Казахский национальный университет имени аль-Фараби

Факультет информационных технологий

Методические указания для выполнения СРС по дисциплине **“Интеллектуальный анализ данных”**

**Задания на СРС: Блок А.** (Тест, выберите правильный ответ): (9 балл)

1. Что напечатает следующий фрагмент кода?

case class KaznuStudent(name: String, age: Int) val aidar1=new KaznuStudent(“Aidar”, 22) val aidar2=new KaznuStudent(“Aidar”, 22) println(bob1==bob2)

a) False

б) True

**с) Выдаст ошибку (throw error)**

д) Выдаст исключение (throw Exception)

2. Следующие утверждения верны для сопутствующих объектов и сопутствующих классов:

a) Сопутствующий объект-это объект с тем же именем, что и класс

б) Сопутствующие классы и объекты могут получить доступ к частным членам своих компаньонов

**с) Оба вышеперечисленных**

3. Что такое функция высшего порядка в Scala?

a) Он принимает другие функции в качестве параметров

б) В результате он возвращает функцию

**с) Оба вышеперечисленных**

4. Выделите правильные утверждения из следующих:

**a) Классы Case (case class) позволяют сопоставлять паттерны(поиск по шаблону pattern matching)**

**б) Мы должны использовать ключевое слово для создания экземпляра класса case(case class)**

с) Мы должны вручную определить методы доступа для всех аргументов конструктора

д) Мы должны сгенерировать методы equals(), hashcode () и toString()

5. Что содержит переменная x в следующем коде: var x,y,z=(1,2,3)

a) 1

**б) (1,2,3)**

с) Код выдаст ошибку

6. Коллекция типа collection.Seq неизмененная(immutable)

a) False

**б) True**

**Блок Б.** (Открытый вопрос) (12 балл)

Вопрос 1. Перечислите разницу между объектом и классом ?

**Объект** - это класс, но у него уже есть экземпляр, поэтому вы не можете вызвать new ObjectName(). С другой стороны, **Класс** является просто типом, и он может быть экземпляром, вызывая new ClassName(). **Класс**  -это определение, описание. Он определяет тип с точки зрения методов и состава других типов. **Объект**  - это singleton-экземпляр класса, который гарантированно уникален. Для каждого **объект**  в коде создается анонимный класс, который наследуется от любых классов, для реализации которых вы объявили **объект** . Этот класс нельзя увидеть из исходного кода Scala , хотя вы можете добраться до него через отражение.

class X {

// class X can see private members of object X

// Prefix to call

def m(x: Int) = X.f(x)

// Import and use

import X.\_

def n(x: Int) = f(x)

private def o = 2

}

object X {

private def f(x: Int) = x \* x

// object X can see private members of class X

def g(x: X) = {

import x.\_

x.o \* o // fully specified and imported

}

Вопрос 2. Что такое “Trait” в языке Scala, перечислите особенности ?

Трейты (Traits) используются чтоб обмениваться между классами информацией о структуре и полях. Они похожи на интерфейсы из Java 8. Классы и объекты могут расширять трейты, но трейты не могут быть созданы и поэтому не имеют параметров.

Минимальное объявление трейта - это просто ключевое слово trait и его имя: *trait Aida*

Трейты наиболее полезны в качестве обобщенного типа с абстрактными методами. Чтоб использовать трейты, необходимо наследовать класс от него используя ключевое слово extends. Затем необходимо реализовать все абстрактные члены трейта, используя ключевое слово override. Туда где требуется определенный тип трейта, мы можем передавать любой наследованный от требуемого трейта класс.

Вопрос 3. Что такое Case Class, перечислите особенности?

Классы образцы (Case classes) похожи на обычные классы с несколькими ключевыми отличиями, о которых мы поговорим ниже. Классы образцы хороши для моделирования неизменяемых данных. Минимальный вариант объявления класса образца: указание ключевого слова case class, название и список параметров (которые могут быть пустыми).

**case** **class** **Book**(isbn: **String**)

**val** b1 = **Book**("978-0486282114")

При создании класса образца с параметрами, эти параметры являются публичными и неизменяемыми. Классы образцы сравниваются по структуре, а не по ссылкам.

