

УДК 378.147

Методология внедрения экологической компоненты в структуру и содержание лабораторно-практических занятий учебных дисциплин по программированию для инженерных направлений и профилей в техническом вузе

Тюльпинова Н. В.

*Брянский государственный технический университет
Брянск, Россия
ninatulpinova@mail.ru*

Описана нормативно-правовая база национальной стратегии экологического образования. Отмечена необходимость развития у обучающихся способности оценивать результаты и последствия своей деятельности с точки зрения нанесения или минимизации вреда окружающей среде. Рассмотрены методологические и прикладные аспекты формирования экологической культуры обучающихся в междисциплинарном контексте. Отмечена ключевая роль программирования в организации межпредметных связей между экологией, информатикой и математикой. Выявлено связующее звено между программированием и экологическим экспериментом. Предложена методология внедрения экологической компоненты в структуру и содержание лабораторно-практических занятий в курсе программирования. Сделан акцент на включение обучающихся в познавательную-исследовательскую деятельность, направленную на изучение экологических проблем и поиск путей их решения, на переход от объяснительно-иллюстративного к деятельностному подходу в обучении. Представлен пример разработки экологического приложения. Приведен соответствующий программный код и верификационный протокол. Обозначен педагогический потенциал программирования в борьбе с экологическим невежеством и неумением видеть и чувствовать связь между своим самочувствием, здоровьем, безопасностью и процессами, происходящими в природной среде, поскольку основой общечеловеческой культуры третьего тысячелетия является экологическая культура и взгляд на себя как на творца нового экологосообразного мира, базирующегося на междисциплинарном научном знании.

Ключевые слова: экология, программирование, междисциплинарная связь, планирование эксперимента.

ВВЕДЕНИЕ

В настоящее время основополагающим документом в области охраны окружающей среды и обеспечения экологической безопасности в нашей стране являются «Основы государственной политики в области экологического развития Российской Федерации до 2030 года», утверждённые Президентом РФ 30 апреля 2012 года. Согласно этому документу, одной из основных задач государства является формирование экологической культуры, развитие экологического образования и воспитания. При этом в качестве механизма реализации данной задачи предусмотрено формирование у всех слоев населения, а прежде всего у молодежи, экологически ответственного мировоззрения; включение вопросов охраны окружающей среды в новые образовательные стандарты; обеспечение направленности процесса воспитания и обучения в образовательных учреждениях на формирование экологически ответственного поведения. В этой связи весьма актуальной является экологизация образовательного процесса в междисциплинарном контексте, поскольку вопросы экологии в той или иной степени могут быть освещены в рамках всех без исключения учебных дисциплин в контексте области знаний каждой конкретной дисциплины. В настоящей статье рассматривается методология внедрения экологической компоненты в процесс преподавания учебных дисциплин по программированию для инженерных направлений и профилей в техническом вузе, так как именно программирование позволяет эффективно организовать междисциплинарные связи между экологией, информатикой и математикой, а также дает возможность сформировать исследовательские навыки решения экологических задач.

Цель настоящих исследований – выработать методологию внедрения экологической компоненты в структуру и содержание лабораторно-практических занятий учебных дисциплин по программированию для инженерных направлений и профилей в техническом вузе

МАТЕРИАЛ И МЕТОДЫ

Основу курса программирования в техническом вузе составляет изучение алгоритмов обработки массивов (одномерных и двумерных, векторов и матриц), а основу исследований в экологии составляет планирование эксперимента. Матричный подход в теории планирования эксперимента – это то связующее звено между программированием и экологическим экспериментом, которое позволяет наполнить смысловым экологическим содержанием решение традиционных программистских задач обработки массивов: объявление массивов, заполнение массивов, обработка массивов, вывод массивов. Сущность предлагаемой методологии заключается в том, что обучающийся в рамках лабораторно-практических занятий по теме «Массивы» в курсе программирования создаёт не какие-то абстрактные программы из задачника по программированию, а разрабатывает прикладные программы, посвящённые заданной экологической проблеме и её решению методом планирования экологического эксперимента. Для удобства программирования наиболее подходит двухуровневый полный факторный эксперимент. Исходными данными для таких экологических приложений могут являться опубликованные научные результаты экологических экспериментов различной направленности (Брындина, Полянский, 2013; Антонова, 2016*a*, 2016*b*; Ищенко, 2016; Холодов, 2016; Житенев, Андреюк, 2019; Кузнецов и др., 2020; Кулакова и др., 2021).

РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Ниже представлена методология внедрения экологической компоненты в структуру и содержание лабораторно-практических занятий по теме «Массивы» в курсе программирования, включающая 10 этапов.

