Министерство образования Республики Беларусь

Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем

и технологий»

**Отчёт**

по лабораторной работе №1

по дисциплине ***«Программирование на языке Java»***

Выполнили**:** студенты группы *10702121*

Алюнина П. А.

Ведров З. А.

Проверил**:**  Сидорик В. В.

Минск – 2023

Цель: ознакомление с синтаксисом, основными методами языка программирования Java.

Задачи лабораторной работы:

1. Изучить и законспектировать теорию
2. Повторить выполнения примеров заданий
3. Выполнить задания с индивидуальными данными
4. Оформить отчёт с теорией и скриншотами индивидуальных заданий
5. Составить глоссарий
6. Ответить на контрольные вопросы

**Задание №1 HellowWorld**

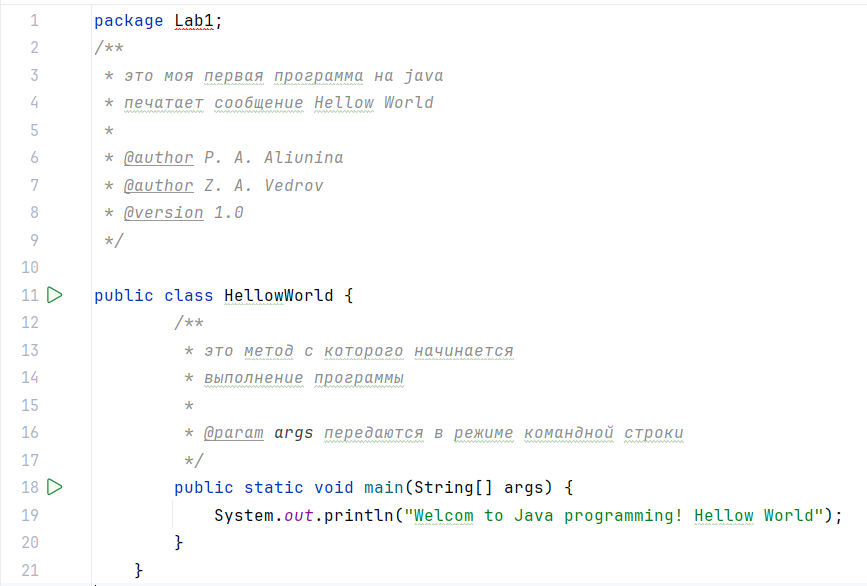
Изучите и законспектируйте теорию:

1. Метод main и его параметры.
2. Консоль, печать с использованием println

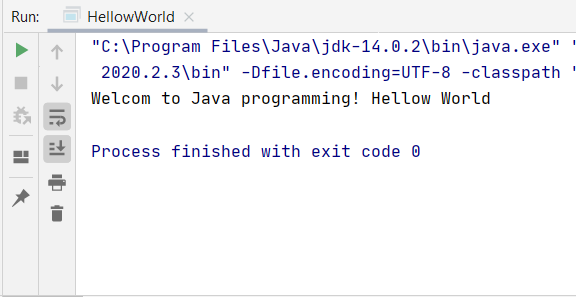
Параметры – это переменные (данные / значения), которые мы можем передать методу, который может использоваться инструкциями в методе для настройки его поведения. Основной метод всегда должен принимать массив объектов String.

Для создания потока вывода в класс System определен объект **out**. В этом объекте определен метод **println**, который позволяет вывести на консоль некоторое значение с последующим переводом курсора консоли на следующую строку.

1. **Код программы**

****

1. **Скриншоты выполнения программы**



1. **Ответы на контрольные вопросы**
2. Что такое консоль?

Консоль — интерфейс пользователя, окно для вывода системных сообщений и приёма команд; обеспечение, реализующее текстовый интерфейс пользователя.

1. Как оформляется javadoc комментарий к классу?

/\*\*

\* Описание класса.

\*

\* @тег1 Описание тега 1

\* @тег2 Описание тега 2

\* ...

\*/

public class MyClass {

// тело класса

}

1. Что такое пакет?

Как правило, в Java классы объединяются в пакеты. Пакеты позволяют организовать классы логически в наборы. По умолчанию java уже имеет ряд встроенных пакетов, например, java.lang, java.util, java.io и т.д. Кроме того, пакеты могут иметь вложенные пакеты.

1. С какого метода начинает выполняться программа?

С метода main.

1. Какие параметры у метода main?

Параметры – это переменные (данные / значения), которые мы можем передать методу, который может использоваться инструкциями в методе для настройки его поведения. Основной метод всегда должен принимать массив объектов String, это т. н. аргументы командной строки. String[] args представляет собой механизм, с помощью которого можно передавать в Java-приложение данные прямо при его запуске. Это может быть полезно во многих случаях, например, если необходимо указать при запуске некоторые параметры настройки приложения, пути к файлам, значения переменных и т.д.

**Задание №2 PrimitiveTypes**

Изучите и законспектируйте теорию:

1. Примитивные типы
2. Печать с использованием print и println

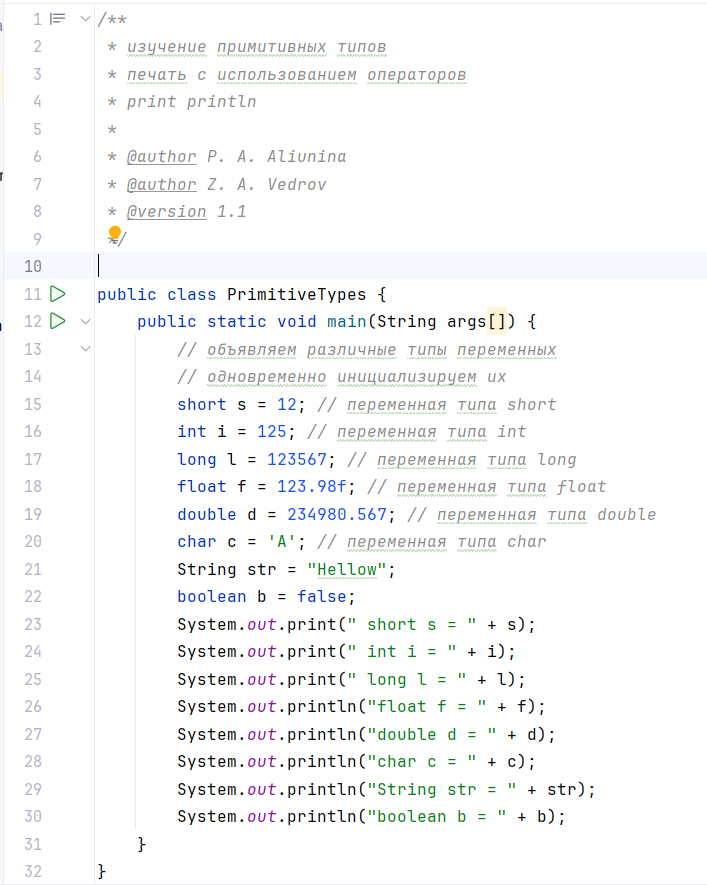
В Java есть 8 примитивных типов которые делят на 3 группы, вот они:

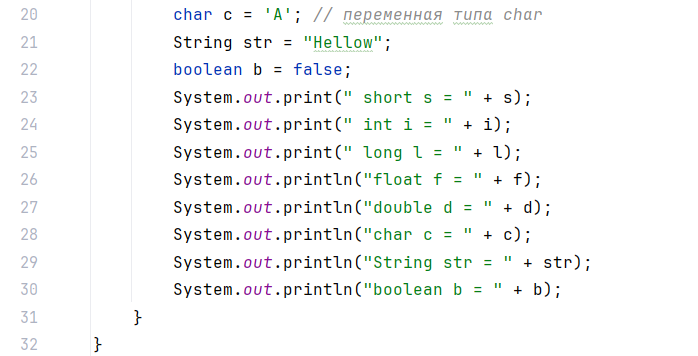
1. Целые числа — byte, short, char, int, long
2. Числа с плавающей точкой (иначе дробные) — float, double
3. Логический — Boolean

**print ()** : метод print () в Java используется для отображения текста на консоли. Этот текст передается в качестве параметра этому методу в виде строки. Этот метод печатает текст на консоли, а курсор остается в конце текста на консоли. Следующая печать происходит отсюда.

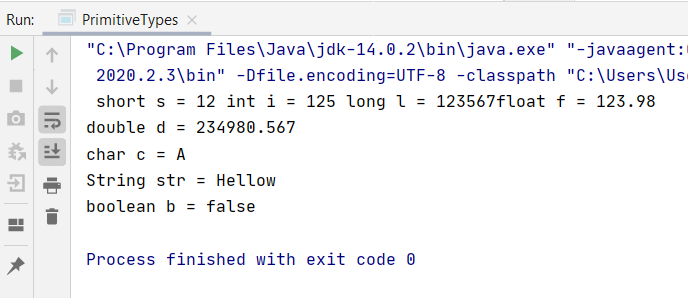
**println ()** : метод println () в Java также используется для отображения текста на консоли. Этот текст передается в качестве параметра этому методу в виде строки. Этот метод печатает текст на консоли, а курсор остается в начале следующей строки на консоли. Следующая печать происходит со следующей строки.

1. **Код программы**





1. **Скриншоты выполнения программы**



1. **Ответы на контрольные вопросы**
2. Перечислите примитивные типы

Целые числа — byte, short, char, int, long. Числа с плавающей точкой (иначе дробные) — float, double. Логический — boolean.

1. Как объявляется примитивный тип переменной?

Например: int x; double y = 34.3; boolean = false; и т.д.

1. Что такое инициализация переменной?

Инициализация переменной – это присваивание переменной значения, которое в дальнейшем она, переменная, будет хранить, и которое можно будет использовать, путем обращения к имени переменной. Инициализация может производиться одновременно с объявлением, а может – отдельно от него.

1. В чём отличие применения print и println?

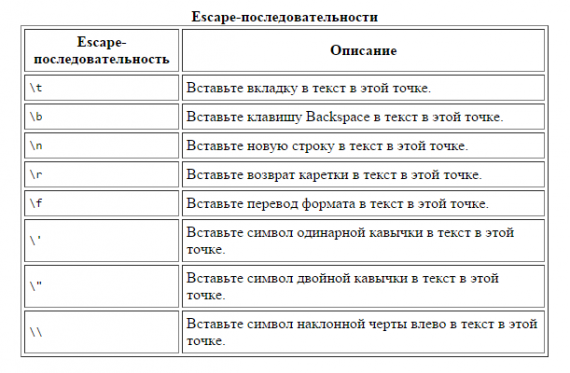
После использования println курсор переходит на новую строку, а после использования print – нет.

**Задание № 3 EscapeSequences**

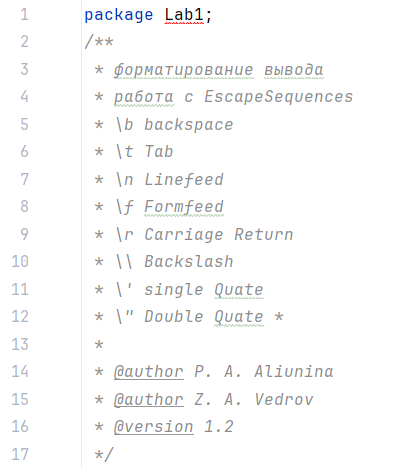
Изучите и законспектируйте теорию:

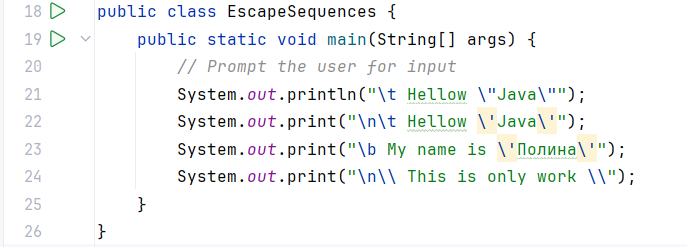
1. EscapeSequences

Символ, которому предшествует наклонная черта влево (\), является escape-последовательностью и имеет особое значение к компилятору.  Следующая таблица показывает escape-последовательности Java:

