МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

***Факультет информационных технологий и робототехники***

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 2**

по дисциплине: “Программирование на языке Java”

|  |  |  |
| --- | --- | --- |
| Выполнили: |  | студенты группы 10702121  Фатющенков С.О.  Шелег А.О. |
| Принял: |  | Сидорик В. В. |
|  |  |  |

Минск 2023

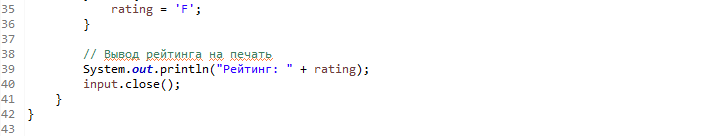
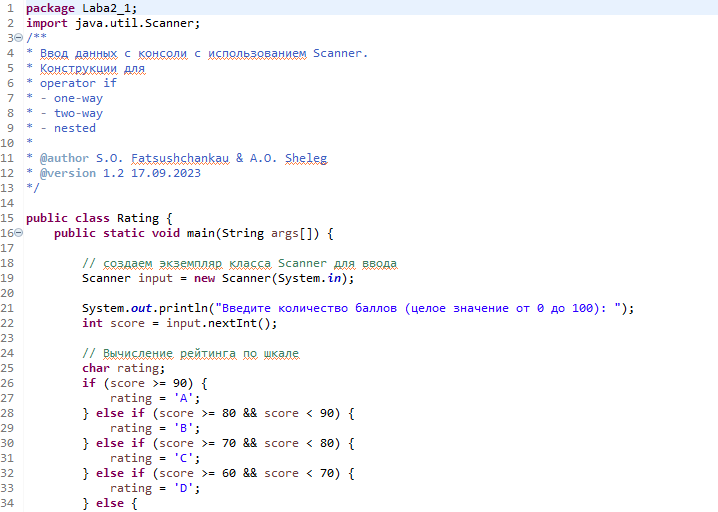
# Лабораторная работа №2.

## PRACTICE\_IfTestSmall

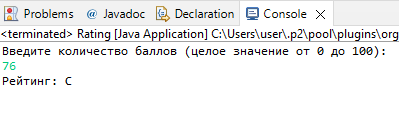
### Задание:

### 

### Код программы:



### Результат:



### Контрольные вопросы:

1. В каких случаях используется if оператор?

Оператор if используется в Java для выполнения определенного блока кода, только если некоторое условие истинно. Он является одной из основных конструкций управления потоком выполнения программы. Вот основные случаи, в которых используется оператор if:

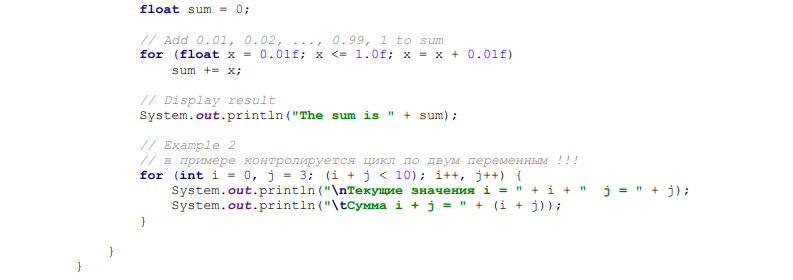
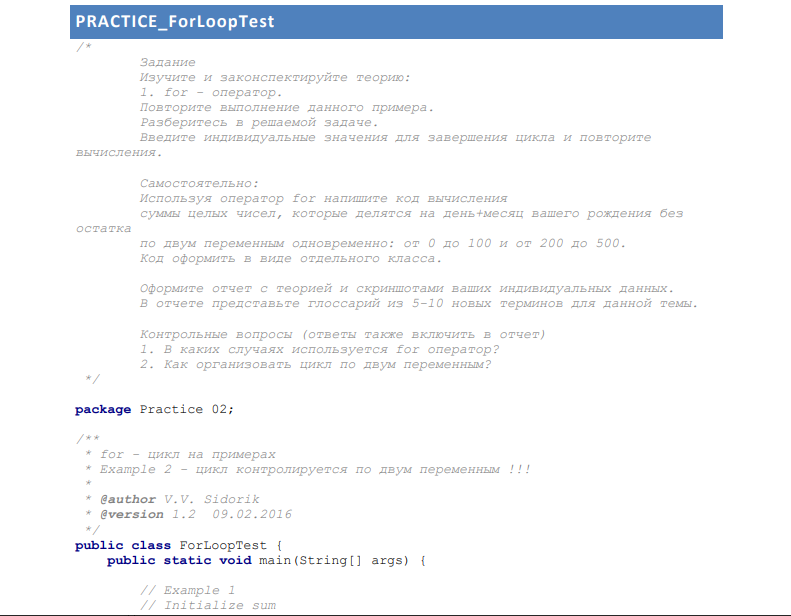
1. Выполнение кода, если некоторое условие истинно.