Этап №1. Изучение обучающимися методов и приемов разработки алгоритмов и программ решения типовых задач обработки массивов (Тюльпинова, 2019*a*, 2019*b*).

Этап №2. Подготовка индивидуальных заданий для обучающихся на разработку экологических приложений.

Этап №3. Подготовка числовых верификационных расчётных примеров для обучающихся (Тюльпинова, 2020*a*, 2020*b*).

Этап №4. Постановка экологической проблемы перед обучающимися, например, «Загрязнение воды нитратами».

Этап №5. Рассмотрение этой проблемы совместно с обучающимися по некоторому заранее разработанному плану, например: 1) вред здоровью человека вследствие загрязнения воды нитратами; 2) источники загрязнения воды нитратами; 3) методы очистки воды от загрязнения её нитратами.

Этап №6. По итогам вышеуказанного рассмотрения обучающиеся разрабатывают первую часть своего экологического приложения, тезисно излагая суть рассматриваемой проблемы. Пример организации пользовательского интерфейса представлен на рисунке 1*a–d*. Такое приложение следует разрабатывать в среде визуального программирования (например, в IDE Lazarus) в целях закрепления навыков как формирования внешнего вида программы и её интерфейса, так и написания программного кода, заставляющего работать элементы этого интерфейса.

Этап №7. Описание обучающимся экологического эксперимента, направленного на решение поставленной проблемы; представление числовых данных этого эксперимента; описание алгоритма обработки этого эксперимента с приложением числовых верификационных расчётных примеров, например, исследуется эффективность ионообменной очистки воды от нитратов в зависимости от трёх факторов: 1) скорости фильтрации воды; 2) отношения высоты загрузки фильтрационной колонки к её диаметру; 3) температуры очищаемой воды (Житенев, Андреюк, 2019; Кулакова и др. 2021).

