Sample:

public class QuizProperty{

private String \_type;

private int \_level;

private String \_operator;

private String \_nameOfOperator;

private String \_title;

private String \_description;

private List \_problem;

public QuizProperty(String type,

int level,

String operator,

String nameOfOperator,

String title,

String description,

List problem){

\_type = type;

\_level = level;

\_operator = operator;

\_nameOfOperator = nameOfOperator;

\_title = title;

\_description = description;

\_problem = problem;

}

public String getType(){return \_type;}

public int getLevel(){return \_level;}

public String getTitle(){return \_title;}

public String getOperator(){return \_operator;}

public String getDescription(){return \_description;}

public List getAllProblems(){return \_problem;}

public String getOperatorName(){return \_nameOfOperator;}

public Quiz makeQuiz(int count) throws IllegalArgumentException{ if (count > \_problem.size())

throw new IllegalArgumentException("The number of problems in quiz must be less than "

+\_problem.size());

Random rnd = new Random(); Vector quizChosen = new Vector(); Hashtable quizLooked = new Hashtable(); BasicProblemParser parser = new BasicProblemParser(); try{

while (quizChosen.size() < count){

int i = rnd.nextInt(\_problem.size());

String prob = (String)\_problem.get(i);

if (!quizLooked.containsKey(prob)){

quizLooked.put(prob,prob);

parser.parse(prob);

quizChosen.add( new Problem( prob,parser.calculate()));

}

}

}catch(IllegalArgumentException iae){

throw new IllegalArgumentException("The quiz type "+\_type+" has an illegal format. & "+iae);

}

return new Quiz(this.getType(), quizChosen);

}

}

**Motivation to refactor**: the class name did not represent its entire responsibility. This problem could be solved by splitting this class into two pieces. By using the factory method pattern and then applying the Extract Class refactoring technique, makeQuiz() method for the creation of Quiz instance is going to be taken out of QuizProperty and moved to QuizFactory class where it should belong. The code shows QuizProperty class before any refactoring was started.