**1)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 ?*

**Класс** - некая абстракция, логическая структура, описывающая поведение и характеристики. Например, машина. Она может ехать, сигналить и т.п

**Объект** - конкретный экземпляр класса. Например, конкретная ваша машина.

|  |  |  |
| --- | --- | --- |
| № | Объект | Класс |
| 1 | Объект - экземпляр класса | Класс - шаблон, по которому создаются объекты |
| 2 | Сущность из реального мира | Группа похожих объектов |
| 3 | Физическая сущность | Логическая сущность |
| 4 | Объект СОЗДАЁТСЯ | Класс ОПИСЫВАЕТСЯ |
| 5 | Объект можно создавать, как правило, много раз | Класс определяется единожды |

*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 Car*

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?*

**Объявление case class**

Указание ключевого слова case class, название и список параметров (которые могут быть пустыми). Пример:

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

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

Мы можем создать копию экземпляра case class, просто воспользовавшись методом copy. При этом по желанию можно изменить аргументы конструктора.

**case** **class** **Message**(sender: **String**, recipient: **String**, body: **String**)

**val** message4 = **Message**("message1", " message2", "message3")

**val** message5 = message4.copy(sender = message4.recipient, recipient = "ukwsdnh")

message5.sender

message5.recipient

message5.body

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

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

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

- перевернуть список kaznuLabs.reverse

- отфильтруйте все значения в списке, кратные 3 kaznuLabs.filter(\_%3==0)

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;