

Практическая работа “Java: исключительные ситуации”

Продолжая работу над проектом коллекций задач необходимо учесть исключительные ситуации в работе системы.

Обязательное задание

Проанализировать методы в классах указанных ниже на предмет исключительных ситуаций. Попытаться понять, какие могут возникнуть ситуации, которые могут негативно повлиять на работу программы. Обработать каждую исключительную ситуацию путем выбрасывания исключения нужного типа. Пример выбрасывания исключений рассмотрен ниже.

```
com.nc.edu.ta.yourname.prN.Task  
com.nc.edu.ta.yourname.prN.ArrayTaskList  
com.nc.edu.ta.yourname.prN.LinkedList
```

Обратите внимание, что:

1. В предыдущих заданиях время задавалось целым числом, как количество единиц времени, прошедших с начала отсчета. Согласно этому отметки времени не могут быть отрицательными.
2. Список задач должен содержать задачи, поэтому в него нельзя добавлять пустые ссылки (null).
3. Интервал повторения задачи должен быть больше нуля. Конструктор класса Task обязательно должен генерировать включение `IllegalArgumentException` в случае, когда время было передано отрицательным числом, а метод `getTask` должен генерировать `IndexOutOfBoundsException` в случае, когда индекс выходит за допустимые пределы списка.

Во всех остальных случаях решение о выбрасывании исключения и его типе остается на Ваш выбор.

Пример

Предположим, что существует класс, который умеет вычислять деление одного числа на другое:

```
class TestMath {  
    public static double divide(double dividend, double divisor) {  
        return dividend / divisor;  
    }  
}
```

Может возникнуть ситуация, когда в качестве второго параметра передается 0:

```
TestMath.divide(5, 0);
```

В таком случае нужно указать, что может возникнуть ситуация, в которой по правилам математики невозможно выполнить операцию деления. Для этого используем тип исключений, который будет указывать на арифметическую ошибку, которую программист должен будет обработать в соответствии с требованиями его программы в вызываемом месте. Для этого укажем в объявлении метода исключение:

```
public static double divide(double dividend, double divisor)
throws ArithmeticException{
    if(divisor==0) throw new ArithmeticException();

    return dividend/divisor;
}
```

Тестирование

Необходимо добавить в проект тесты, которые находятся в архиве `unit.zip` (будут доступны на занятии), а так же использовать тесты из предыдущей практики.