Қазақстан Республикасы білім және ғылым министрлігі

**Әл - Фараби атындағы Қазақ Ұлттық Университеті**



**СӨЖ**

**Пән: Интеллектуальный анализ данных**

**Орындаған: Нұсқабек Аружан Бекжанқызы**

**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 ?**

Классы в Scala - это статические шаблоны, экземпляры которых могут быть созданы во многих объектах во время выполнения.

Объект - это класс, но у него уже есть экземпляр, поэтому мы не можем вызвать new ObjectName. С другой стороны, Класс является просто типом, и он может быть экземпляром, вызывая new ClassName().

Объект равен статическому классу в Java до некоторого расширения, статические характеристики означают, что статический класс не должен создавать объект при вводе в JVM, он может быть использован им непосредственно именем класса. В scala нет концепции static. Таким образом, scala создает объект singleton для предоставления точки входа для выполнения вашей программы. Если вы не создадите одноэлементный объект, ваш код будет скомпилирован успешно, но не будет выводить какой-либо вывод.

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

**Особенности Trait в Scala:**

* **Синтаксис наследования. В языке scala существует два ключевых слова для выражения наследования extends и width, при этом эти ключевые слова не эквивалентны инструментам java.**
* **Анонимное смешивание. Также возможно создавать анонимные классы, которые смешиваются по нескольким признакам.**
* **Трейт может наследовать класс. Трейт может наследовать другие трейты и, таким образом, использовать, реализовывать, уточнять и переопределять унаследованные элементы. Но трейты также может унаследовать поведение от класса и делать то же самое со своими членами**
* **Пересечение типов. Такой тип следует читать как то, что является экземпляром как A, так и B. К сожалению, использование ключевого слова with в таком синтаксисе, сбивает с толку, потому что его значение существенно отличается от синтаксиса наследования. В частности: A и B могут быть произвольными типами, а не только классами или чертами порядок типов не имеет значения**

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

**Разница между обычным class и case class**

* **Классы кейсов могут быть сопоставлены по шаблону**
* **Классы Case автоматически определяют hashcode и equals**
* **Классы Case автоматически определяют методы getter для аргументов конструктора.**
* **Классы Case можно рассматривать как простые и неизменяемые объекты хранения данных, которые должны зависеть исключительно от их аргументов конструктора .**

**Эта функциональная концепция позволяет нам**

* **использовать компактный синтаксис инициализации (Node(1, Leaf(2), None)) )**
* **разложить их с помощью сопоставления шаблонов**
* **имеют сравнения равенства неявно определенные**
* **В сочетании с наследованием классы case используются для имитации алгебраических типов данных .**
* **Если объект выполняет вычисления с сохранением состояния внутри или проявляет другие виды сложного поведения, он должен быть обычным классом.**