2. Выполнение альтернативного блока кода, если условие ложно. Для этого используется оператор else.

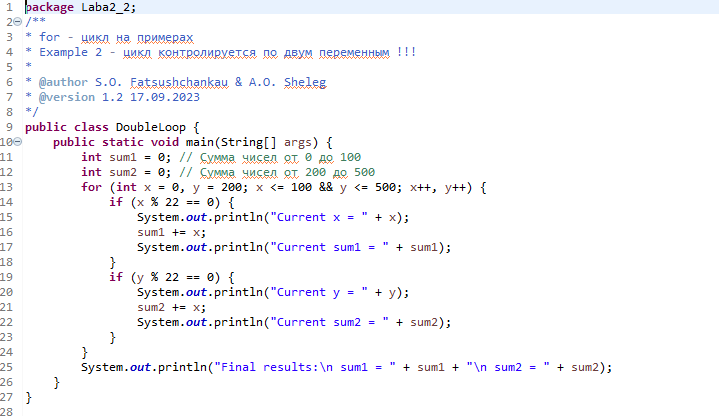
3. Проверка нескольких условий с помощью операторов else if.

## PRACTICE\_ForLoopTest

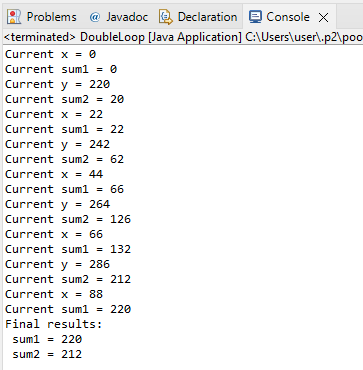
### Задание:



### Код программы:



### Результат:



### Контрольные вопросы

1. В каких случаях используется for оператор?

1. Для выполнения цикла с заранее известным числом итераций.

2. Для прохождения по элементам массива или коллекции.

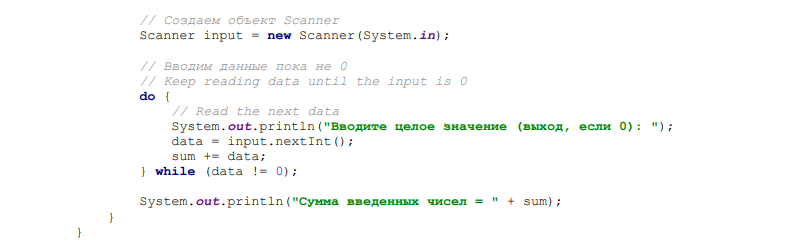
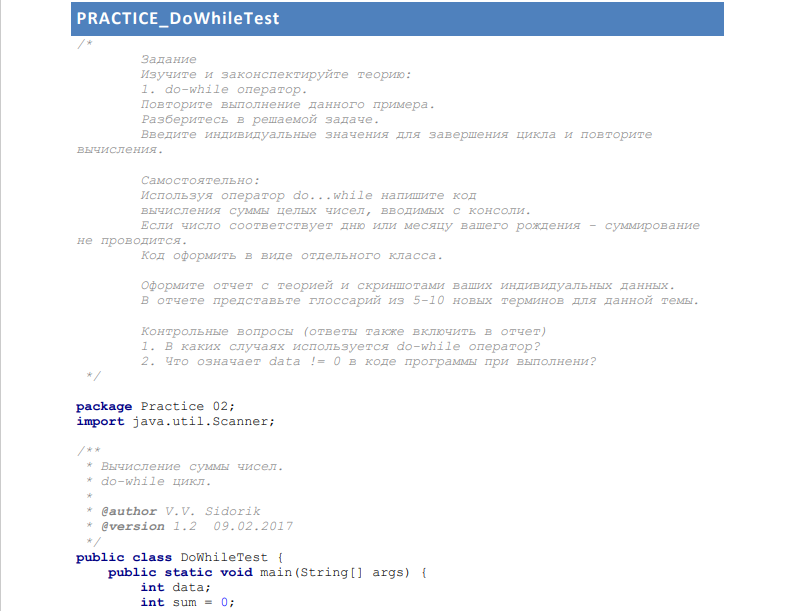
3. Для использования вложенных циклов. Можно использовать несколько операторов for для создания вложенной структуры циклов.

1. Как организовать цикл по двум переменным?

