模版·中缀表达式转后缀

```
public class InfixToPostfix{

    private static final Map<Character,Integer> precedence=new HashMap<>();

    static{
        precedence.put('+',2);
        precedence.put('-',2);
        precedence.put('*',4);
        precedence.put('/',4);
    }

    public static String infixToPostfix(String infixExpression){
        Stack<Character> stack=new Stack<>();
        StringBuilder postfixExpression=new StringBuilder();

        for(char c : infixExpression.toCharArray()){
            if(Character.isDigit(c)){
                postfixExpression.append(c);
            }else{
                switch(c){
                    case'(':
                        stack.push('(');
                        break;
                    case')':
                        while(!stack.isEmpty()&&stack.peek()!='('){
                            postfixExpression.append(stack.pop());
                        }
                        if(stack.isEmpty()){
                            throw new IllegalArgumentException("Invalidinfixexpression");
                        }
                        stack.pop();
                        break;
                    case'+':
                    case'-':
                    case'*':
                    case'/':
                        while(!stack.isEmpty()&&precedence.get(stack.peek())>=precedence.get(c)){
                            postfixExpression.append(stack.pop());
                        }
```

```
                    stack.push(c);
                    break;
                }
            }
        }

        while(!stack.isEmpty()){
            postfixExpression.append(stack.pop());
        }

        return postfixExpression.toString();
    }
}
```