**Қазақстан Республикасы Білім және ғылым министрлігі**

**Әл-Фараби атындағы Қазақ Ұлттық Университеті**

**Ақпараттық технологиялар факультеті**

**Ақпараттық жүйелер кафедрасы**

****

**CӨЖ**

**Қабылдаған:** Токпаев Қ.Қ.

**Орындаған:** Бекберген О.Б.

**Тобы: ИС18-10 R&D**

**Aлматы 2021**

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

*Дұрыс жауаптарды қызыл түспен белгілеп шықтым.*

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)

Сұрақтарға *кітаптардан оқып, теориялық жағынан* және *практикадан түсінгендерім* бойынша жауап жаздым.

**Вопрос 1. Перечислите разницу между объектом и классом ?**

***Объект*** – класста бір объект экземпляры ғана бола алады.

Әрбір объект үшін кодта анонимді класс жасалады, ол object-ті іске асыру үшін жариялаған барлық кластардан мұра етеді және оны белгілі бір класстың экземплярларына қатысы жоқ статикалық мүшелерді сақтау үшін пайдалануға болады.

***Класс*** - бұл статикалық үлгілер, олардың экземплярлары көптеген объектілерде жасалуы мүмкін.

***Айырмашылықтары*:**

Егер объект пен класстың аты бірдей болса, онда класс *сопутствующий класс*, ал объект *сопутствующий объект* деп аталады.

Объектінің бір ғана экземпляры болады, ал класста бірнеше экземплярлар болса да болады.

Объект интерфейсті *class* немесе бір немесе бірнеше *trait* кеңейту арқылы жүзеге асыра алады.

Объектіге конструктор параметрлерін бере алмаймыз. Ал класста конструктордың аргументтері арқылы параметрленеді.

Объект тип емес, сондықтан *new* операторы арқылы оның экземплярларын құра алмаймыз. Бірақ оның өрістері, әдістері болуы мүмкін және суперкласты кеңейте алады. Ал класс тип ретінде қолданылады. Оның экземплярларын құра аламыз.

Программа құру үшін негізгі әдісті *class*-та емес, *object*-те қолдану керек.

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

Трейттерді код блоктарында бірнеше рет қолдана аламыз. Трейтте методтар мен өрістер инкапсуляцияланады, сондықтан класстарда бірнеше рет қолданыла алады. Мұрагерлеу класстарында әр класс тек тек бір класстың мұрагері ғана бола алады, ал трейттердің айырмашылығы бір класста көптеген трейттер бола алады.

Трейт жариялануы классқа ұқсас болып келеді. Бірақ трейт құрған кезде ***trait*** кілттік сөзі қолданылады.

Класстардан айырмашылығы, трейттерде *конструктор параметрлері* болмауы мүмкін.

Егер трейт құрған кезде оған родительский класс жарияланбаған болса, онда по умолчанию ***AnyRef*** болады.

Трейт құрылғаннан кейін оны ***extends*** кілттік сөзін немесе ***with*** кілттік сөзін қолдана отырып, классқа қоса аламыз. Егер дополнительный трейттер қосқымыз келсе, ***with*** кілттік сөзін қолданамыз.

Сонымен қатар трейт ***тип*** те бола алады.

Трейттерде өрістер жарияланып, состояниелер сақталуы мүмкін.

Трейттерде классқа тән параметрлер болмайды.

Классдарда super вызовтар статикалық, ал трейтте — динамикалық болып табылады.

**Вопрос 3. Что такое Case Class, перечислите особенности?**

***Case Class*** - бұл дизайн параметрлерін экспорттайтын және үлгіні сәйкестендіру арқылы рекурсивті ыдырау механизмін ұсынатын қарапайым класстар.

*Case Class* конструкторының параметрлері жалпы қол жетімді ретінде өңделеді және тікелей қол жетімді болады.