**case** **class** **Book**(isbn: **String,** page**:Int**)

**val** b1 = **Book**("978-0486282114", 250)

**val** b2 = **Book**("978-0486282114", 250)

Даже если b1 и b2 ссылаются на разные объекты, значения каждого из них равны. Вы можете создать копию экземпляра класса образца, просто воспользовавшись методом copy. При этом по желанию можно изменить аргументы конструктора.

**Блок В.** (Задача, требуется приложить ответ компилятора помимо решения самой задачи) (10 балл)

Задача 1.

Дано:

val someValue: Option[Double] = Some(20.0) val value = someValue match {

case Some(v) => v case None => 0.0

}

Найти значение “value”, чему равен?

**Value=20.0**

val noValue: Option[Double] = None val value1 = noValue match {

case Some(v) => v case None => 0.0

}

Найти значение “value1”, чему равен?

**Value1=0.0**

Задача 2.

Задан следующий список: val kaznuLabs = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10) нужно найти:

* длину списка

kaznuLabs.length

val res0: Int = 10

* удвоить значение элементов списка

for(i<-kaznuLabs)*print*(s"**$**{i \* 2} ")

2 4 6 8 10 12 14 16 18 20

* перевернуть список

kaznuLabs.reverse

val res0: List[Int] = List(10, 9, 8, 7, 6, 5, 4, 3, 2, 1)

* отфильтруйте все значения в списке, кратные 3

kaznuLabs.filter(\_%3==0)

val res0: List[Int] = List(3, 6, 9)

Задача 3.

Вычислите правильный выход:

val numbers=List(11,22,33)

var total=0 for(i<-numbers){ total+=i

}

println(total)

**66**

Задача 4.

Определите следующий метод, если задан параметр individual\_analysis\_data(("scala", "the best")) где метод individual\_analysis\_data:

def individual\_analysis\_data(expr: Any) =

expr match {

case ("go", "simple") => "Go language must have" case ("scala", "the best") => "Scala is the best high-load backend language" case ("python", "excellent") => "Python excellent for data and scripting" case \_ => "Okey, every language matter"

}

**Scala is the best high-load backend language**

**Блок Д.** (Гитхаб загрузка ответов) (4 балл)

Создайте в гите репозитории(проект) с названием kaznu\_name\_surname, где name и surname ваше имя и фамилия;

Залейте вашу работу в созданную гит репозиторию(инструкция доступна на самом сайте github);

Важно создавать проект под своим аккаунтом;

Важно создать открытый репозитории(public, not private) чтобы мог получить доступ;

**Дедлайн и формат выполнения СРС:**

Срок выполнения СРС на 48 часов(2 дня). C 11.00 четверг (01.04.2021) по 11.00 суббота (03.04.2021);

Нужно отправить ответы с оформлением на почту до 11.00(утро) 3 апрель(суббота) : kairattokpaev@gmail.com, и обязательно поставить в копию: alpysbay.gulbanu@gmail.com;

Также необходимо залить в github в свой аккаунт ваши решения до окончания дедлайна;

**Методические рекомендации по выполнению задания**

Примечание. При оценке работы будут учитываться следующие критерии: 1)соблюдение сдачи и полное выполнение требований к работе; 2)все работы должны включать титульный лист, реализация задачи и ответ компилятора (в виде текста или скриншота) 3) ваша работа должна бы отражать логику вашего мыслительного процесса по рассматриваемому вопросу.

Разработанная тематика СРС предусматривает изучение как теоретических, так и практических вопросов. При изучении каждой из тем, студент должен продумать план ответа по каждому вопросу, связывая теоретические положения с умениями, полученными в процессе практических занятий.

Целью выполнения самостоятельных работ является развитие умения анализировать код и практическое освоение синтаксиса и теории, закрепление знании полученных в ходе лекционных и лабораторных занятии;

уч. г. 2021 Токпаев К. К.

**Литература для самостоятельного чтения:**

1.Профессиональное программирование на Scala. Мартин Одерски, Лекс Спун, Билл Веннерс - П.,2017.

***Интернет-ресурсы***

https://www.scala-exercises.org/std\_lib https://www.scala-exercises.org/scala\_tutorial