Лекция Программирование на Java: переменные, операторы, работа с отладчиком.

Горячие клавиши Идея:

Reformate code (ctrl+alt+L)

New class (alt +insert)

Создать конфигурацию запуска (Ctrl + Shift +F10)

Поставить точку остановки (Ctrl + F8)

Литералы – Это непосредственной значение, записанное в коде.

Например, в выражении: float amount = 1000.50, 1000.50 – это литерал.

В Java к численным литералам применяются два ключевых правила:

1 если литерал представляет из себя целое число, то он имеет тип int

2 если литерал представляет из себя вещественное число (число с точкой), то он имеет тип double

Получается, что 1000.50 – это double.

Для того чтобы закастить тип данных, нужно :

1. Для float указывать после литерала суффикс f или F.
2. Для Long нужно указывать после литерала суффикс lили L.
3. Для byte и short ничего указывать не нужно, Java сама разберется, если литерал не умещается в границы типа.

Узкие места примитивных типов.

Это связанно с особенностями хранения вещественных чисел в памяти компьютера: не все числа можно аккуратно и точно представить в памяти компьютера.

У целых чисел есть свои особенности.

JSHELL console – позволяет выполнять выполнять кусочки кода, без необходимости создавать полноценный проект.

Если нужны точные вычисления, то следует отказаться от вещественных чисел.

Есть у вещественных чисел еще одна особенность:

Float before = 999\_999\_999\_999F;

Float after = before – 1000F;

RUN^

Field float before = 1.0E12

Field float after = 1.0E12

Before == after = true

1.0E12 – специальная запись (scientific nоtation), которая означает 1.0\*1012

А оператор == проверяет на равенство два числа :true – равны false – нет равны

Приведение типов

Если в выражении встречаются разные типы данных, то используются четкие правила для выведения типа всего выражения(т.к. результатом выражения может быть только одно значение):

1. Если один из операндов double, то все выражение double.
2. Если один из операндов float, то все выражение float.
3. Если один из операндов long, то все выражение long.
4. Если один из операндов int, то все выражение int.

Таким образом long + float ,будет Float

Для того чтобы преобразовать float в int есть специальный оператор cast.

Int result = (int)99.99F

Условия

В зависимости от того, зарегистрирован пользователь или нет, логика нашей программы должна изменяться.

В Java для этого есть конструкция if.

Существует она в двух формах:

If (expression) {

//код внутри фигурных скобок выполняется только если expression = true

}

If (expression) {

//код внутри фигурных скобок выполняется только если expression = true

} else {

//код внутри фигурных скобок выполняется только если expression = false

Операторы сравнения

Теперь когда мы знаем if, осталось добавить лимит. В этом помогут операторы сравнения:

== - проверка на равенство(не использовать с вещественными числами)

!= - проверка на неравенство;

< - меньше

<= - меньше или равно

>>

> - больше

>= - больше или равно

Операторы сравнения всегда возвращают Boolean.

Тернарный оператор

Используется для сокращения кода

expression ? if-true-value : if-false-value

**Автотесты**

Автотесты запускаются одной командой

Проверяют функциональность приложения без нашего участия

Сообщают нам о результатах проверок (pass/fail)

Тестирование

В самом упрощенном сценарии тестирование нашей утилиты сводится к следующему:

1. Запустить приложение с нужными входными данными (registered и amount)
2. Сравнить фактический результат
3. Сравнить фактический результат с ожидаемым (совпадает pass, не совпадает fail)

Testability

Функции: мы могли бы из нашего кода сделать функцию «рассчитать бонус», которая на вход принимает два параметра:

1 Зарегистрирован пользователь или нет

2 Сумму покупок

А возвращать функция будет сумму бонуса.

Объекты в Java функции не могут существовать сами по себе.

В реальном мире вы тоже не можете просто позвонить, вам нужен телефон, который умеет звонить.

Свойства и методы:

У объектов есть свойства и функции (в Java они называются методами)

Свойства – это уровень заряда, баланс и тд

Методы – это позвонить, заблокировать, проиграть музыку

В Java функции называются методами.

Представим, что мы программируем систему, управляющую лифтом.

Система должна уметь:

1 Давать возможность вызывать лифт на определенный этаж это метод

2 Давать возможность перемещать лифт на определенный этаж(когда вы внутри лифта) это тоже метод.

Пример из жизни:

При этом у системы есть свойства:

1 Исправность – исправна или нет

2 Максимально допустимая для перевозки масса

3 Перевозима я масса (масса тех, кто находится в лифте)

4 Этаж (на котором находится лифт)

5 Статус лифта – свободен лифт в данный момент или перемещается

Состояние

Фактически сделали то же самое, что и при проектировании тестов;

1 выполнили анализ ключевых возможностей (что система умеет делать – методы)

2 определили ключевые характеристики (какими свойствами/1параметрами система обладает – свойства)

3 определили возможные состояния системы (конкретные значения свойств – состояние)

Для того чтобы получить объект, с которым можно работать, нам нужно сделать два шага:

1 Описать этот объект (какие свойства и методы у него будут)

2 создать объект из этого описания

Класс

Описание объекта мы будем называть классом.

Класс создается так же, как Main название вводите как BonusService