**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 {

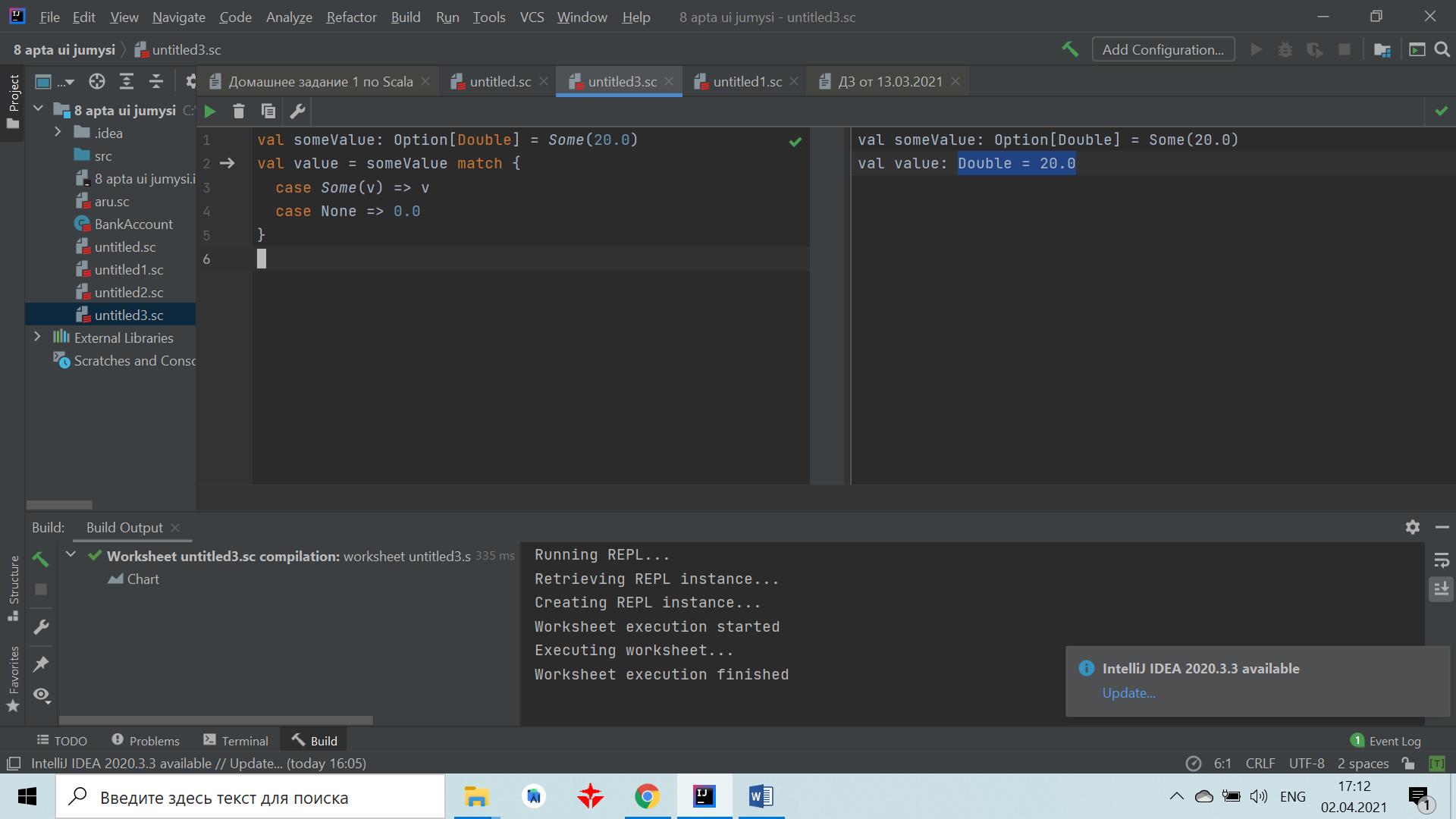
case Some(v) => v

case None => 0.0

}

Найти значение “value”, чему равен?