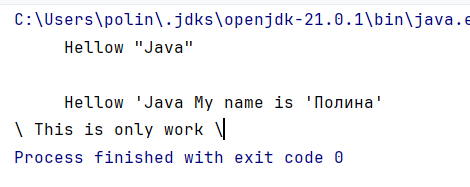


1. **Код программы**

****

****

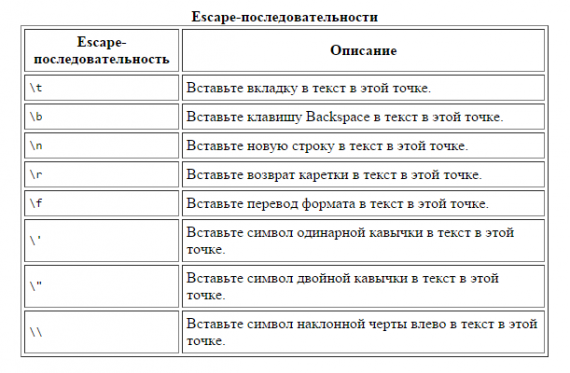
1. **Скриншот выполнения программы**



1. **Ответы на контрольные вопросы**
2. Для чего предназначены EscapeSequences?

Необходимость логического кодирования текстовых знаков возникла от того, что некоторые из них в программах, работающих с текстом, имеют специальное (управляющее) значение. Для того, чтобы использовать эти знаки непосредственно как текст, их кодируют специфическими для каждой системы обозначения ([программы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)) способами (управляющими последовательностями).

1. Перечислите основные EscapeSequences?



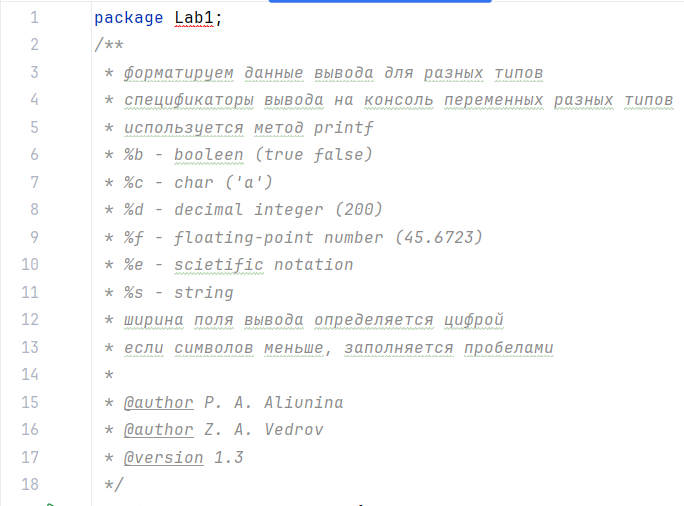
**Задание № 4 FormatOutput**

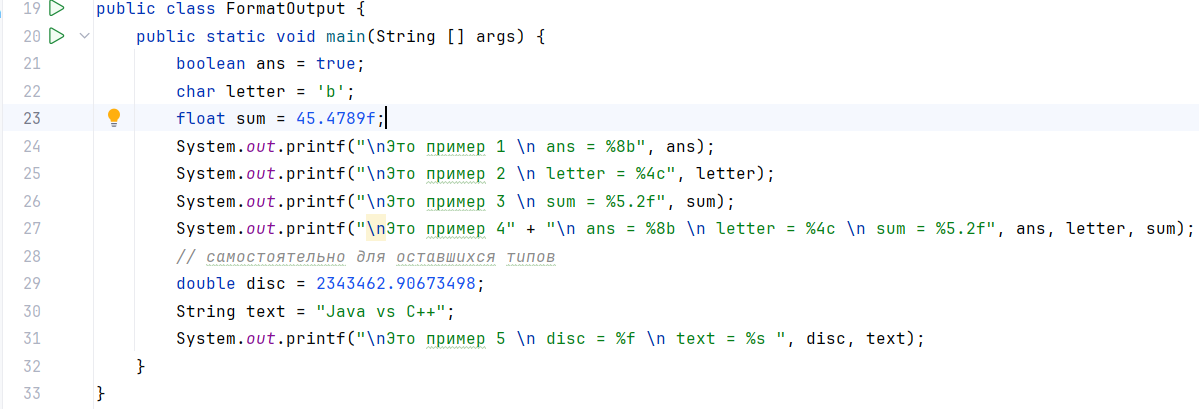
Изучите и законспектируйте теорию:

1. Форматирование при выводе на печать (консоль).

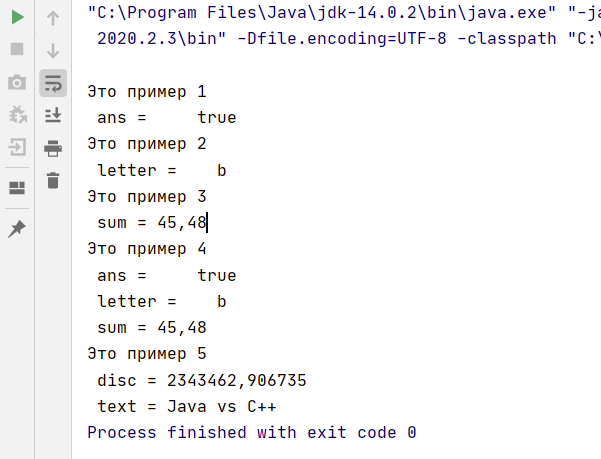
* **%x**: для вывода шестнадцатеричных чисел
* **%f**: для вывода чисел с плавающей точкой
* **%e**: для вывода чисел в экспоненциальной форме, например, 1.3e+01
* **%c**: для вывода одиночного символа
* **%s**: для вывода строковых значений

1. **Код программы**

****

****

1. **Скриншот выполнения программы**



1. **Ответы на контрольные вопросы**
2. Перечислите основные символы для форматирования вывода и их значение.

