Занятие 2 07.02.2024

Задача 1

Создайте класс **Rectangle**, который моделирует область рабочего пространства на плоскости. Снабдите его двумя конструкторами. Первый конструктор принимает 4 параметра - координаты верхнего левого угла, в системе координат, где ось X направлена слева направо и ось Y направлена сверху вниз, и размеры прямоугольника - ширину и высоту. Все параметры - целые неотрицательные числа не превосходящие $2^{63} - 1$. Второй конструктор принимает только 2 параметра - ширину и высоту. Координаты левого верхнего угла следует принять за $(0, 0)$. Используйте делегирование при конструировании. Так как координаты могут быть переданы в том числе и отрицательные, сделайте в конструкторе проверку, что координты не отрицательны, а ширина и высота положительны. Если проверка провалена - выбросьте исключение **IllegalArgumentException** с разумным сообщением об ошибке.

Добавьте в класс открытый метод **overlap**:

```
1 public Optional<Rectangle> overlap(Rectangle r);
```

Метод **overlap** возвращает новый объект Rectangle - прямоугольник, по которому пересекаются два прямоугольника: тот, на котором вызван метод, и переданный в качестве аргумента. Если прямоугольники не пересекаются, то возвращается пустой Otional. Переопределите методы: **equals** и **hasCode**. Переопределите метод **toString**, чтобы он возвращал строку в формате $(x, y) : (w, h)$, где x и y - координаты левого верхнего угла, w и h - ширина и высота.

Задача 2

Реализуйте набор классов для валидации строк (проверки строк на соответствие некоторым правилам). Корнем будет интерфейс **ValidateString**:

```
1 public interface ValidateString {
2     boolean isValid(String s);
3 }
```

Добавьте в систему классы для комбинирования проверок: ValidateOr - применяет два валидатора, если хотя бы один прошёл, то строка верна, ValidateAnd - применяет два валидатора, если хотя бы один не про-

шёл, то строка не верна, ValidateInverse - обращает уже написанный валидатор. Дополните систему классом ValidatePalindrom, который проверяет, что строка является палидромом. Дополните систему классом ValidateNoSpace, который проверяет, что строка не содержит пробелов. Все классы должны быть открытыми и располагаться в отдельных файлах. **Указание**: используйте шаблон проектирования Composite на основе Decorator.

Задача 3

Реализуйте программу поточного вычисления на коммандах. Пользователь может выполнять 4 операции над целым числом:

1. add <number> - складывает текущее число с числом <number> и результат становится текущим числом;
2. sub <number> - вычитает из текущего числа число <number> и результат становится текущим числом;
3. mul <number> - умножает текущее число на <number> и результат становится текущим числом;
4. div <number> - делит текущее число на <number> и результат становится текущим числом, если <number> равен нулю, то пользователю выдаётся сообщение об ошибке, вычисление откатывается.

Так же у пользователя есть три важные команды:

1. undo - вернуться на одну операцию назад к предыдущему числу;
2. redo - выполнить отменённую операцию повторно;
3. exit - завершить программу.

Если пользователь вводит неверную команду или не число в том месте, где нужно число - выведите сообщение об ошибке и откатите вычисления. Программа продолжает работать. Начальное число пользователя равно нулю. Количество возможных откатов не превышает 10. Если пользователь пытается откатиться дальше, чем существует операций, которые запомнены, то выдаётся сообщение об ошибке. Если пользователь

вызывает операции redo, но все операции уже применены, то выдаётся сообщение об ошибке. Если пользователь откатывается на несколько операций, а затем вызывает новую операцию, то операция, которая бы выполнилась при следующем redo затирается, но последующие операции остаются. **Указание**: используйте шаблон Command.

0
> add 10
10
> mul -5
-50
> div 2
-25
> sub -10
-15
> div 0
Деление на 0!
-15
> undo
-25
> undo
-50
> add 60
10
> redo
0
> redo
Нет операций в стеке
> doun
Ошибочная операция
> add rut
После операции следует ввести целое число
> add 3.4
После операции следует ввести целое число
> exit

Указание: если обработака ошибок ввода с помощью java.util.Scanner вызывает у вас затруднения, то подготовьте решение в предположении, что пользователь не совершает ошибок ввода.