

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

по лабораторной работе №2

по дисциплине: «Качество и тестирование программного обеспечения»

Выполнил: Евдокимов Н.А.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверили: Конюхова О.В., Ужаринский А.Ю.

Отметка о зачете:

Дата: «\_\_\_» \_\_\_\_\_ 2020 г.

Орёл, 2020

**Тема:** Структурное тестирование программного обеспечения.

**Задание:**

1.1 Необходимо протестировать требования, предъявляемые заказчиком. Требования берём из документа lab1V2.pdf. Для каждого требования необходимо написать уточняющие вопросы если требование сформулировано не корректно или не однозначно.

1.2 Провести структурное тестирование алгоритма по методу базового пути. Берём алгоритм из файла Lab32017.pdf пишем алгоритм для решения задачи. Для каждой функции строим потоковый граф. Ищем все возможные пути в графе и пишем для каждого пути свой тест.

**Вариант 4:**

1.1 Спецификация на разработку микроволновой печи.

1.2 Функция поиска пути в неориентированном графе методом A\*. На вход подается карта (граф с географическими координатами вершин) и два угла. На выходе – путь между этими узлами.

## 1.1

*Микроволновая печь должна иметь дверцу:*

Каким образом и в какую сторону должна открываться и закрываться дверца (ручка, кнопка и т. д.)?

*Микроволновая печь должна позволять помещать вместе с продуктами металлические столовые приборы и посуду, при этом они не должны нагреваться или искрить:*

Имеется ли в виду выключенное состояние печи? Если нет, то следует спросить у заказчика, знает ли он что такое микроволновая печь. Возможно он хочет от нас другой вид печи, но не знает, как она называется.

*Микроволновая печь должна иметь подсветку:*

Следует спросить подсветка какой части печи требуется: кнопок, цифрового табло (спросить о необходимости их наличия), отсека для пищи, декоративная подсветка. Также уточнить какого цвета подсветка должна быть.

*Микроволновка не должна сушить продукты, которые в ней готовятся:*

В связи с тем, что при нагреве ускоряется испарение воды, абсолютно формально соблюсти требование не представляется возможным. Следует спросить имеется ли ввиду ограничение мощности печи или времени приготовления продуктов, чтобы они не доходили до состояния сублиматов.

## 1.2

### Исходный код:

```
class AStar {
    companion object {
        private fun calculateDistance(point1: GraphPoint, point2: GraphPoint):
Double {
            return sqrt(
                (point1.x - point2.x).toDouble().pow(2)
                + (point1.y - point2.y).toDouble().pow(2)
            )
        }

        fun findPath(pointFrom: GraphPoint, pointTo: GraphPoint): List<GraphPoint>
{
            val path = mutableListOf<GraphPoint>()
            val visitedPoints = mutableSetOf<GraphPoint>()
            val heuristic = { point: GraphPoint ->
                calculateDistance(pointFrom, point) + calculateDistance(point,
pointTo)
            }

            var currentPoint = pointFrom
            while(currentPoint != pointTo) {
                visitedPoints.add(currentPoint)
                path.add(currentPoint)
            }
        }
    }
}
```

```

        currentPoint = currentPoint.neighbours.run {
            // Best point calculated using heuristic
            val nextPoint = filter {
                it !in visitedPoints
            }.minBy(heuristic) ?: throw IllegalStateException("Can't find a
way. There's no not visited points")
            foreach { visitedPoints.add(it) }

            nextPoint
        }
    }
    path.add(currentPoint)

    return path
}
}

```

**Потоковый граф:**

```

findPath
1) path = emptyList()
   pointToIsVisited = emptyMap()
   heuristic = (point) -> F(point) + G(point)
   currentPoint = pointFrom

   while(currentPoint != pointTo)

2) visitedPoints.add(currentPoint)
   path.add(currentPoint)

3) currentPoint.neighbours.filter(it !in pointsToIsVisited).minBy(heuristic)

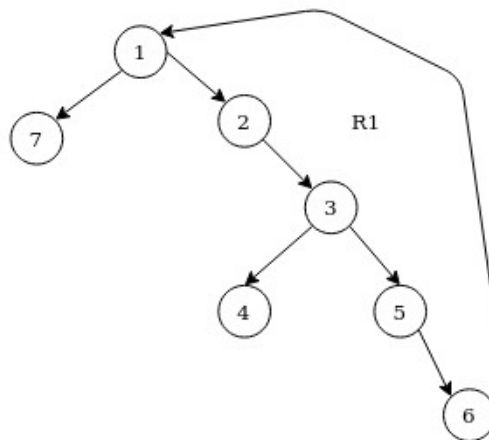
4) exception // (3) == null

5) currentPoint.neighbours.forEach { visitedPoints.add(it) }
   currentPoint = (3)

6) end while

7) path.add(currentPoint)
   end findPath

```



Пути:

1-7

1-2-3-4

1-2-3-5-6-1-7

**1-7:**

*Название теста:*

equalPointsTest

*Описание сценария:*

pointFrom == pointTo

*Входные параметры:*

val p1 = GraphPoint(0, 0, "A")

val p2 = GraphPoint(0, 0, "A")

*Результат*

path: List<GraphPoint>

path.size == 1

path[0] == p1 && path[0] == p2

**1-2-3-4:**

*Название теста:*

thereIsNoPathTest

*Описание сценария:*

currentPoint.neighboiurs

.filter { it !in visitedPoints }

.minBy(heuristic) == null

*Входные параметры:*

val p1 = GraphPoint(0, 0, "A")

val p2 = GraphPoint(0, 1, "B")

*Результат*

IllegalStateException("Can't find a way. There's no not visited points")

**1-2-3-5-6-1-7:**

*Название теста:*

thereIsAPathTest

*Описание сценария:*

pointFrom == pointTo

currentPoint.neighboiurs

.filter { it !in visitedPoints }

.minBy(heuritic) != null

*Входные параметры:*

val p1 = GraphPoint(0, 0, "A", mutableListOf(GraphPoint(0, 1, "B")))

val p2 = p1.neighbours[0]

*Результат*

path.size == 2

path.first() == p1

path.last() == p2