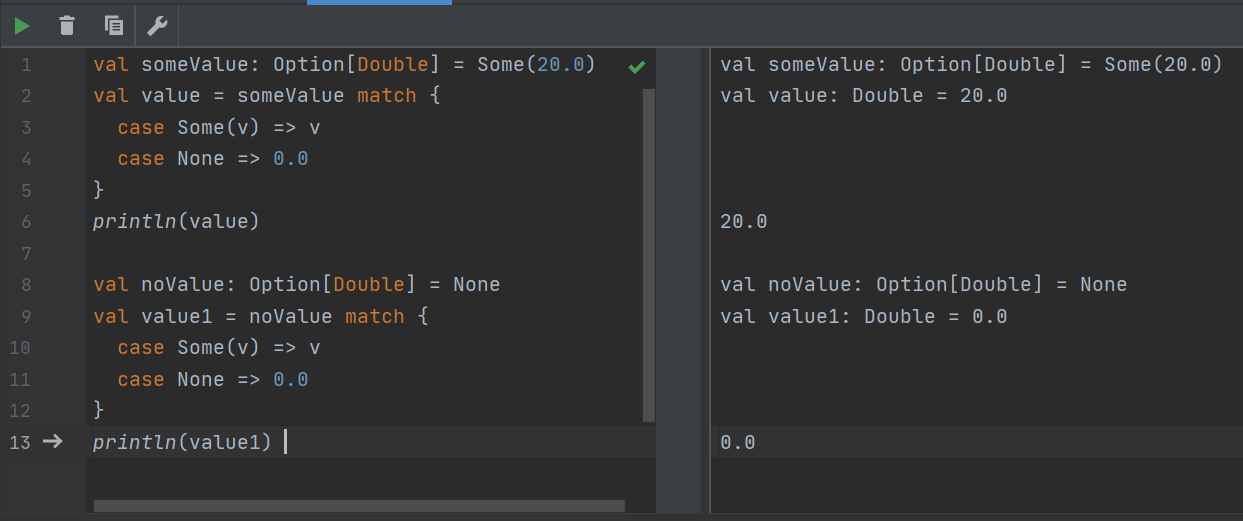
*Case Class*-тың экземплярларын құру үшін кілттік сөзін қолдану қажет емес.

Әрбір *Case Class* үшін Scala компиляторы құрылымдық теңдікті және *toString* әдісін қолданатын *equals* әдісін генерациялайды.

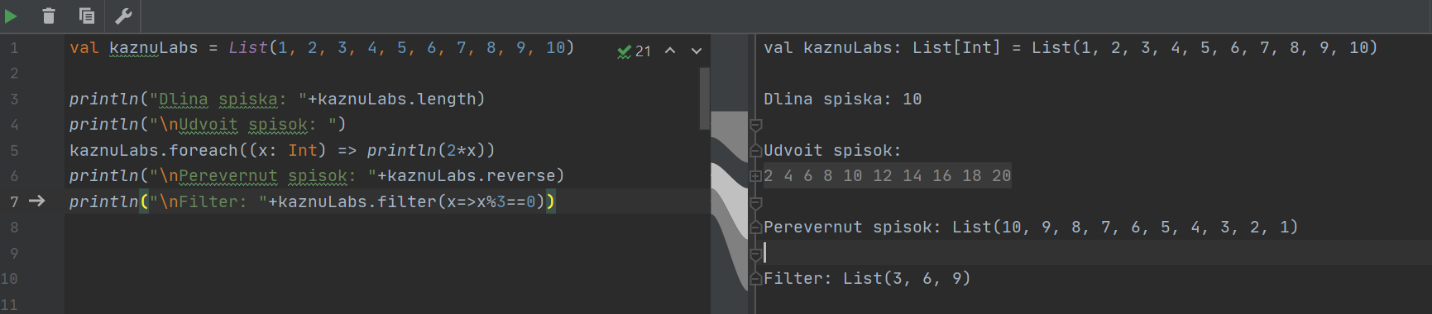
*Case Class*-та по умолчанию параметрлер және именованный параметрлер бола алады.