Жауап: 20.0



val noValue: Option[Double] = None

val value1 = noValue match {

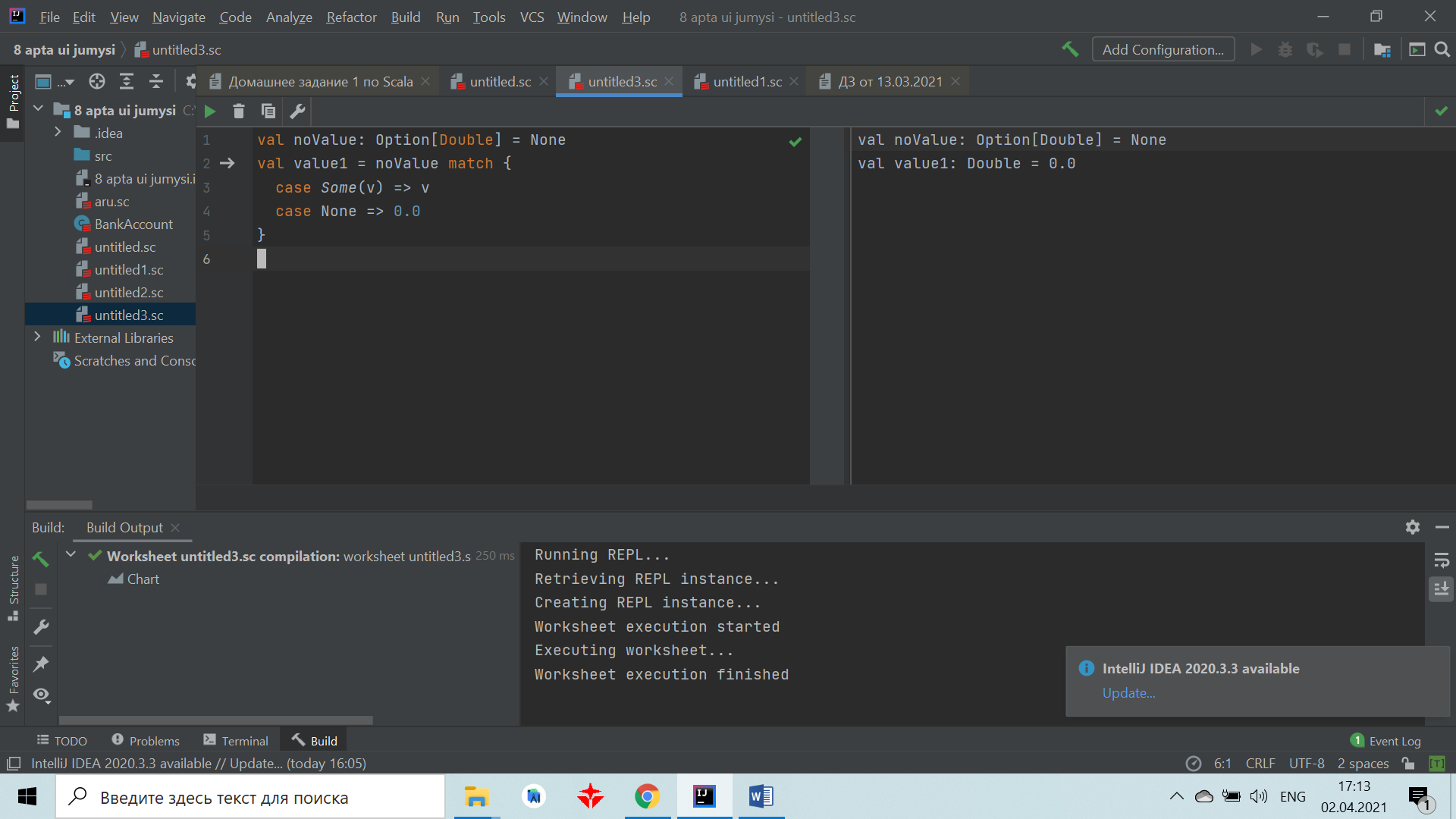
case Some(v) => v

case None => 0.0

}

Найти значение “value1”, чему равен?

Жауап: 20.0



**Задача 2.**

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

нужно найти:

- длину списка

- удвоить значение элементов списка

- перевернуть список

- отфильтруйте все значения в списке, кратные 3

Жауап:

val kaznuLabs = List(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

kaznuLabs.length

kaznuLabs.map { v =>

v \* 2

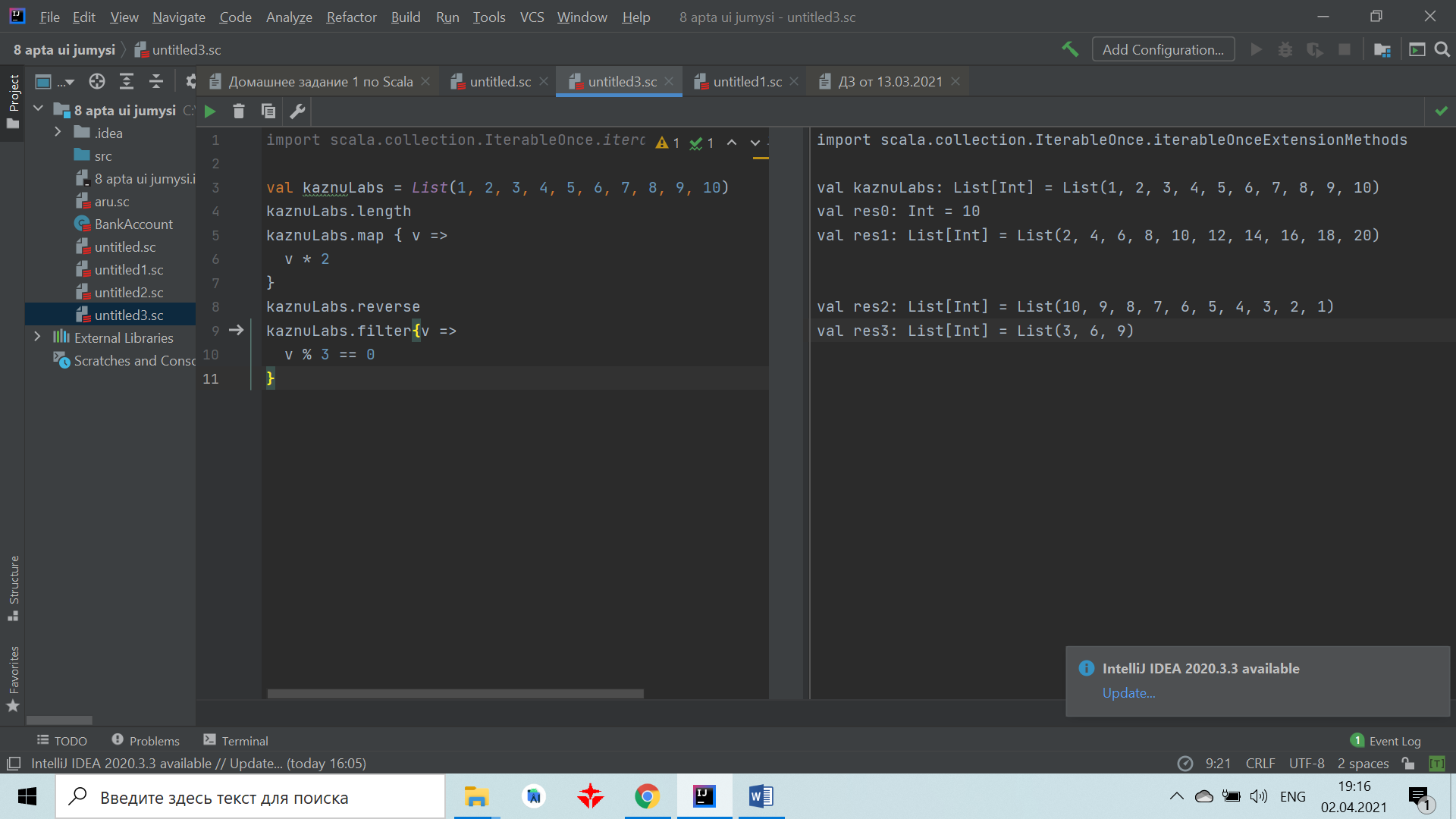
}

kaznuLabs.reverse

kaznuLabs.filter{v =>

v % 3 == 0

}



**Задача 3.**

Вычислите правильный выход:

val numbers=List(11,22,33)