**%x**: для вывода шестнадцатеричных чисел

**%f**: для вывода чисел с плавающей точкой

**%e**: для вывода чисел в экспоненциальной форме, например, 1.3e+01

**%c**: для вывода одиночного символа

**%s**: для вывода строковых значений

Например:

int x=5;

int y=6;

System.out.printf("x=%d; y=%d \n", x, y);

1. Какой метод используется для вывода?

System.out.printf();

**Задание №5 ShortHandOperator**

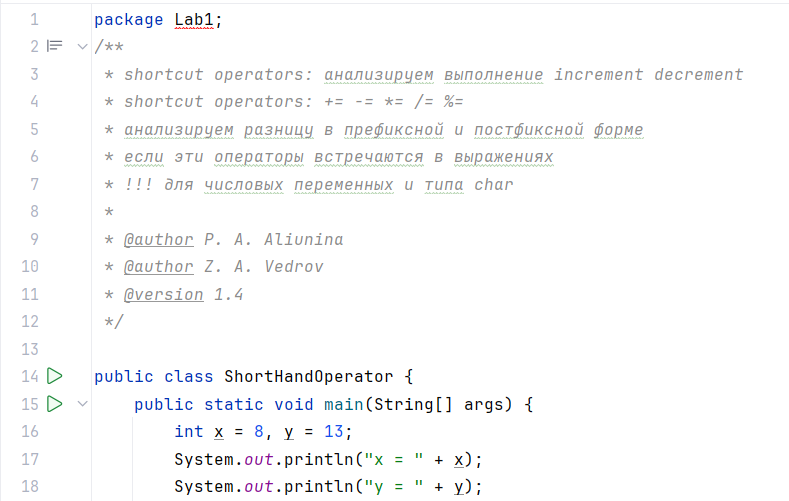
Изучите и законспектируйте теорию:

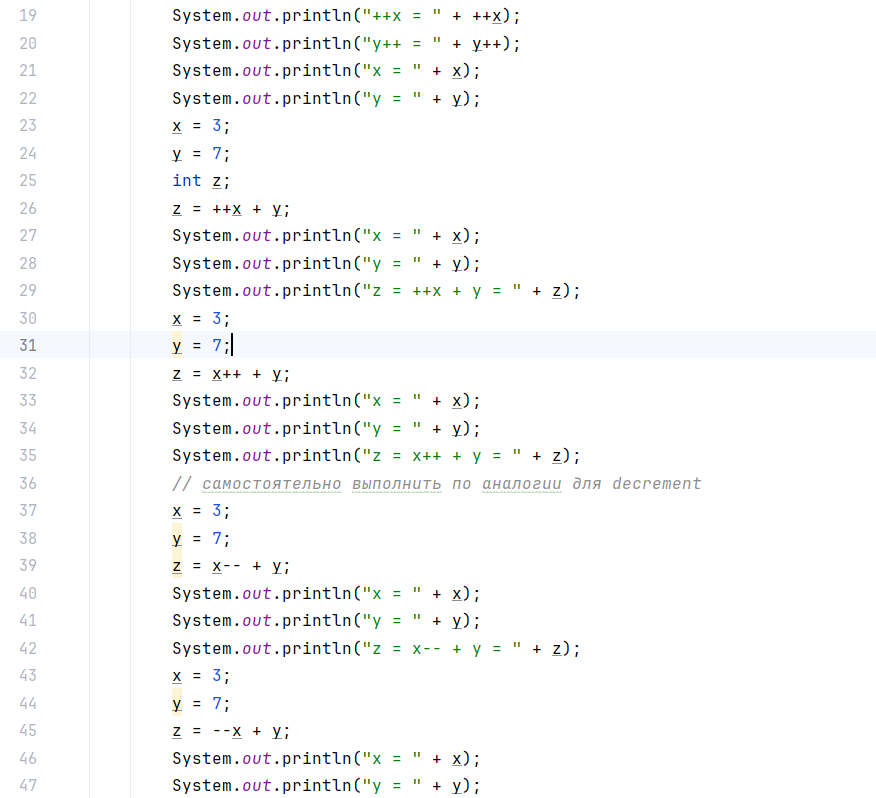
1. shortcut operators

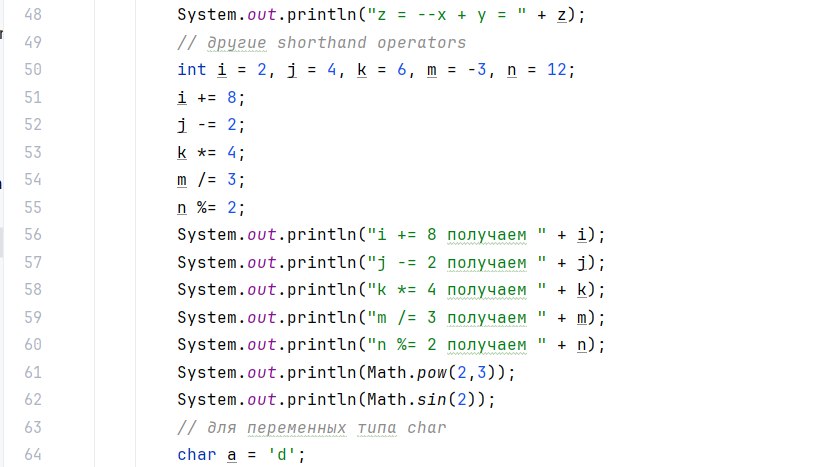
**Инкремент** — это операция во многих языках программирования, которая выполняет увеличение переменной. Чаще всего под инкрементом подразумевается увеличение переменной на 1 единицу. Обратной операцией называют декремент. **Декремент** — это уменьшение переменной. Чаще всего также на единицу. В Java инкремент и декремент — это в первую очередь унарные операторы, которые применяются к какой-либо числовой переменной. **Инкремент:** обозначается двумя знаками плюс "++" и увеличивает значение переменной на 1. **Декремент:** обозначается двумя знаками минус "--" и уменьшает значение переменной на 1.

