Министерство образования Республики Беларусь

Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение информационных систем

и технологий»

**Отчёт**

по лабораторной работе №03

по дисциплине ***«Объектно-ориентированные технологии***

***программирования и стандарты проектирования»***

тема: «Примитивные типы данных Java и простые операторы (операции)»

|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | студент группы 10701119  Прадед Кирилл Витальевич |
| Преподаватель: |  | Иванченко Виктор Викторович |

2020-2021 учебный год

**ЛАБОРАТОРНАЯ РАБОТА #03**

Базовый синтаксис языка Java. Примитивные типы данных Java и простые операторы (операции)

**Цель работы:**

Ознакомиться с базовым синтаксисом и системой типов Java; изучить

примитивные типы данных и простые операторы (операции).

**Требования:**

1) При разработке кода можно использовать любую интегрированную среду разработки. Однако, при запуске программы рекомендуется вручную задействовать основные компоненты Java (компилятор – javac, утилиту для запуска JVM – java).

2) При разработке программ придерживайтесь соглашений по написанию кода на JAVA (Java Code-Convention).

**Задание**

Необходимо создать исследовательское приложение, которое тестирует все возможные (разрешённые) операции (арифметические, операции отношения (сравнения), логические, побитовые (бинарные), операции составного присваивания и другие операции) над разрешёнными соответствующими примитивными типами данных языка программирования Java (byte, short, int, long, float, double, char, boolean) с использованием статических методов класса.

*ПРИЛОЖЕНИЕ А*

Листинг исходного кода Lab03 **на языке программирования Java**

**package by.praded;**

**import java.util.InputMismatchException;**

**import java.util.Scanner;**

**Контрольные вопросы**

1. Опишите систему типов в Java.

В Java есть 8 примитивных типов, которые делят на 4 группы

1. Целые числа - byte, short, int, long
2. Числа с плавающей точкой (иначе вещественные) - float, double
3. Логический - boolean
4. Символьный – char

Также есть ссылочные типы данных. Их программист может создавать самостоятельно.

2. Каковы основные отличия примитивных типов данных от ссылочных?

Примитивные типы данных нельзя создавать самостоятельно.

Примитивные типы представляют собой одиночные значения. Ссылочные типы агрегируют типы, которые содержат нулевые или более примитивные значения или объекты. Примитивные типы требуют от одного до восьми байтов памяти.

Также отличается принцип хранение в зависимости от времени жизни переменной.(local(stack) global(heap))

3. Опишите базовые примитивные типы данных языка Java (byte, short, char, int, long, float, double, boolean) и их характеристики.

|  |  |  |
| --- | --- | --- |
| **Тип** | **Размер (бит)** | **Диапазон** |
| byte | 8 бит | от -128 до 127 |
| short | 16 бит | от -32768 до 32767 |
| char | 16 бит | беззнаковое целое число, представляющее собой символ UTF-16 (буквы и цифры) |
| int | 32 бит | от -2147483648 до 2147483647 |
| long | 64 бит | от -9223372036854775808L до 9223372036854775807L |

|  |  |  |
| --- | --- | --- |
| **Тип** | **Размер (бит)** | **Диапазон** |
| float | 32 | от 1.4e-45f до 3.4e+38f |
| double | 64 | от 4.9e-324 до 1.7e+308 |

4. Какие пять вещей характеризуют операции в языках программирования?

???

5. Какие существуют группы операций в Java и по какому признаку операции собираются в группы?

arithmetic operators

унарный -+, -+ \* / % ++ --

relational operators

== != < > <= >=

logical operators

&& || !

bitwise operator s

& | ^ ~ << >> >>>

assignment operators

= += -= \*= /= %= <<= >>= &= ^= |=

misc operators

переменная x = (выражение) ? значение if true : значение if false

операторы определяются в группы по взаимодействию с переменными

6. Опишите доступные операции в языке Java: − arithmetic operators (арифметические операции); − relational operators (операции отношения/сравнения); − logical operators (логические операции); − bitwise operators (побитовые/бинарные операции); − assignment operators (присваивание и операции составного присваивания); − misc operators (другие операции: instanceof – оператор проверки принадлежности типу, ? : – условный оператор (conditional operators), (type) – оператор приведения типа (type cast operator), оператор new и др.).

7. Для каких типов данных у языка Java и JMV нет встроенных операций?

Enum ?

8. Как в языке Java осуществляет вычисление выражения, если в нём находятся данные различных типов?