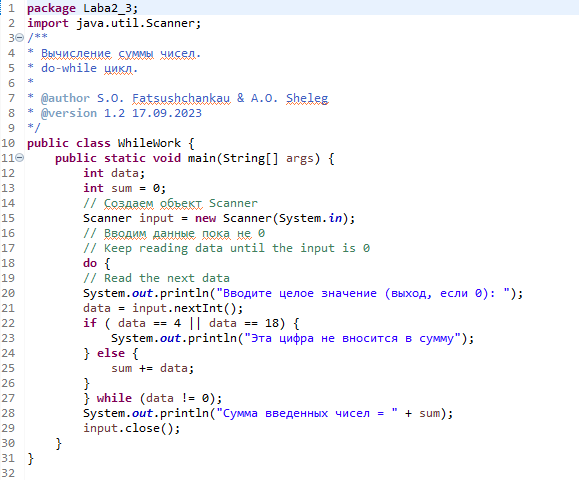
Для этого достаточно объявить вторую переменную в теле самого цикла

## PRACTICE\_DoWhileTest

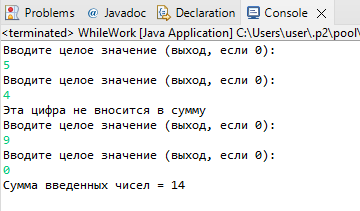
### Задание:



### Код программы:



### Результат:



### Контрольные вопросы:

1. В каких случаях используется do-while оператор?

1. Когда нужно выполнить блок кода хотя бы один раз, а затем проверять условие для продолжения цикла. То есть, код внутри do-while будет выполнен как минимум один раз, даже если условие цикла изначально не выполняется.

2. Когда вы хотите, чтобы блок кода выполнялся до тех пор, пока условие цикла истинно. В этом случае код внутри do-while будет продолжать выполняться, пока условие цикла верно.

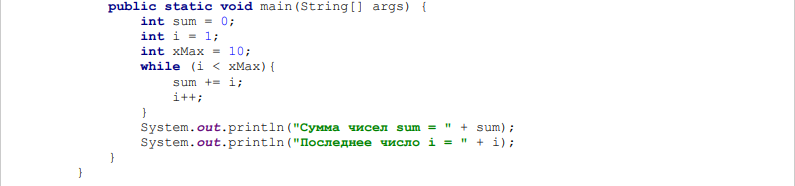
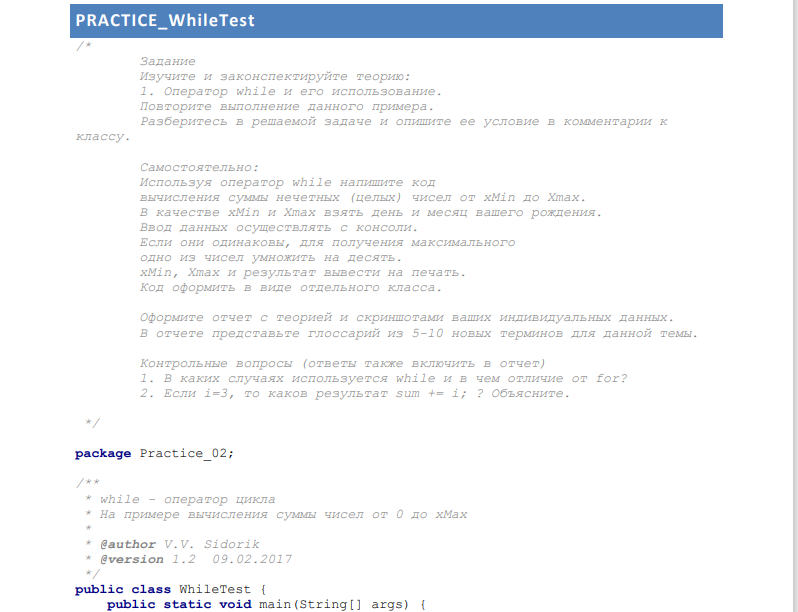
3. Когда нужно обрабатывать пользовательский ввод и продолжать выполнение кода, пока пользователь не введет конкретную команду для завершения цикла.

1. Что означает data != 0 в коде программы при выполнени?

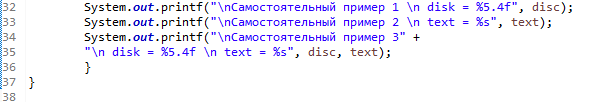
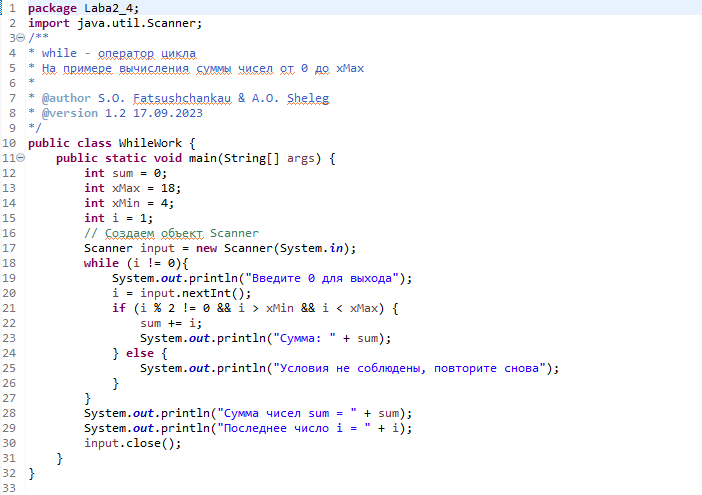
Выражение while (data != 0) в коде программы означает, что цикл while будет выполняться до тех пор, пока условие data != 0 истинно. В нашем случае это специально используется для того, чтобы польователь мог вводить данные до тех пор, пока он самостоятельно не захочет замкнуть цикл.