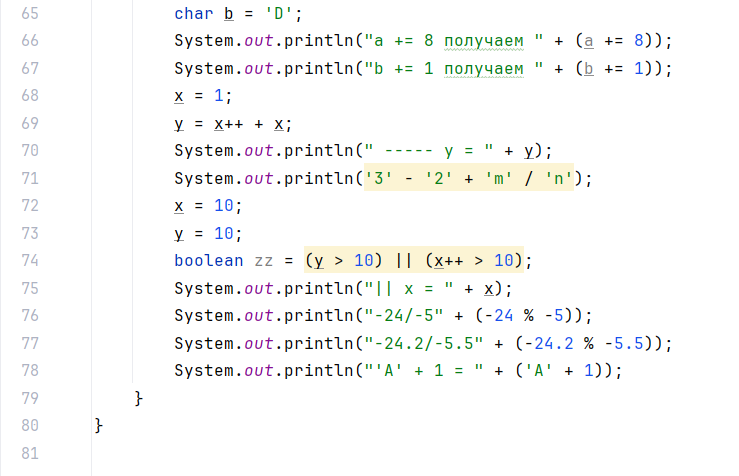
**Prefix**, или Префиксная форма, описывается перед переменной "++x или --x", изначально икрементирует или декрементирует переменную. **Postfix**, или Постфиксная форма, описывается после переменной "x++ или x--", икрементирует или декрементирует переменную после вычисления.

1. **Код программы**

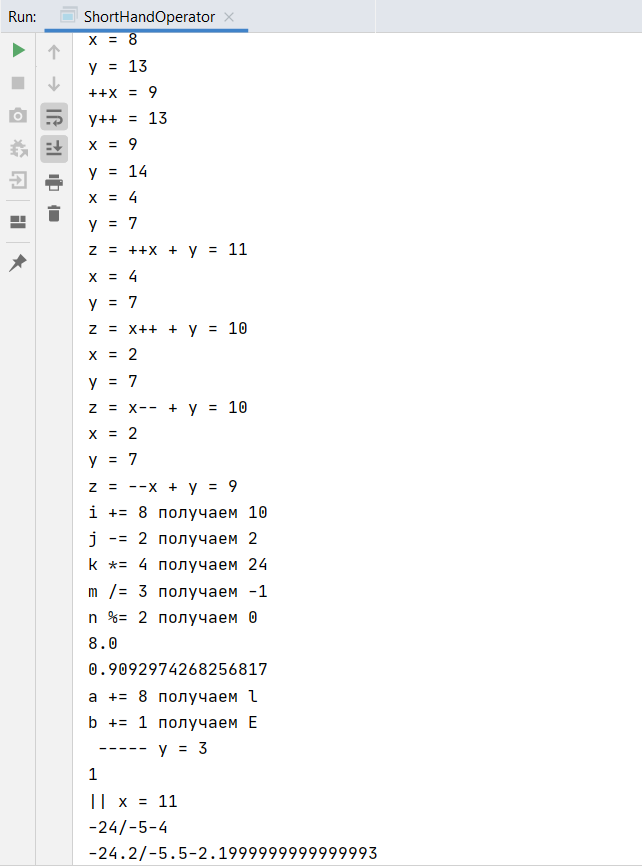








1. **Скриншоты выполнения программы**



1. **Ответы контрольные вопросы**
2. В чем различие префиксной и постфиксной форм в выражениях?

Prefix**,** или Префиксная форма, описывается перед переменной "++x или --x", изначально икрементирует или декрементирует переменную. Postfix**,** или Постфиксная форма, описывается после переменной "x++ или x--", икрементирует или декрементирует переменную после вычисления.

**Задание № 6 ConsoleInput**

Изучите и законспектируйте теорию:

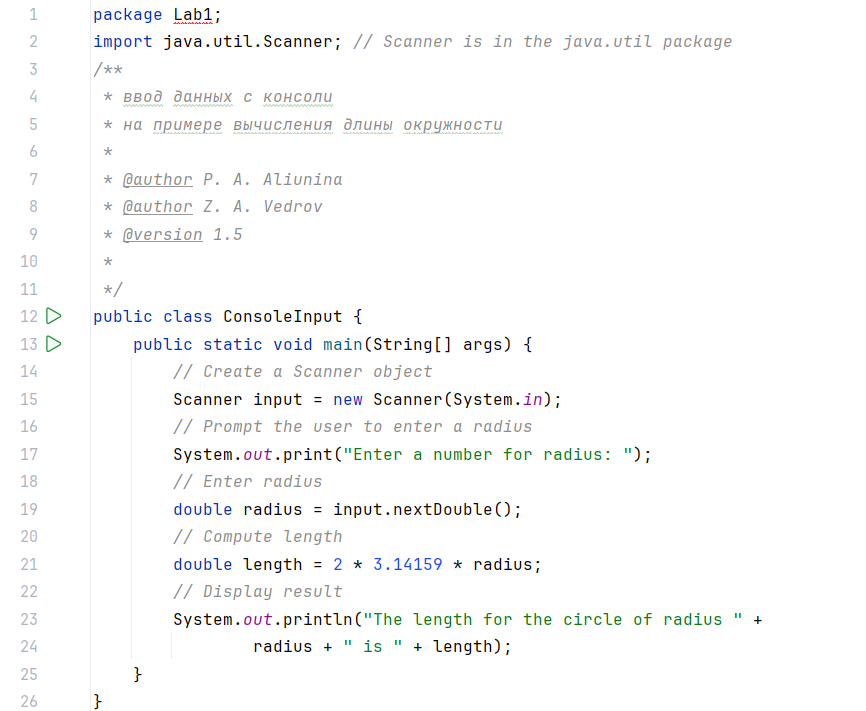
1. ввод данных с консоли

Для получения ввода с консоли в классе System определен объект **in**. Однако непосредственно через объект System.in не очень удобно работать, поэтому, как правило, используют класс **Scanner**, который, в свою очередь использует System.in.

