверя, кроме барсука. Волк может ловить любого зверя на земле. Если хищники долго не кушают, они умирают.

Задача: построить модель и попробовать её для разных исходных ситуаций.
Когда система будет устойчива?