## PRACTICE\_WhileTest

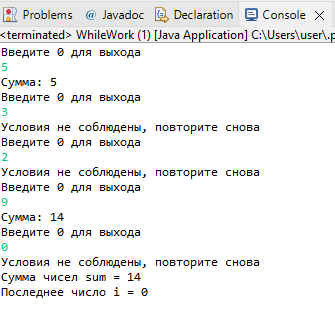
### Задание:



### Код программы:



### Результат:



### Контрольные вопросы:

1. В каких случаях используется while и в чем отличие от for?

1. Условие проверки: while проверяет условие перед каждой итерацией, в то время как for задает условие в самом начале цикла.

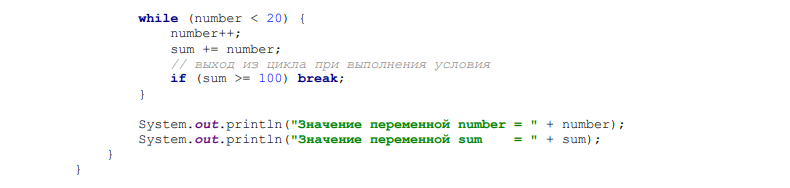
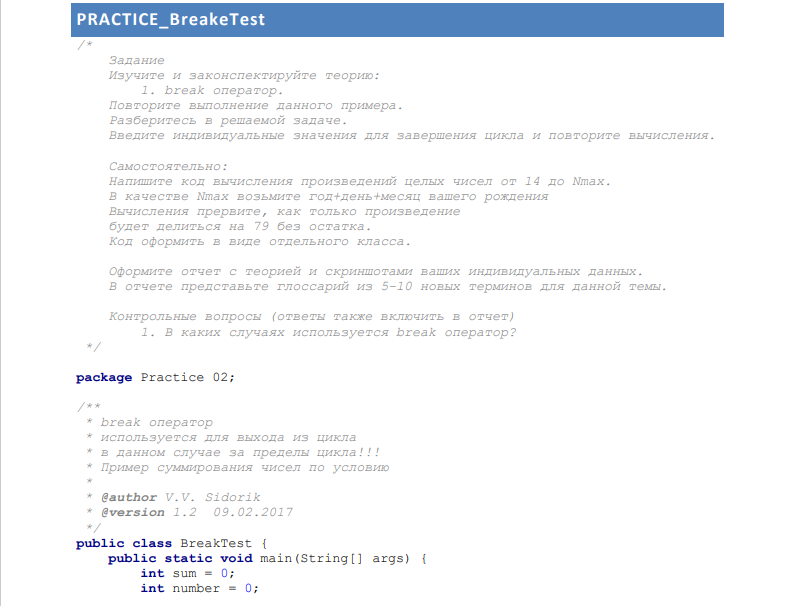
2. Инициализация и изменение счетчика: for позволяет инициализировать счетчик, определять условие его изменения и изменять его значение на каждой итерации, что полезно при переборе массивов или коллекций. В while возможно использование счетчика, но его изменение должно быть выполнено внутри блока кода цикла.

2. Если i=3, то каков результат sum += i; ? Объясните.

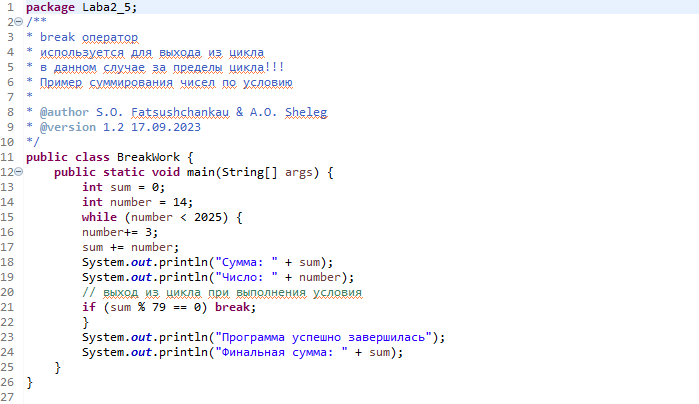
Ответ: 42. Если изначальное число ` i ` = 3 и с каждым циклом он будет увеличиваться, то переменная `sum` будет равна сумме чисел с 3 до 9. 10 не будет учитываться, т.к. инкримент переменной `i` идёт позже проверки условия цикла.