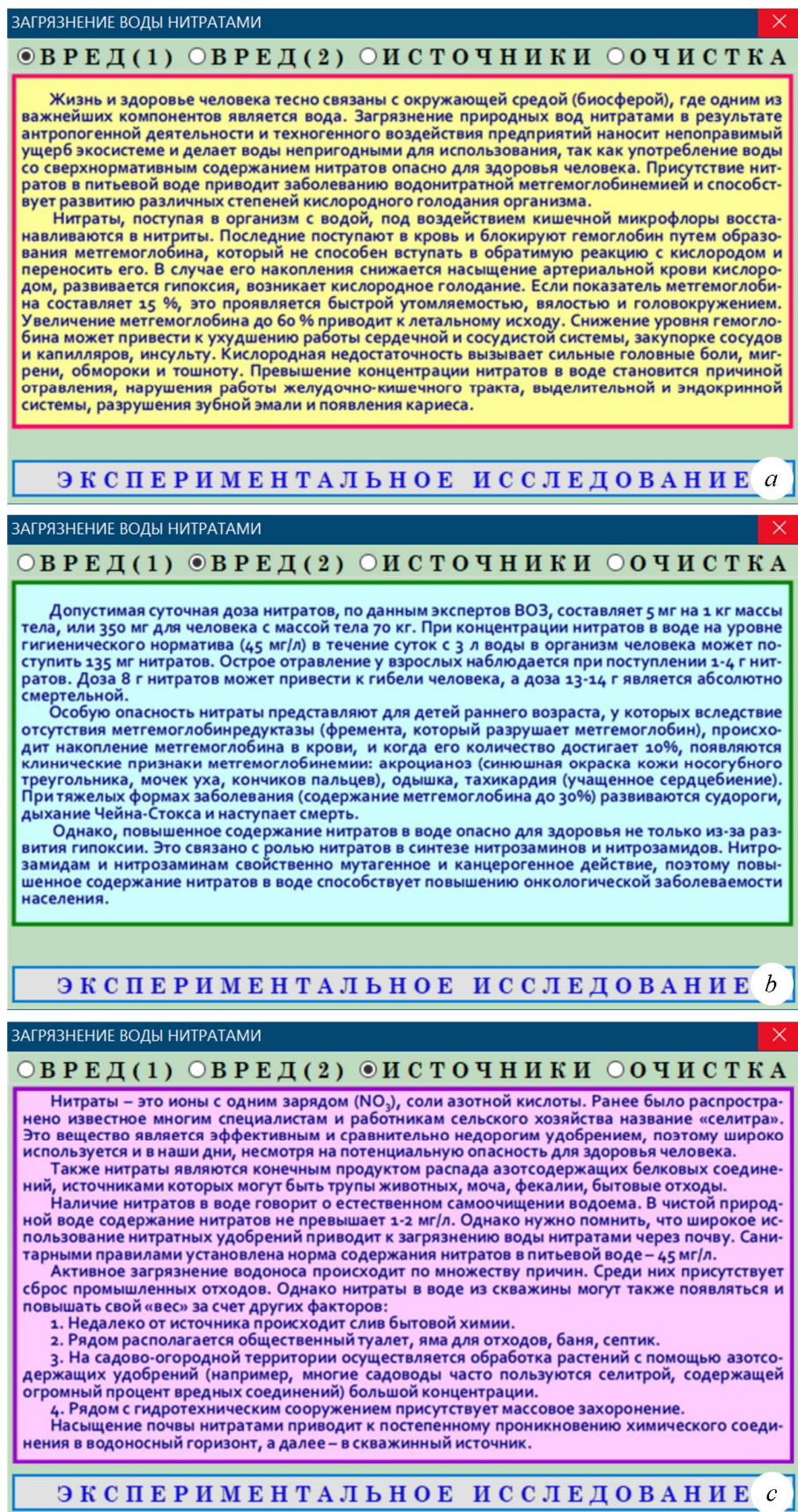


Рис. 1. Пользовательский интерфейс экологического приложения (начало)

ЗАГРЯЗНЕНИЕ ВОДЫ НИТРАТАМИ

○ ВРЕД (1) ○ ВРЕД (2) ○ ИСТОЧНИКИ ○ **ОЧИСТКА**

Исследование методов физико-химической очистки природных вод является актуальной задачей. Перспективным методом очистки вод является метод ионного обмена. Ионообменные фильтры обеспечивают максимальный уровень очистки. Они могут применяться не только для подготовки питьевой воды, но и для очистки промышленных стоков. Прочие методы не способны обеспечить достаточный уровень очистки.

Ионный обмен – это специфический случай сорбции заряженных частиц (ионов), когда поглощение одних ионов сопровождается выходом в раствор других ионов, входящих в состав сорбента. При этом ион, присутствие которого в воде нежелательно, фиксируется на сорбенте. Таким образом происходит «замещение» одних ионов на другие. Сорбенты, работающие по такому механизму, называются ионообменными материалами или ионитами.

Водоподготовку путем ионного обмена выполняют при помощи специальных фильтрующих устройств (ионообменных колонок) — сначала их заполняют ионитами, а потом запускают воду. Для улучшения характеристик фильтрующего устройства, работающего по принципу ионного обмена, необходимо экспериментальным путем изучить зависимость эффективности водоочистки от различных технологических и конструктивных факторов и затем подобрать оптимальные параметры.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ *d*

ЗАГРЯЗНЕНИЕ ВОДЫ НИТРАТАМИ



ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ПРОЦЕССА ОЧИСТКИ ВОДЫ ОТ НИТРАТОВ в зависимости от изменения следующих факторов работы ионообменной колонки:
 x1 – скорость фильтрации воды (диапазон: [10 м/час – 15 м/час])
 x2 – отношение высоты загрузки фильтрационной колонки к её диаметру (диапазон: [2 – 8])
 x3 – температура очищаемой воды (диапазон: [5 °С – 25 °С])

Метод исследований: полный факторный эксперимент ПФЭ 2³
 Число факторов: 3 Число уровней: 2 Число повторений: 3

Факторы очистки воды			Эффективность очистки воды (%)		
x1	x2	x3	y1	y2	y3
15	8	25	85.69	84.48	82.69
10	8	25	81.01	84.40	83.67
15	2	25	79.52	78.61	77.68
10	2	25	78.03	80.03	79.99
15	8	5	55.53	59.61	58.40
10	8	5	52.88	55.98	55.68
15	2	5	56.71	54.06	55.82
10	2	5	49.39	52.12	51.99

ОБРАБОТКА ЭКСПЕРИМЕНТА *e*

ЭФФЕКТИВНОСТЬ ПРОЦЕССА ОЧИСТКИ ВОДЫ ОТ НИТРАТОВ (%)

$$Y = 68.082 + 0.985X_1 + 1.920X_2 + 13.235X_3 - 0.856X_1X_3$$

ДИАГРАММА ВЛИЯНИЯ ФАКТОРОВ И ВЗАИМОДЕЙСТВИЙ ФАКТОРОВ

1: (+) X3 температура очищаемой воды
 2: (+) X2 отношение высоты загрузки колонки к её диаметру
 3: (+) X1 скорость фильтрации воды
 4: (-) X1X3 взаимодействие скорости фильтрации и температу*r*

f

Рис. 1. Пользовательский интерфейс экологического приложения (окончание)