var total=0

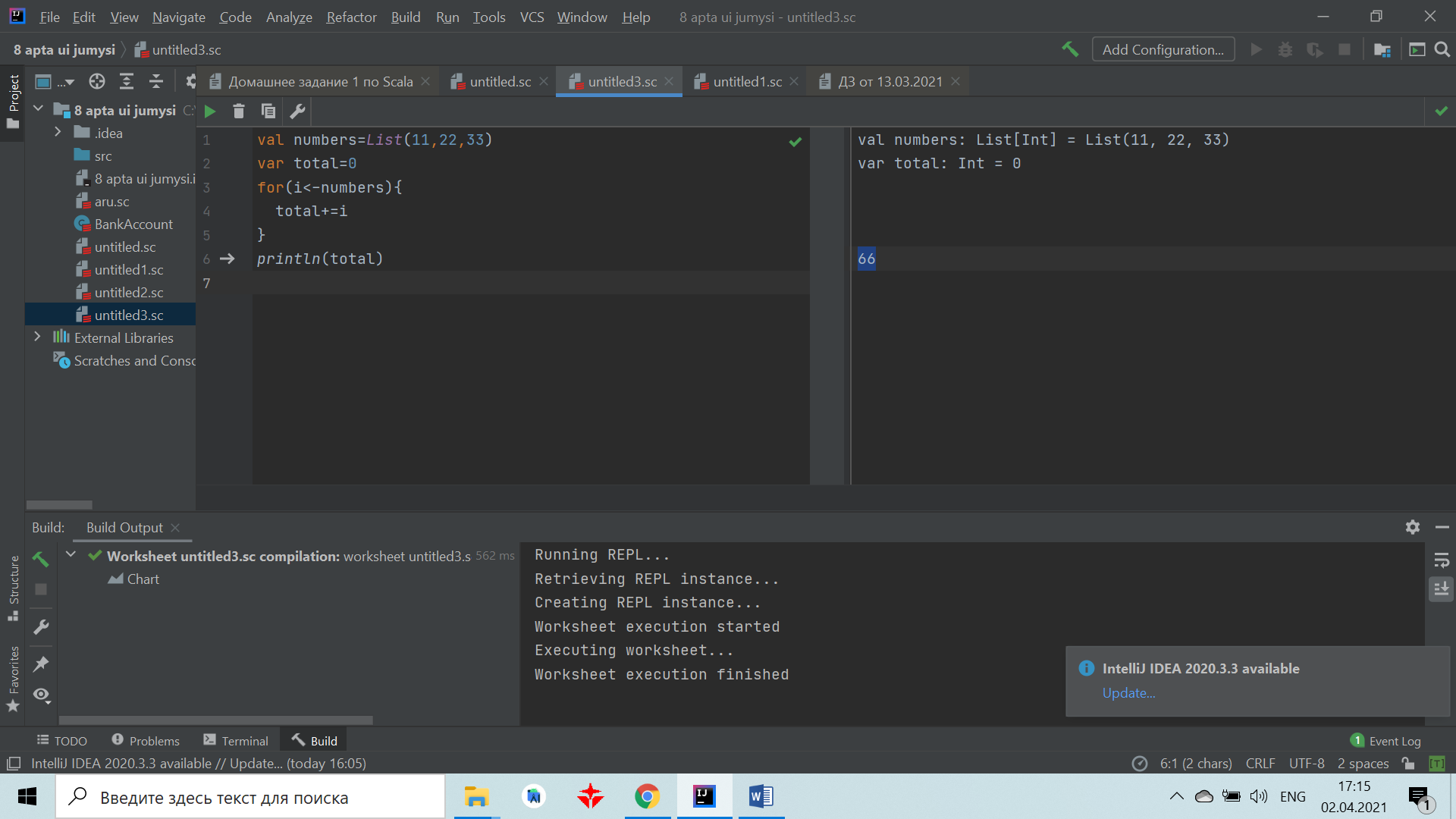
for(i<-numbers){

total+=i

}

println(total)