## PRACTICE\_BreakeTest

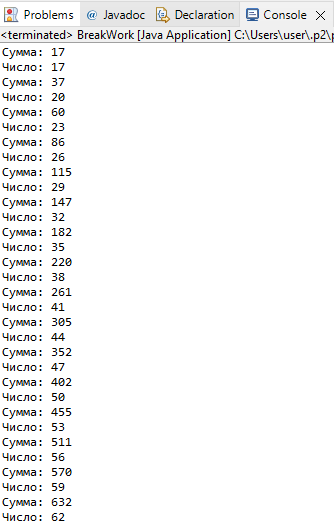
### Задание:



### Код программы:



### Результат:



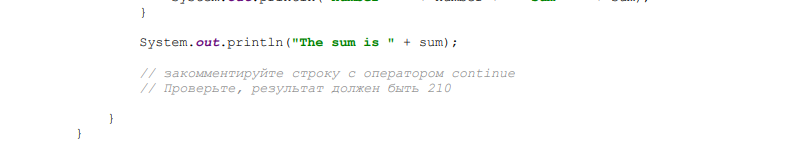
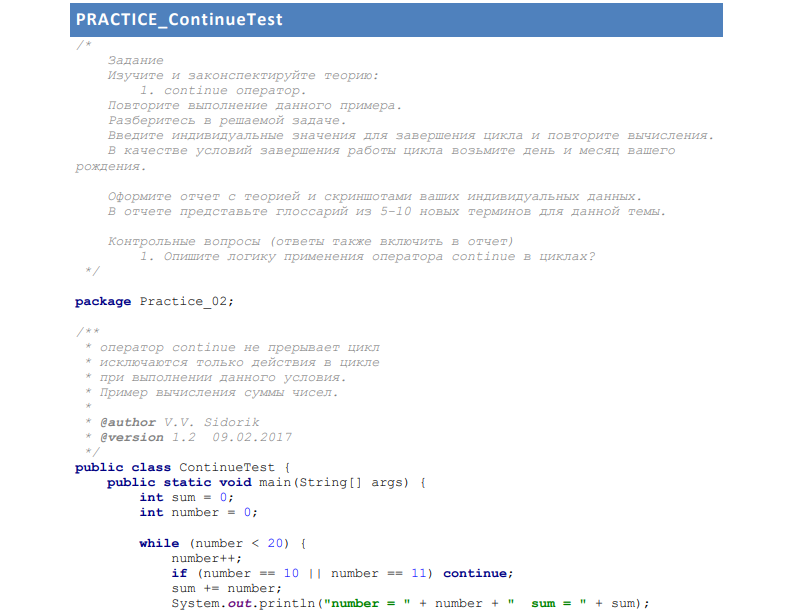


### Контрольные вопросы:

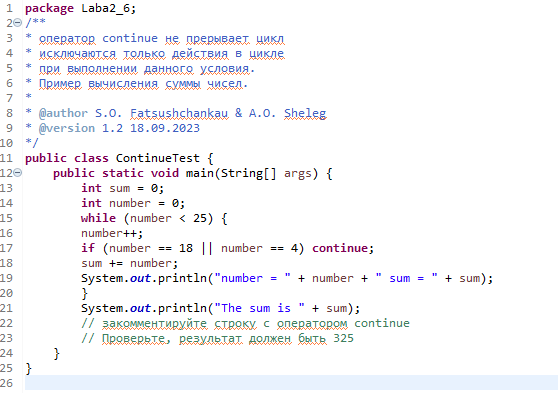
1. В каких случаях используется break оператор?
2. Внутри цикла: break прерывает выполнение цикла и переходит к следующей инструкции после цикла. Это может быть полезно, если достигнуто определенное условие и необходимо выйти из цикла раньше, чем он закончится.
3. Внутри switch оператора: break используется для выхода из switch-блока. После выполнения инструкции с ключевым словом break, выполнение программы продолжается после switch-блока.

## PRACTICE\_ContinueTest

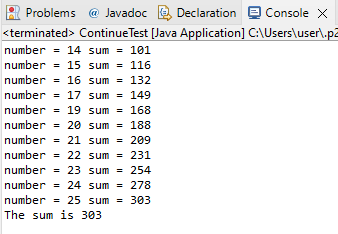
### Задание:



### Код программы:



### Результат:



### Контрольные вопросы:

1. Опишите логику применения оператора continue в циклах?

Оператор continue в языке Java используется для пропуска текущей итерации цикла и перехода к следующей итерации.