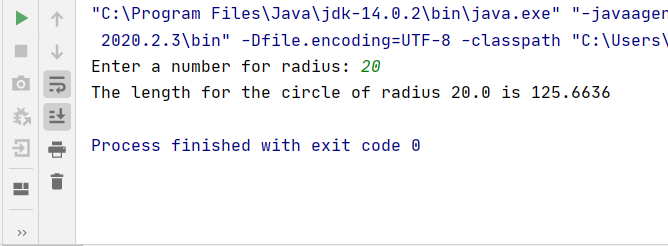
Так как класс Scanner находится в пакете **java.util**, то мы вначале его импортируем с помощью инструкции import java.util.Scanner.

Для создания самого объекта Scanner в его конструктор передается объект System.in. После этого мы можем получать вводимые значения.

1. **Код программы**



1. **Скриншот выполнения программы**



1. **Ответы на контрольные вопросы**
2. Как создается Scanner object?

Для получения ввода с консоли в классе System определен объект **in**. Однако непосредственно через объект System.in не очень удобно работать, поэтому, как правило, используют класс **Scanner**, который, в свою очередь использует System.in.

Так как класс Scanner находится в пакете **java.util**, то мы вначале его импортируем с помощью инструкции import java.util.Scanner.

Для создания самого объекта Scanner в его конструктор передается объект System.in. После этого мы можем получать вводимые значения.

1. Как ввести данные float?

Scanner sc = **new** Scanner(System.in);

**int** number = sc.nextFloat();

System.out.println("Спасибо! Вы ввели число " + number);

**Задание №8 ConditionalOperators**

Изучите и законспектируйте теорию:

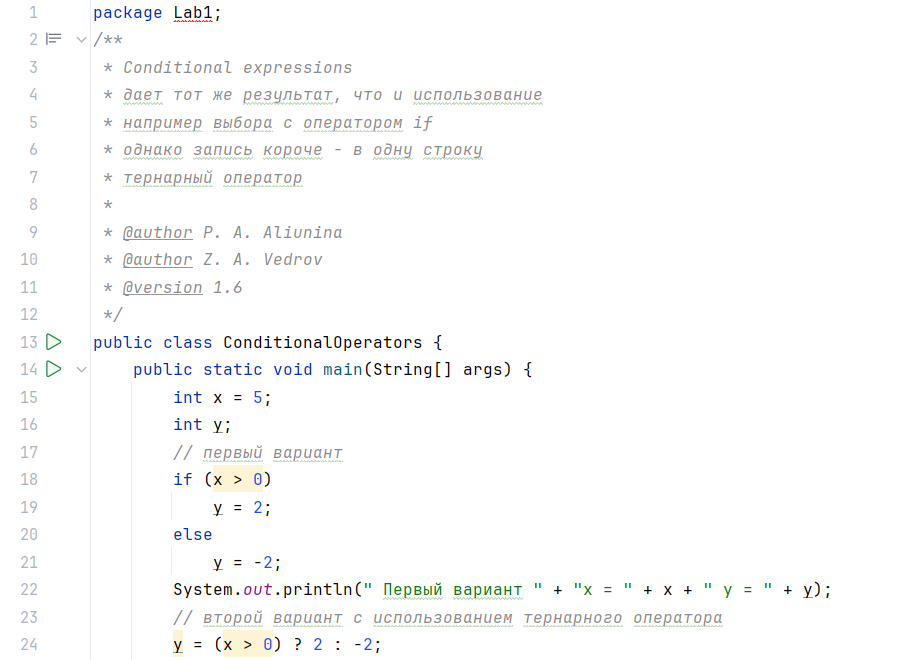
1. Conditional expressions

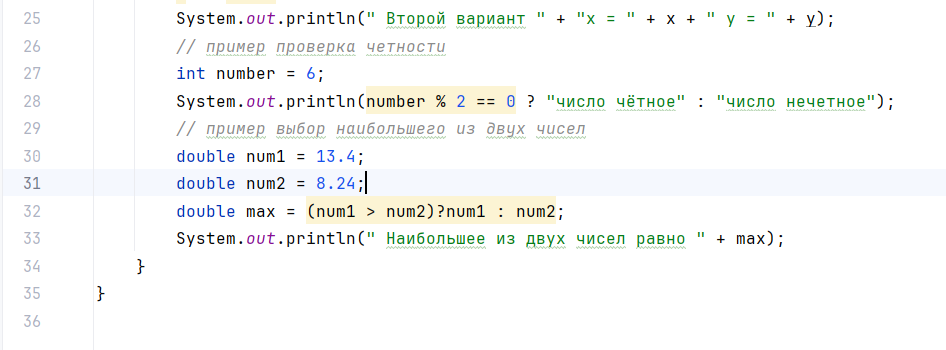
Java-тернарный оператор является единственным условным оператором, который принимает три операнда. Это условный оператор, который обеспечивает более короткий синтаксис выражения if..else. Они компилируются в эквивалентное выражение if-else, то есть они будут точно такими же.

* Условие: первая часть - это раздел условия.
* trueStatement: второй код, который выполняется в случае условия первой части, .
* falseStatement: блок кода третьей части выполняется, если условие имеет значение false.

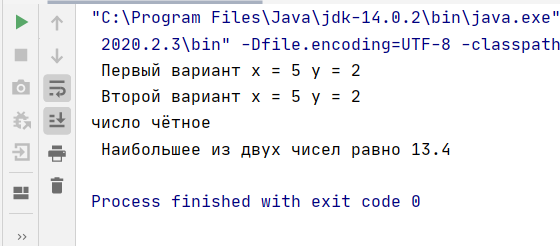
Тернарный оператор использует? и: символы. Первый операнд является булевым выражением; если выражение истинно, тогда возвращается значение второго операнда, иначе возвращается значение третьего операнда. Значение переменной часто зависит от того, является ли конкретное булево выражение истинным или нет.

