Лабораторная работа №2

СТРУКТУРНОЕ ПРОГРАММИРОВАНИЕ

Цель: познакомиться с особенностями структурного программирования. Решить задания в структурном стиле. Составить отчет.

Теоретические сведения

Структурное программирование – методология разработки программного обеспечения, в основе которой лежит представление программы в виде иерархической структуры блоков.

В соответствии с данной методологией любая программа строится без использования оператора goto из трех базовых управляющих структур: последовательность, ветвление, цикл; кроме того, используются подпрограммы. При этом разработка программы ведется пошагово, методом «сверху вниз».

Цель структурного программирования – повысить производительность труда программистов, в том числе при разработке больших и сложных программных комплексов, сократить число ошибок, упростить отладку, модификацию и сопровождение программного обеспечения.

Теорема Бёма – Якопини – положение структурного программирования, согласно которому любой исполняемый алгоритм может быть преобразован к структурированному виду, то есть такому виду, когда ход его выполнения определяется только при помощи трех структур управления: последовательной, ветвлений и повторов или циклов.

Цикл – разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций.

Бесконечный цикл – цикл, написанный таким образом, что условие выхода из него никогда не выполняется.

```
repeat {
  # Выполняем действие
}
```

Рисунок 1 – Бесконечный цикл в R

```
while(TRUE) {
  # Выполняем действие
}
```

Рисунок 2 – Бесконечный цикл в R (на основе цикла while)

Цикл с предусловием – цикл, который выполняется, пока истинно некоторое условие, указанное перед его началом. Это условие проверяется **до** выполнения тела цикла, поэтому тело может быть не выполнено ни разу (если условие с самого начала ложно). В большинстве процедурных языков программирования реализуется оператором **while**, отсюда его второе название – while-цикл.

```
i <- 0
while(i <- 5) {
  i <- i + 1
  print(i)
}
```

Рисунок 3 – Цикл с предусловием в R (while)

Цикл со счетчиком – цикл, в котором некоторая переменная изменяет свое значение от заданного начального значения до конечного значения с некоторым шагом, и для каждого значения этой переменной тело цикла выполняется один раз. В большинстве языков программирования реализуется оператором for, в котором указывается счетчик (так

называемая «переменная цикла»), требуемое количество проходов (или граничное значение счетчика) и, возможно, шаг, с которым изменяется счетчик

```
for (i in 1:5) {
  print(i)
}
```

Рисунок 4 – Цикл со счетчиком в R (for)

Во многих языках программирования вместо стандартного счетчика, имеющего индексное значение, можно пройтись по итерируемой последовательности.

```
cities <- c("New York",
            "Paris",
            "Saint-Denis",
            "Oslo",
            "Montreal",
            "Helsinki")
for (city in cities) {
  print(city)
}
```

Рисунок 5 – Цикл со счетчиком в R (for), перебирающий итерируемый вектор

Досрочный выход из цикла. Команда досрочного выхода применяется, когда необходимо прервать выполнение цикла, в котором условие выхода еще не достигнуто. Такое бывает, например, когда при выполнении тела цикла обнаруживается ошибка, после которой дальнейшая работа цикла не имеет смысла.

Команда досрочного выхода обычно называется break или exit, а ее действие аналогично действию команды безусловного перехода (goto) на команду, непосредственно следующую за циклом, внутри которого эта команда находится.

```
for(i in 1:5) {
  if (i == 3) {
    break
    }
  print(i)
}
```

Рисунок 6 – Досрочный выход из цикла for в R

Пропуск итерации. Данный оператор применяется, когда в текущей итерации цикла необходимо пропустить все команды до конца тела цикла. При этом сам цикл прерываться не должен, условия продолжения или выхода должны вычисляться обычным образом.

В языке C, его языках-потомках и во многих других языках программирования в качестве команды пропуска итерации используется оператор continue в конструкции цикла. Действие этого оператора аналогично безусловному переходу на строку внутри тела цикла, следующую за последней его командой. В языке R специальным оператор для пропуска итерации является next.

```
for(i in 1:5) {
  if (i == 3) {
    next
    }
  print(i)
}
```

Рисунок 7 – Пропуск итерации в цикле for в R

Практическая часть

Задание 1 – Написать программу, выполненную в структурном стиле. Программа должна рассчитывать площадь фигур (программа должна корректно отрабатывать данные

согласно варианту в приложении А). На вход программа запрашивает строку, если в нее введено название фигуры, то программа запрашивает необходимые параметры фигуры, если введено значение отличное от названия фигуры, то программа повторно предлагает ввести название фигуры, если пользователь не справляется с этой задачей более 3 раз подряд, то программа сообщает о некорректности действий пользователя и завершается. В случае введения корректных данных программа должна выдать ответ, а также описание хода решения.

Программа должна быть выполнена в виде блок-схемы и на ЯВУ.

Задание 2 – Написать программу вычисляющую площадь неправильного многоугольника. Многоугольник на плоскости задается целочисленными координатами своих N вершин в декартовой системе. Стороны многоугольника не соприкасаются (за исключением соседних - в вершинах) и не пересекаются. Программа в первой строке должна принимать число N – количество вершин многоугольника, в последующих N строках – координаты соответствующих вершин (вершины задаются в последовательности против часовой стрелки). На выход программа должна выдавать площадь фигуры.

Программа должна быть выполнена в виде блок-схемы и на ЯВУ.

Вопросы для контроля из материалов лабораторного занятия

1. Особенности структурного программирования
2. Теорема Бёма – Якопини
3. Пропуск итерации и досрочный выход из цикла

Вопросы для поиска и письменного ответа

1. Цикл с постусловием
2. Совместный цикл
3. Вложенные циклы
4. Принцип проектирования программ «сверху-вниз».

ПРИЛОЖЕНИЕ А – ВАРИАНТЫ К ЗАДАНИЮ 1

Вариант	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Треугольник														
Квадрат														
Прямоугольник											+	+	+	+
Параллелограмм					+	+	+	+	+	+				
Ромб		+	+	+				+	+	+				+
Трапеция	+		+	+		+	+			+		+	+	
Круг	+	+		+	+		+		+		+		+	
Эллипс	+	+	+		+	+		+			+	+		+
Вариант	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**
Треугольник														
Квадрат							+	+	+	+	+	+	+	+
Прямоугольник	+	+	+	+	+	+								
Параллелограмм			+	+	+	+							+	+
Ромб	+	+				+				+	+	+		
Трапеция		+			+			+	+			+		
Круг	+			+			+		+		+			+
Эллипс			+				+	+		+			+	
Вариант	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**	**41**	**42**
Треугольник								+	+	+	+	+	+	+
Квадрат	+	+	+	+	+	+	+							
Прямоугольник			+	+	+	+	+							
Параллелограмм	+	+					+							+
Ромб		+				+					+	+	+	
Трапеция	+				+				+	+			+	
Круг				+				+		+		+		
Эллипс			+					+	+		+			+
Вариант	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**
Треугольник	+	+	+	+	+	+	+	+	+	+	+	+	+	+
Квадрат									+	+	+	+	+	+
Прямоугольник				+	+	+	+	+						+
Параллелограмм	+	+	+					+					+	
Ромб			+				+					+		
Трапеция		+				+					+			
Круг	+				+					+				
Эллипс				+					+					