Зависит от типов данных, в каком то случае за кулисами происходит cast , а в каком то случае ошибка компияции

9. Какие операции и над какими типами данных могут выкинуть исключительную ситуацию во время выполнения программы?

Арифметические операции

10. Есть ли в Java перегрузка операторов (операций)? Какие операторы (операции) всё же перегружены на уровне языка Java и для каких типов данных?

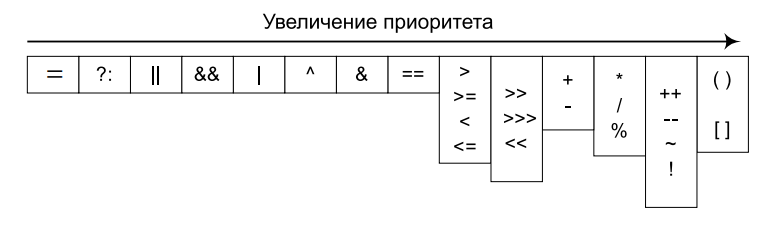
Арифметические операторы перегружены, унарный плюс унарный минус

11. В чём существенная разница между операциями || (логическое ИЛИ) и | (побитовое ИЛИ), а также && (логическое И) и & (побитовое И)?

Логические операнды && и || - это укороченные версии операторов & и |. Отличие заключается в том, что например в укороченной версии (или), если первый операнд имеет значение (true), то выражение сразу становится (true) и второй операнд даже не рассматривается

12. Что такое приоритет выполнения операций?

Порядок их выполнения



13. Какие операции имеют наибольший приоритет в языке Java?

14. Какие операции имеют наименьший приоритет в языке Java?

15. Как распределён приоритет между префиксным и постфиксным инкрементом (декрементом)?

16. Как распределён приоритет между операциями логической группы?

17. Что такое ассоциативность выполнения операций?

18. Какие операции выполняются слева направо?

 ?:, ||, &&, |, ^, &, ==, !=, <, >, <=, >=, instanceof, <<, >>, >>>, +, —, \*, /, %,

19. Какие операции выполняются справа налево?

пред++, —, +, -, ~, !,пост ++, —  =, +=, -=,\*=, /=, %=, &=, ^=,  |=, <<=, >>=, >>>=

20. Контролируется ли в Java переполнение (выход значения за границы выделенной памяти) при вычислении выражений?

При инициализации литералами – да, в рантайме или при выполнении операций – нет.

21. Приведите схему и опишите, каким образом работает сужающее ( ≈ понижающее ≈ явное) и расширяющее (≈ повышающее ≈ разрешённое ≈ неявное ≈ автоматическое) приведение примитивных типов.

Может происходить автоматически:

1. byte и short. byte имеет меньший диапазон допустимых значений, чем short. То есть byte это как бы коробочка поменьше, а short - это коробочка побольше. И значит, мы можем byte вложить в short.

2. byte и int. byte имеет меньший диапазон допустимых значений, чем int. То есть byte это как бы коробочка поменьше, а int - это коробочка побольше. И значит, мы можем byte вложить в int.

3. int и long. int имеет меньший диапазон допустимых значений, чем long. То есть int это как бы коробочка поменьше, а long - это коробочка побольше. И значит, мы можем int вложить в long.

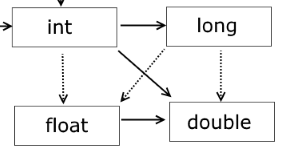
Понижение типа нужно проводить явно

int a=0;

long b=15;

a = (int) b;

22. Опишите особенности потери точности при явном и неявном приведении типов.

пунктирными стрелками показаны приведения типов при которых теряется точность.

23. При работе с какими операциями не требуется явное приведение типов?

24. Зачем в объектно-ориентированном языке Java поддерживаются примитивные типы данных?

Потому что на основе этих данных составлены все ссылочные. Они являются строительными блоками языка Java

25. Зачем в Java существуют Wrapper-классы (классы-обвёртки, каждый из которых соответствует примитивному типу данных)?

Для коллекций.

Например, обычный int занимает меньше места, и если нет необходимости проводить над ним особые операции, Ваш компьютер будет работать быстрее.

В свою очередь, с помощью класса-оболочки Integer можно выполнять специальные операции - например, перевести текст в число (с помощью метода .parseInt() для Integer-а ). Если попробовать сделать это с помощью кода самому придется изрядно повозиться.