1. **Код программы**

****

****

1. **Скриншот выполнения программы**



1. **Ответы на контрольные вопросы**
2. Что такое тернарный оператор?

Тернарный оператор (от латинского слова ternarius — тройной)- это оператор, состоящий из двух символов ? и :, который формирует условную тернарную операцию, возвращающую свой второй или третий операнд в зависимости от выполнения или невыполнения определённого условия, или другими словами от результата логического выражения.

**Задание №9 IncrementDecrement**

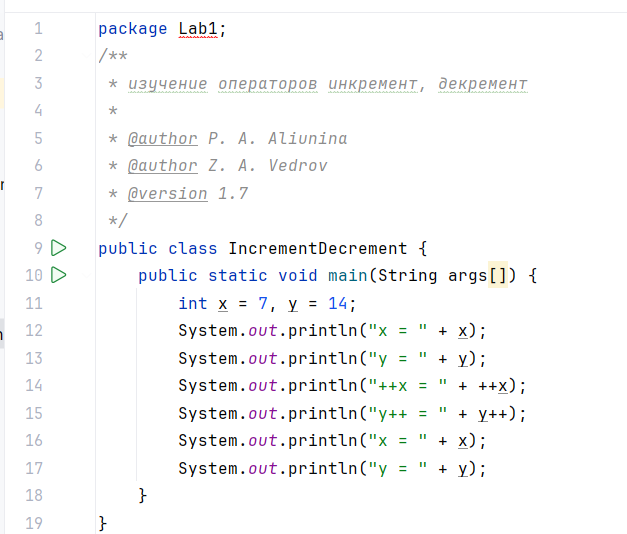
Изучите и законспектируйте теорию:

1. инкремент, декремент

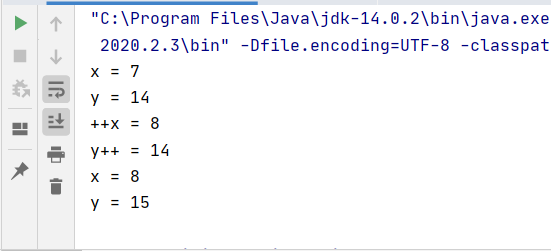
**Инкремент** — это операция во многих языках программирования, которая выполняет увеличение переменной. Чаще всего под инкрементом подразумевается увеличение переменной на 1 единицу. Обратной операцией называют декремент. **Декремент** — это уменьшение переменной. Чаще всего также на единицу. В Java инкремент и декремент — это в первую очередь унарные операторы, которые применяются к какой-либо числовой переменной. **Инкремент:** обозначается двумя знаками плюс "++" и увеличивает значение переменной на 1. **Декремент:** обозначается двумя знаками минус "--" и уменьшает значение переменной на 1.

**Prefix**, или Префиксная форма, описывается перед переменной "++x или --x", изначально икрементирует или декрементирует переменную. **Postfix**, или Постфиксная форма, описывается после переменной "x++ или x--", икрементирует или декрементирует переменную после вычисления.

1. **Код программы**

****

1. **Скриншот выполнения программы**



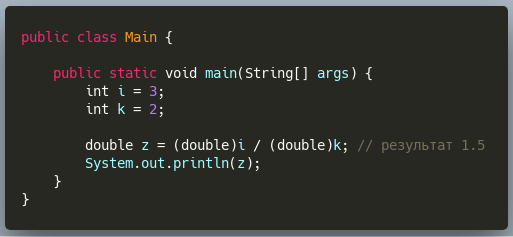
1. **Ответы на контрольные вопросы**
2. Каково отличие инкремент, декремент в арифметических врыжениях?

Инкремент — это операция во многих языках программирования, которая выполняет увеличение переменной. Чаще всего под инкрементом подразумевается увеличение переменной на 1 единицу. Обратной операцией называют декремент. Декремент — это уменьшение переменной. Чаще всего также на единицу. В Java инкремент и декремент — это в первую очередь унарные операторы, которые применяются к какой-либо числовой переменной. Инкремент: обозначается двумя знаками плюс "++" и увеличивает значение переменной на 1. Декремент: обозначается двумя знаками минус "--" и уменьшает значение переменной на 1.

**Задание № 10 CASTING**

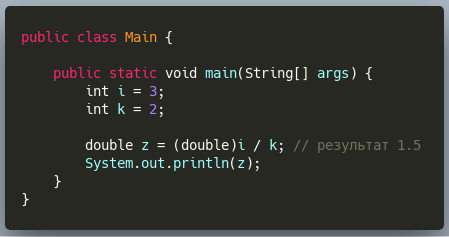
Изучите и законспектируйте теорию:

1. смешение типов переменных в арифметических выражениях



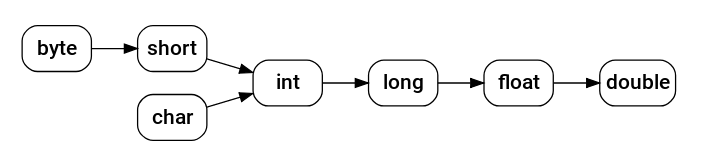
В этом примере операция ***(double)i*** — это приведение значения переменной iк типу double. И дальше у нас операция деления двух ***double*** значений (дробных), результат тоже ***double*** (***1.5***).