Жауап: 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 {

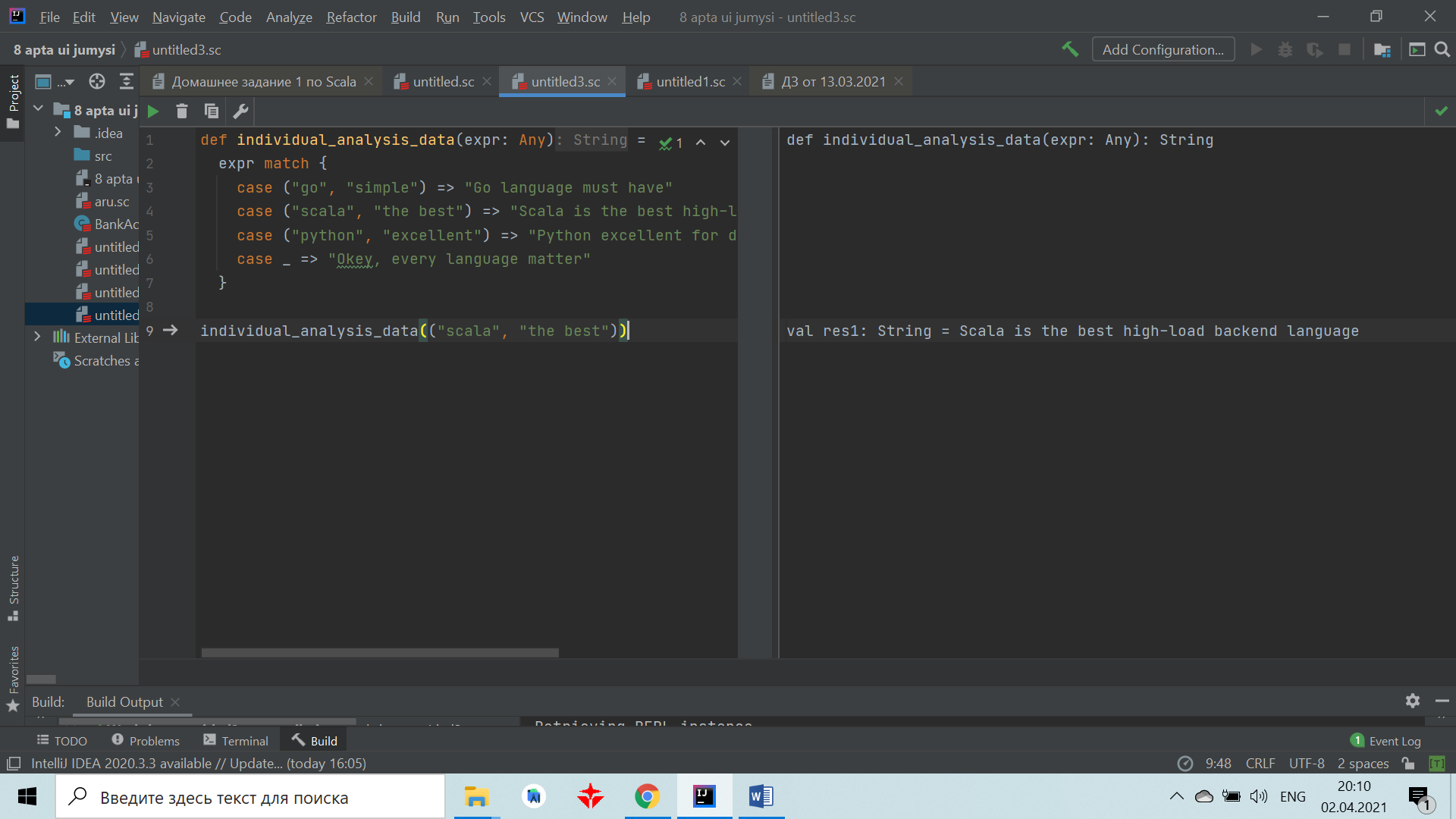
case ("go", "simple") => "Go language must have"

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"

}



**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;