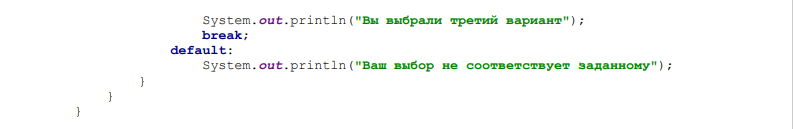
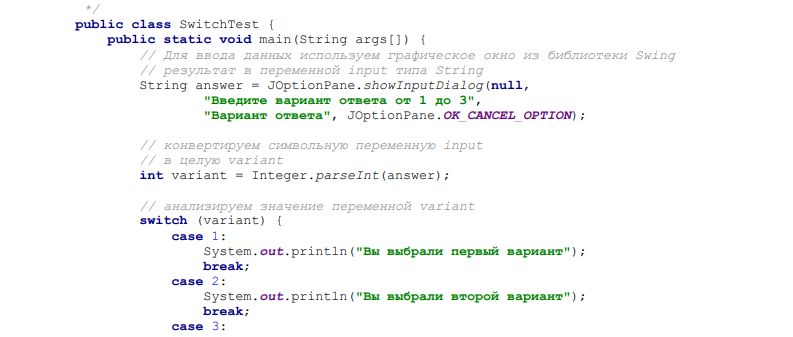
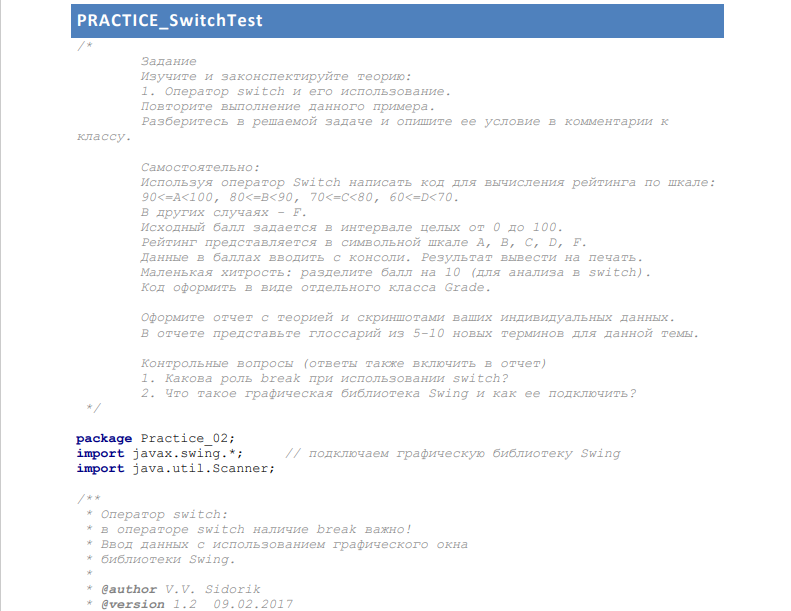
Когда оператор continue встречается внутри цикла, все последующие инструкции в теле цикла после оператора continue не выполняются, и управление переходит к следующей итерации цикла.

1. В циклах for и while: когда оператор continue встречается внутри цикла, он пропускает все последующие инструкции в теле цикла и переходит к следующей итерации.

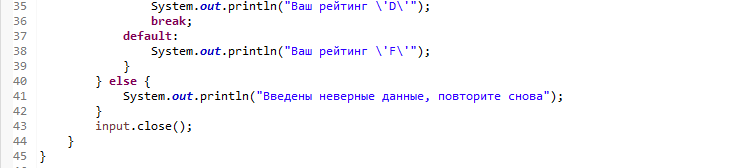
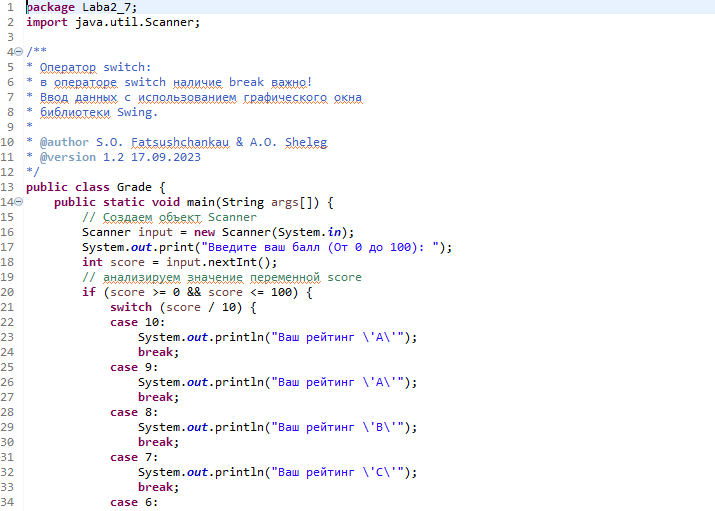
2. В цикле do-while: при использовании оператора continue в цикле do-while, выполнение программы переходит к условию проверки условия цикла, и, если условие выполнено, происходит следующая итерация цикла. Если условие не выполняется, цикл завершается.

## PRACTICE\_SwitchTest

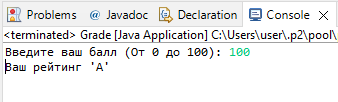
### Задание:



### Код программы:



### Результат:



### Контрольные вопросы:

1. Какова роль break при использовании switch?

1. Остановка выполнения кода: Когда в теле switch-блока выполняется инструкция "break", выполнение программы останавливается и управление передается к следующей инструкции сразу за пределами switch-блока. Это предотвращает выполнение кода и case-инструкций под ним, так что только один вариант будет выполнен.

