**Домашнее задание № 1. Диалоговый калькулятор.**

**Создать консольное Java-приложение, реализующее диалог калькулятора, выполняющего указываемые операции с двумя операндами х и y.**

**Приложение запускается и выводит в консоль приглашение пользователю. Получив приглашение, пользователь может вводить через консоль команды диалога (ввод команды завершается нажатием клавиши <Enter>, после чего команда выполняется), как указано ниже:**

**1. Команда выхода из приложения (exit – case insensitive). Пользователю печатается текущее время в формате ЧЧ:ММ:СС и приложение завершается. Выполнение всех других команд завершаются приглашением ввода следующей команды.**

**2. Команда задания значения операндам вида <имя\_операнда> = <значение>. Здесь <имя\_операнда> задает один из двух операндов: x или y (калькулятор может выполнять операции только с этими двумя операндами). Пробелы вокруг знака ‘=’ не учитываются. <значение> операнда задается десятичными цифрами, между которыми может быть одна точка (‘.’). Если точка есть, число понимается как значение типа float с целой и дробной частью слева и справа от точки, соответственно. Если точки нет, число задает значение типа int. Выполняется проверка возможности представления вводимого числа значением соответствующего типа (float или int) по правилам, принятым в языке Java, и с выдачей в консоль диагностики ошибки при необходимости. Например:**

**x = 7 // операнд х получает целочисленное значение 7;**

**y = 1.5 // операнд y получает значение 1.5F**

**Если какому-либо из операндов (x или y) не присвоено значение или еще не задана операция, никакая операция не выполняется. Если калькулятору задана операция (см. п.3 ниже) и оба операнда имеют заданные значения, в консоль выводится сообщение вида:**

**x <знак\_операции> y = <результат\_операции>,**

**где <результат\_операции> представляется в виде, соответствующем правилам языка Java о приведении типов (например, 7 / 5 = 1, так как, т.к. деление целых чисел выдает целую часть частного, но 7.0 / 5 = 1.40). При выводе знаки операции и равенства (‘=’) окружаются знаком пробела слева и справа. При выполнении операций с плавающей точкой, результат выводить с точностью до сотых.**

**3. Команда задания операции определяет действие калькулятора над операндами и кодируется знаком операции: ‘+’ - сложение, ‘-‘ - вычитание, ‘\*’ - умножение, ‘/’ - деление. Если операнды имеют заданные значения, то команда операции приводит к печати в консоль результата соотвествующей операции, как указано в п.2 выше. Если какой-либо из операндов не задан, вычисление указанной операции не производится. Таким образом, при наличии заданных операндов и операции, любое их изменение приводит к соответствующему вычислению и выдаче результата – как реакции на эти изменения; иначе – вычисления не производятся. Если запрошена неподдерживаемая операция – напечатать в консоль диагностику ошибки.**

**Оценивается полнота и качество реализации приложения:**

* **осознанное использование уместных приемов программирования;**
* **качество кодирования и документирования кода;**
* **отсутствие ошибок и наличие JUnit-тестов.**

**Результатом выполнения являются два архива:**

* **архив (.rar) с Idea-проектом,**
* **исполняемый jar-архив, который собирается как артефакт этого проекта (production-код без тестов) и который можно выполнить из командной строки:**

**java –jar Calculator.jar**

**Ниже приведены примеры работы калькулятора. Знак ‘>’ используется для обозначения приглашение системы к вводу пользователя.**

**> java -jar Calculator.jar**

**> x = 5**

**> y = 3**

**> +**

5 + 3 = 8

**> x = 3**

3 + 3 = 6

**> exit**

14:43:14

**> java -jar Calculator.jar**

**> x = 5**

**> /**

**> y = 2.0**

5 / 2.0 = 2.50

**> \***

5 \* 2.0 = 10.00

**> exit**

16:17:32