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

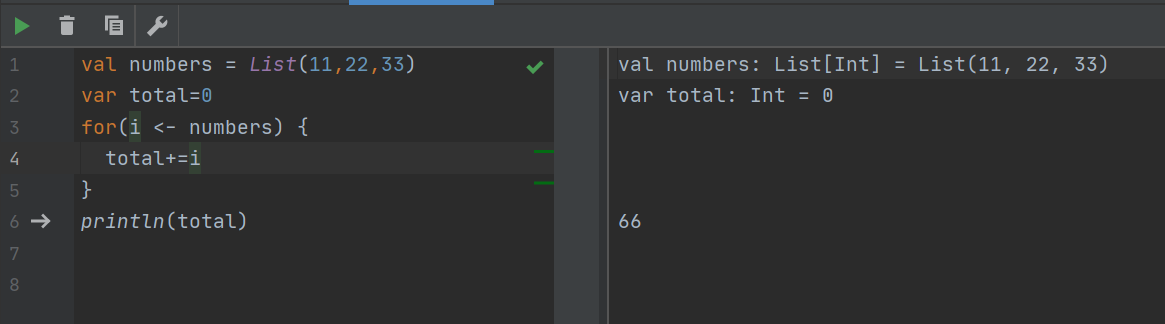
**Задача 1.** Бұл есепте **value мәні 20**, ал **value1 мәні 0.0** шығады. Себебі біріншісі *Some-ге* сәйкес келеді және Some ішіндегі санды қайтарады. Екіншісі *None - ға* сәйкес келеді, егер None – ге сәйкес келсе 0.0 мәнін қайтар деп есеп шартын берді, сондықтан 0.0 мәнін қайтарады.



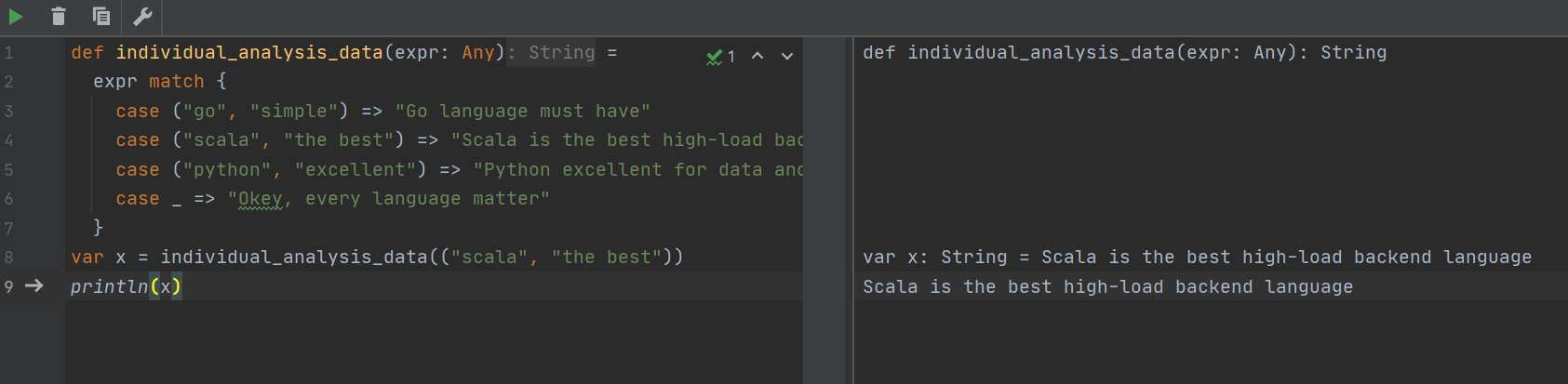
**Задача 2.** Бұл есепте списоктың ұзындығын, екі еселенген списокты, керісінше жазылған списокты және 3-ке бөлінетін сандарды фильтровать ететін списокты шығарады. Оларды шығару үшін мен List классының әдістерін қолданып шығардым.



**Задача 3.** Бұл есепте шығысты табу қажет. Списоктың ішінде үш сан бар. total параметрінің мәнін 0 деп алдық. Сосын циклдің ішіне total параметріне списоктағы әр объектіні жеке қосып алдық. Осылайшы біз списоктың суммасын табамыз. Бізде Шығыс *total* параметрі болады. Оның мәні 66-ға тең.



**Задача 4.** Бүл есепте individual\_analysis\_*data* методы берілген. Мен оның параметрін(х) жариялап алдым. Сосын оны экранға шығардым.



**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;

**Гитхабтан жаңа репозитории құрдым. Оны *kaznu\_oryngul\_bekbergen* деп атадым. Проектті өз аккаунтымнан аштым. Репозиторииді общедоступный(public) қылып құрдым.**