Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Лабораторна робота №1

з дисципліни «Паралельне програмування -1.

Основи паралельного програмування»

Тема: **АДА. ПІДПРОГРАМИ ТА ПАКЕТИ**

Виконав

студент групи ІП-31

Кобилинський Дмитро

Київ 2015

**Цель работы:** изучение структуры программы и особенностей реализации механизма под-

программ и пакетов в языке Ада, который является базовым в дисциплине «ПП»

**Выполнение работы:** разработать программу на языке Ада, реализующую вычисления

трех функций F1, F2, F3, представленных соответствующим вариантом в Приложении Б.

Программа состоит из пакета Data и основной программы – процедуры Lab1. Пакет

реализует ресурсы, необходимые для вычисления функций F1, F2, F3 через подпрограммы

Func1, Func2, Func3.

При разработке процедур следует разделять формальные параметры на входные (in)

и выходные (out). Необходимо изучить команды и опции компилятора ObjectAda, необхо-

димые для компиляции и редактирования связей программы.

Разработать и реализовать библиотечный пакет Data , ресурсами которого являются

- подпрограммы Func1, Func2, Func3,,

- необходимые т и п ы (например, Vector и Matrix )

- дополнительные процедуры ввода/вывода (Vector\_Input, Vector\_Output, Matrix\_Input, Matrix\_Output.

**Вариант : 1.20 2.20 3.11**

D = MIN(A + B) \* (B + C) \*(MA\*MD)

MM = ML + MK \*MO

R = SORT(S + T)\*TRANS(MS\*MR)

GNAT GPL 2014 (20140331)

Copyright 1992-2014, Free Software Foundation, Inc.

Compiling: f\_main.adb (source file time stamp: 2015-09-21 21:01:58)

1. with Data, Ada.Text\_IO, Ada.Integer\_Text\_IO;

2. use Ada.Text\_IO, Ada.Integer\_Text\_IO;

3.

4. --Main program

5. --Parallel programming

6. --Kobylynskiy D.A.

7. --IP-31

8. --F1 = MIN(A + B) \* (B + C) \*(MA\*MD)

9. --F2 = ML + MK \*MO

10. --F3 = SORT(S + T)\*TRANS(MS\*MR)

11.

12. procedure f\_main is

13. n : Integer :=3;

14. package Lab1 is new data(n);

15. use Lab1;

16. A,B,C,D,S,T,R: Vec;

17. MA,MD,MF,MK,MO,MM,MS,MR: Matr;

18. begin

19. Put("All vectors had ");

20. Put(n);

21. Put(" items");

22. New\_Line;

23.

24. Put\_Line("Arguments for F1");

25. Put("Enter vector A");

26. New\_Line;

27. setVector(A);

28. New\_Line;

29. Put("Enter vector B");

30. New\_Line;

31. setVector(B);

32. New\_Line;

33. Put("Enter vector C");

34. New\_Line;

35. setVector(C);

36. New\_Line;

37.

38. Put("Enter MA");

39. New\_Line;

40. setMatrix(MA);

41. New\_Line;

42. Put("Enter MD");

43. New\_Line;

44. setMatrix(MD);

45.

46. F1(A,B,C,MA,MD,D);

47.

48. Put\_Line("Arguments for F2");

49. New\_Line;

50. Put("Enter MF");

51. New\_Line;

52. setMatrix(MF);

53. New\_Line;

54. Put("Enter MK");

55. New\_Line;

56. setMatrix(MK);

57. New\_Line;

58. Put("Enter MO");

59. New\_Line;

60. setMatrix(MO);

61.

62. F2(MF,MK,MO,MM);

63.

64.

65. Put\_Line("Arguments for F3");

66. Put("Enter MS");

67. New\_Line;

68. setMatrix(MS);

69. New\_Line;

70. Put("Enter MR");

71. New\_Line;

72. setMatrix(MR);

73. New\_Line;

74. Put("Enter vector T");

75. New\_Line;

76. setVector(T);

77. New\_Line;

78. Put("Enter vector S");

79. New\_Line;

80. setVector(S);

81.

82. F3(S,T,MS,MR,R);

83.

84. New\_Line;

85. Put("F1 = ");

86. getVector(D);

87. New\_Line;

88. Put("F2 = ");

89. getMatrix(MM);

90. New\_Line;

91. Put("F3 = ");

92. getVector(R);

93. end f\_main;

93 lines: No errors

GNAT GPL 2014 (20140331)

Copyright 1992-2014, Free Software Foundation, Inc.

Compiling: data.adb (source file time stamp: 2015-09-21 21:01:48)

1. with Ada.Text\_IO, Ada.Integer\_Text\_IO;

2. use Ada.Text\_IO, Ada.Integer\_Text\_IO;

3. with Ada.Streams; use Ada.Streams;

4. with System.Partition\_Interface; use System.Partition\_Interface;

5.

6. package body Data is

7.

8. --Procedure for set every element of vector some value

9. procedure setVector(A : out Vec) is

10. begin

11. for i in 1..N loop

12. Get(A(i));

13. end loop;

14. end setVector;

15. --Procedure for getting vector

16. procedure getVector(A : in Vec) is

17. begin

18. for i in 1..N loop

19. Put(A(i));

20. end loop;

21. end getVector;

22. --Procedure for set every element of matrix some value

23. procedure setMatrix(MA: out Matr) is

24. begin

25. for i in 1..N loop

26. Put("New row");

27. New\_Line;

28. for j in 1..N loop

29. Get(MA(i,j));

30. end loop;

31. end loop;

32. end setMatrix;

33. --Procedure for getting matrix

34. procedure getMatrix(MA: in Matr) is

35. begin

36. for i in 1..N loop

37. New\_Line;

38. for j in 1..N loop

39. Put(MA(i,j));

40. end loop;

41. end loop;

42. end getMatrix;

43.