2. Предотвращение "проваливания" (fall-through): Если в теле switch-блока отсутствует инструкция "break" после каждого case, то выполнение не останавливается и код будет выполняться последовательно до первого "break" или пока не завершится блок switch. Это называется "проваливанием" (fall-through) и может использоваться, когда несколько вариантов должны выполниться одним и тем же кодом.

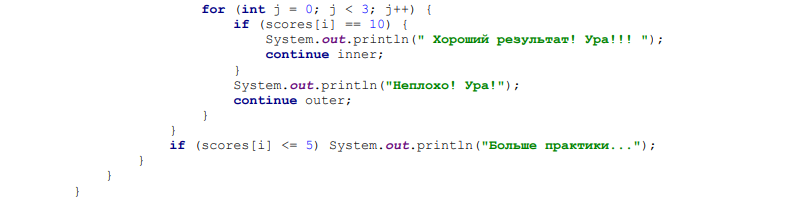
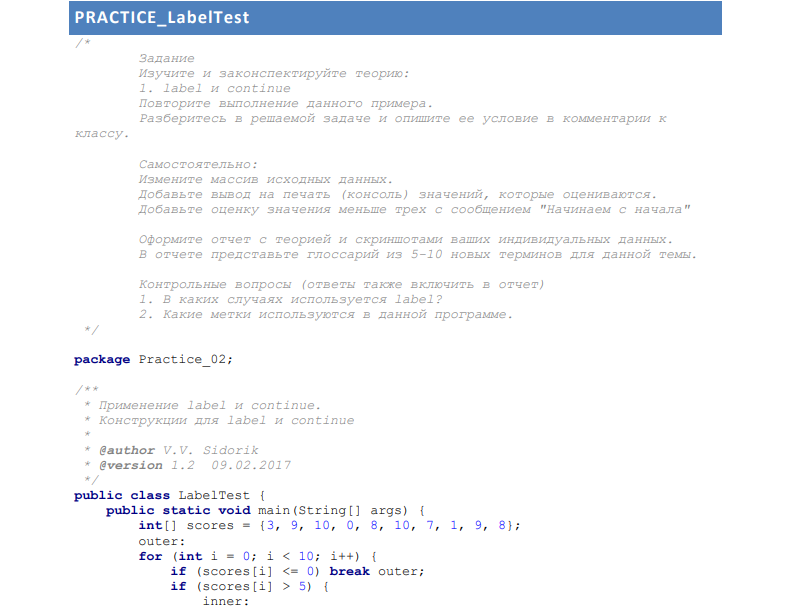
1. Что такое графическая библиотека Swing и как ее подключить?

Графическая библиотека Swing - это набор классов и компонентов, предоставляемых языком программирования Java для создания графического пользовательского интерфейса (GUI). Swing обеспечивает возможности для создания окон, кнопок, текстовых полей, меток, таблиц, списков и других элементов интерфейса.

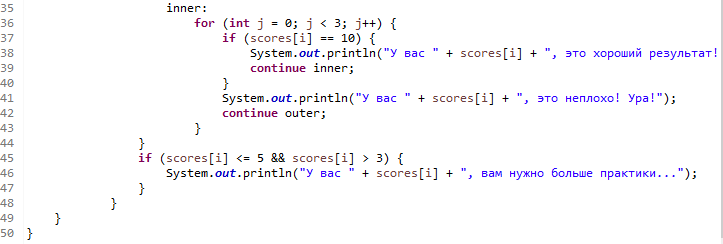
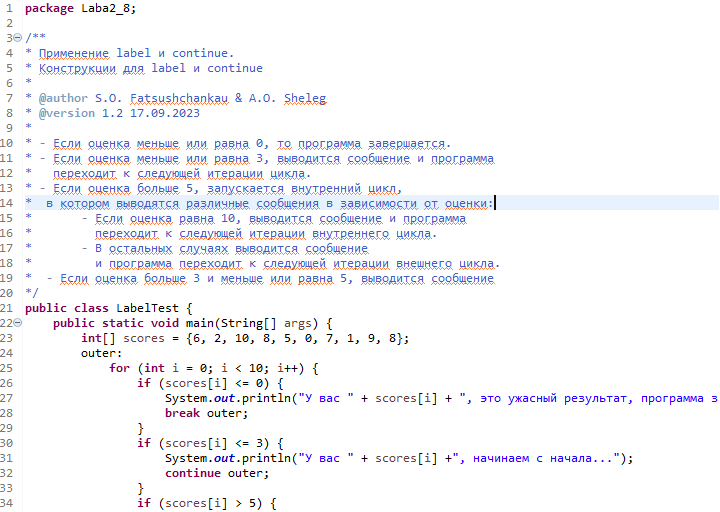
Чтобы подключить библиотеку Swing к проекту в языке Java, необходимо импортирование пакета Swing: В начале файла, где будет использоваться Swing, добавьте импорт пакета Swing, используя следующую инструкцию: import javax.swing.\*;

## PRACTICE\_LabelTest

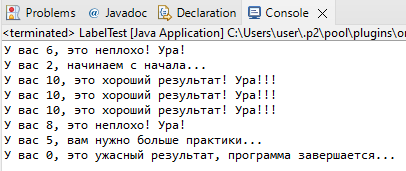
### Задание:



### Код программы:



### Результат:



### Контрольные вопросы:

1. В каких случаях используется label?