Это явное приведение типов, когда мы явно указываем в коде что к чему приводим, но так же есть неявное приведение. Допустим тот же результат 1.5мы получим если явно приведем только одну переменную в выражении:

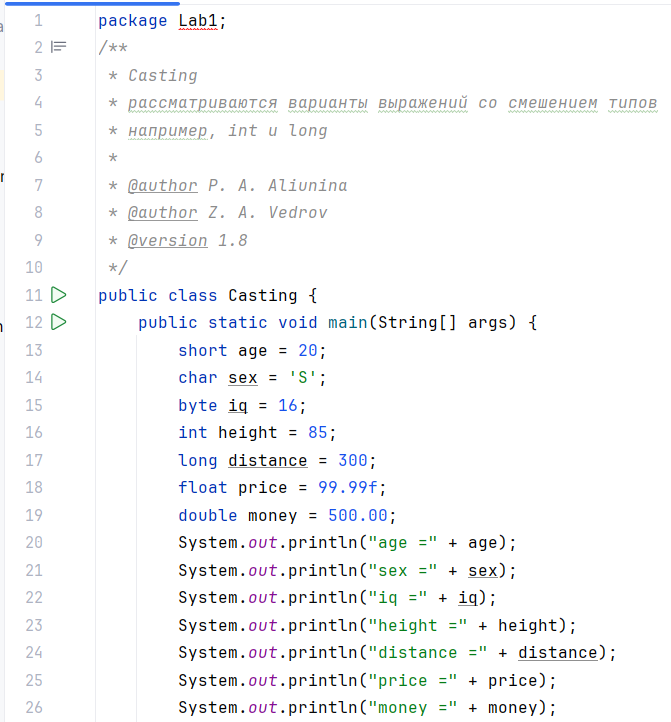


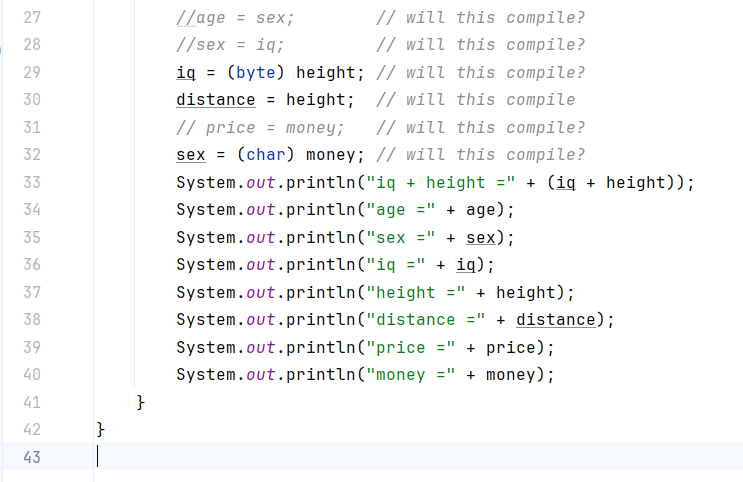
в выражении ***double z = (double) i / k*** переменная k неявно приведется к типу ***double***.

За неявным приведением стоит простое правило - меньшие по размеру типы приводятся к большим. Можно руководствоваться схемой ниже.

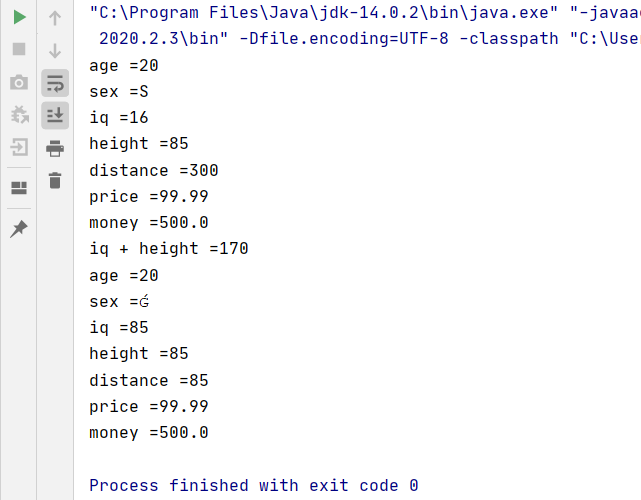


1. **Код программы**

****

****

1. **Скриншоты выполнения программы**



1. **Ответы на контрольные вопросы**

Каковы последствия смешения типов в арифметических выражениях?

1. Потеря точности: например, при делении целого числа на число с плавающей точкой, результат будет приведен к числу с плавающей точкой, и может потеряться дробная часть.
2. Ошибки компиляции: например, если вы попытаетесь сложить строку и число, компилятор выдаст ошибку, так как эти типы данных несовместимы для арифметических операций.
3. Приведение типов: Приведение типов может быть неявным (автоматическим) или явным (пользовательским). Неявное приведение типов может привести к потере данных или неожиданным результатам. Явное приведение типов позволяет вам явно указать, какой тип данных должен использоваться в выражении.

Глоссарий

**Метод main** – в Java является точкой входа в программу. В каждой Java-программе может быть только один метод `main`. Он имеет следующую сигнатуру: public static void main(String[] args).

**Операторы в языке Java** — это специальные символы, которые сообщают транслятору о том, что вы хотите выполнить операцию с некоторыми операндами. Некоторые операторы требуют одного операнда, их называют унарными.

**Тип** **данных** (**тип**) — множество значений и операций над этими значениями или класс данных, характеризуемый членами класса и операциями, которые могут быть к ним применены.

**RelationalOperators** - операторы сравнения (реляционные операторы) в Java используются для сравнения значений и определения их отношений. Эти операторы возвращают логическое значение (`true` или `false`) на основе результата сравнения.

**Conditional expressions** - условные выражения в Java используются для принятия решений на основе определенных условий. Эти выражения оценивают логическое условие и, в зависимости от результата, выполняют различные блоки кода. Самым распространенным условным выражением в Java является оператор `if`.

**Инкремент**, **инкрементирование** (от англ. increment «увеличение») — операция во многих языках программирования, увеличивающая переменную на единицу. Обратную операцию называют **декремент** (уменьшение).

**Класс `Scanner`** - в Java используется для чтения ввода с помощью различных методов. Он является частью пакета `java.util` и предоставляет удобные методы для считывания данных из различных источников, таких как консоль, файлы и строки.