44. --Function for adding two vectors

45. function SumVector (A, B :Vec) return Vec is

46. g : Vec;

47. begin

48. for index in 1..N loop

49. g(index):=A(index)+B(index);

50. end loop;

51. return g;

52. end SumVector;

53.

54. --Function for sorting vector

55. function Sort(A : in out Vec) return Vec is

56. k: Integer;

57. begin

58. for i in 1..N-1 loop

59. for l in 1..N-i loop

60. if A(l)>A(l+1) then

61. k:= A(l);

62. A(l):=A(l+1);

63. A(l+1):=k;

64. end if;

65. end loop;

66. end loop;

67. return A;

68. end Sort;

69.

70. --Function for multiply matrix

71. function MultiplyMatrix (A, B : Matr) return Matr is

72. temp : Integer;

73. Result : Matr;

74. begin

75. for i in 1..N loop

76. for j in 1..N loop

77. temp := 0;

78. for k in 1..N loop

79. temp := temp + A(i,k)\*B(k,j);

80. end loop;

81. Result(i,j):= temp;

82. end loop;

83. end loop;

84. return Result;

85. end MultiplyMatrix;

86.

87. --Function for transpose matrix

88. function Transpose (A : Matr) return Matr is

89. Result : Matr;

90. begin

91. for i in 1..N loop

92. for j in 1..N loop

93. Result(i,j) := A(j,i);

94. end loop;

95. end loop;

96. return Result;

97. end Transpose;

98.

99. --Function for search min element in matrix

100. function Min(A : Matr) return Integer is

101. Var : Integer := A(1,1);

102. begin

103. for i in 1..N loop

104. for j in 1..N loop

105. if A(i,j) < Var then

106. Var := A(i,j);

107. end if;

108. end loop;

109. end loop;

110. return Var;

111. end Min;

112.

113. --Function for search min element in vector

114. function MinVector(A : Vec) return Integer is

115. Var : Integer := A(1);

116. begin

117. for i in 1..N loop

118. if A(i) < Var then

119. Var := A(i);

120. end if;

121. end loop;

122. return Var;

123. end MinVector;

124.

125. --Function for adding matrix

126. function SumMatrix(A, B : Matr) return Matr is

127. Result : Matr;

128. begin

129. for i in 1..N loop

130. for j in 1..N loop

131. Result(i,j) := A(i,j) + B(i,j);

132. end loop;

133. end loop;

134. return Result;

135. end SumMatrix;

136.

137. --Function for multiply vector on matrix

138. function MultiplyVectorOnMatrix(A : Vec; B : Matr) return Vec is

139. temp : Integer;

140. Result : Vec;

141. begin

142. for j in 1..N loop

143. temp := 0;

144. for k in 1..N loop

145. temp := temp + A(k)\*B(k,j);

146. end loop;

147. Result(j):= temp;

148. end loop;

149. return Result;

150. end MultiplyVectorOnMatrix;

151.

152. --Function for multiply matrix on digit

153. function MultiplyOnDigit(a: Integer; MA: Matr) return Matr is

154. Result : Matr;

155. begin

156. for i in 1..N loop

157. for j in 1..N loop

158. Result(i,j):= a\*MA(i,j);

159. end loop;

160. end loop;

161. return Result;

162. end MultiplyOnDigit;

163.

164. --Function for multiply digit on vector

165. function MultiplyDigitOnVector(a: Integer; B: Vec) return Vec is

166. Result : Vec;

167. begin

168. for i in 1..N loop

169. Result(i):= a\*B(i);

170. end loop;

171. return Result;

172. end MultiplyDigitOnVector;

173.

174. procedure F1(A,B,C : in Vec; MA,MD: in Matr; D : out Vec) is

175. begin

176. D := MultiplyDigitOnVector(MinVector(SumVector(A,B)),

177. MultiplyVectorOnMatrix(SumVector(B,C),MultiplyMatrix(MA,MD)));

178. end F1;

179.

180. procedure F2(ML, MK, MO: in Matr; MM: out Matr) is

181. begin

182. MM := SumMatrix(ML,MultiplyMatrix(MK,MO));

183. end F2;

184.

185. procedure F3(S, T: in Vec; MS, MR: in Matr; R: out Vec) is

186. temp : Vec;

187. begin

188. temp:= SumVector(S,T);

189. New\_Line;

190. R:=MultiplyVectorOnMatrix(Sort(temp),Transpose(MultiplyMatrix(MS,MR)));

191. end F3;

192.

193.

194. end Data;

Compiling: data.ads (source file time stamp: 2015-09-21 21:01:48)

1. generic

2. N : in Integer;

3. package Data is

4.

5. type Vec is private;

6. type Matr is private;

7.

8. procedure F1(A,B,C : in Vec; MA,MD: in Matr; D : out Vec);

9.

10. procedure F2(ML, MK, MO: in Matr; MM: out Matr);

11.

12. procedure F3(S, T: in Vec; MS, MR: in Matr; R: out Vec);

13.

14. procedure setVector(A : out Vec);

15.

16. procedure getVector(A : in Vec);

17.

18. procedure setMatrix(MA: out Matr);

19.

20. procedure getMatrix(MA: in Matr);

21.

22. private

23. type Vec is array(1..N) of Integer;

24. type Matr is array(1..N, 1..N) of Integer;

25.

26.

27. end Data;

194 lines: No errors