1. Метка в циклах: Метки могут быть помечены перед циклом (for, while, do-while) для указания конкретной точки перехода в программе.

2. Метки в операторе break и continue: Метки можно использовать вместе с операторами break и continue для указания точки перехода в программе.

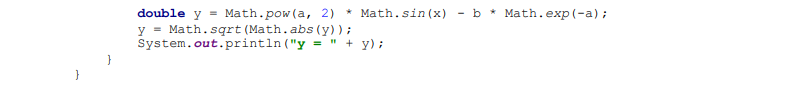
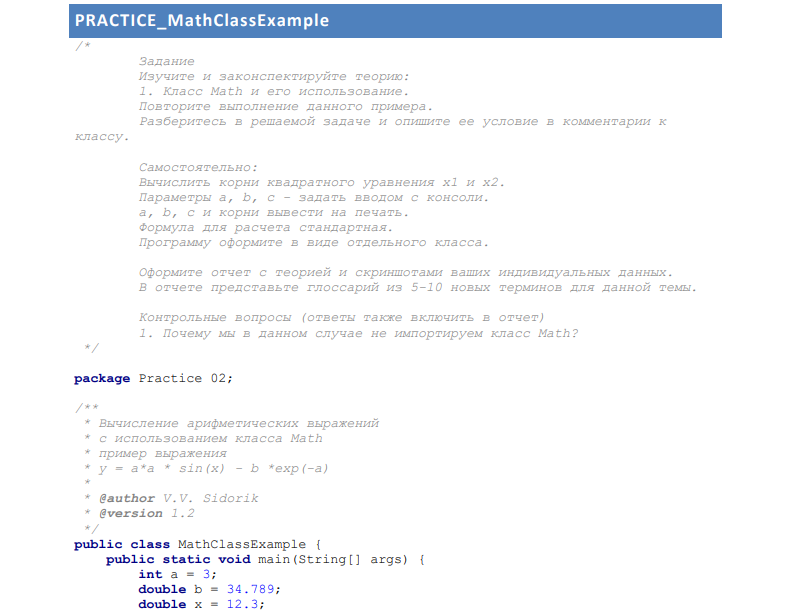
выа

1. Какие метки используются в данной программе.

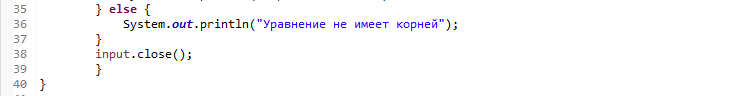
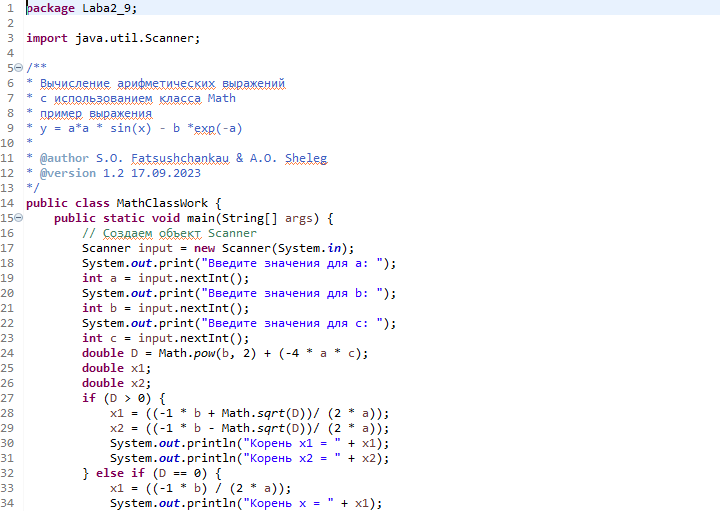
В данной программе используются метки outer и inner

## PRACTICE\_ MathClassExample

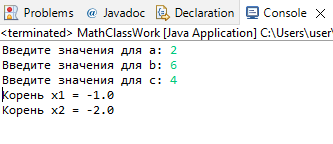
### Задание:



### Код программы:



### Результат:



### Контрольные вопросы:

1. Почему мы в данном случае не импортируем класс Math?.

В Java подключать или импортировать библиотеку Math не требуется, так как она является частью класса java.lang.Object, классы этого пакета уже подключены по умолчанию, что соответствует нашему случаю. Однако если бы в данном примере было использование методов или констант из класса Math, то мы должны были бы добавить следующий импорт в начало кода: import java.lang.Math;

## Глоссарий

* **if – оператор.**
* **for - оператор.**
* **do-while оператор.**
* **Оператор while и его использование.**
* **break оператор.**

л

* continue оператор.
* Оператор switch и его использование.
* label и continue
* Класс Math и его использование.