Этап № 8. По итогам вышеуказанного описания обучающиеся разрабатывают вычислительную программу обработки экологического эксперимента. Пример программного кода на языке программирования Паскаль представлен на рисунке 2, верификационный пример – на рисунке 3. Данную программу следует разрабатывать как консоль-приложение (например, в IDE Free Pascal) в целях закрепления навыков работы с массивами.

```

1: program ecology;          98:
2: uses crt;                 99:
3:                             100: for i:=1 to N do
4: const                     101:   for j:=1 to N do
5:   N=8;                    102:     XT[i,j]:=X[j,i];
6:   M=3;                    103:
7:   G:=0.5157;              104:   for i:=1 to N do
8:   t:=2.1199;              105:     for j:=1 to N do
9:                             106:       begin
10: var                        107:         S:=0;
11: X,XT,MULT1:array[1..N,1..N] of integer; 108:         for k:=1 to N do
12: OBR,TEMP:array[1..N,1..N] of real;    109:           S:=S+XT[i,k]*X[k,j];
13: Yexp:array[1..N,1..M] of real;       110:         MULT1[i,j]:=S;
14: i,j,k,S,token:integer;              111:       end;
15: S2:array[1..N] of real;              112:   for i:=1 to N do
16: t:array[1..N] of boolean;            113:     for j:=1 to N do
17: i,j,k,S,token:integer;              114:       OBR[i,j]:=MULT1[i,j];
18: sum,maxS2,sumS2,G_r,S2Y,S2B1,SB1:real; 115:
19: f:text;                              116:   for k:=1 to N do
20: str1,str2,strG,strt,strY:string;     117:     begin
21:                                     118:       for i:=1 to N do
22: begin                            119:         for j:=1 to N do
23:   clrscr;                          120:           begin
24:                                     121:             if (i=k) and (j=k) then
25:   token:=1;                          122:               TEMP[i,j]:=1/OBR[i,j];
26:   for j:=1 to N do                    123:             if (i=k) and (j<k) then
27:     for i:=1 to N do                  124:               TEMP[i,j]:=-OBR[i,j]/OBR[k,k];
28:       begin                            125:             if (i<k) and (j=k) then
29:         case j of                      126:               TEMP[i,j]:=OBR[i,k]/OBR[k,k];
30:           1: X[i,j]:=1;                 127:             if (i<k) and (j<k) then
31:           2: if i mod 2 = 0 then        128:               TEMP[i,j]:=OBR[i,k]*OBR[k,k]/OBR[k,k];
32:             X[i,j]:=-1;                129:           end;
33:           else                          130:         for i:=1 to N do
34:             X[i,j]:=1;                 131:           for j:=1 to N do
35:           3: begin                       132:             OBR[i,j]:=TEMP[i,j];
36:             X[i,j]:=token;              133:           end;
37:             if i mod 2 = 0 then          134:         for i:=1 to N do
38:               token:=token*(-1);        135:           begin
39:             end;                        136:             sum:=0;
40:           4: begin                       137:               for k:=1 to N do
41:             X[i,j]:=token;              138:                 sum:=sum+XT[i,k]*Y[k];
42:             if i mod 4 = 0 then          139:                 MULT2[i]:=sum;
43:               token:=token*(-1);        140:             end;
44:             end;                        141:         for i:=1 to N do
45:           5: X[i,j]:=X[i,2]*X[i,3];      142:           begin
46:           6: X[i,j]:=X[i,2]*X[i,4];      143:             sum:=0;
47:           7: X[i,j]:=X[i,3]*X[i,4];      144:             for k:=1 to N do
48:           8: X[i,j]:=X[i,2]*X[i,3]*X[i,4]; 145:               sum:=sum+OBR[i,k]*MULT2[k];
49:           end;                          146:             B[i]:=sum;
50:         end;                            147:           end;
51:       assign(f,'exp.txt');              148:       end;
52:       reset(f);                          149:
53:       for i:=1 to N do                  150:         S2Y:=sumS2/N;
54:         for j:=1 to M do                151:         S2B1:=S2Y/(N*M);
55:           read(f,Yexp[i,j]);            152:         SB1:=sqrt(S2B1);
56:         close(f);                        153:
57:       for i:=1 to N do                  154:         for i:=1 to N do
58:         begin                            155:           t_r[i]:=abs(B[i])/SB1;
59:           sum:=0;                        156:         for i:=1 to N do
60:           for j:=1 to M do                157:           begin
61:             sum:=sum+Yexp[i,j];          158:             if t_r[i]>t then
62:           Y[i]:=sum/M;                    159:               t[i]:=TRUE;
63:         end;                              160:             else
64:       for i:=1 to N do                  161:               t[i]:=FALSE;
65:         begin                            162:             sum:=0;
66:           sum:=0;                        163:             for i:=1 to N do
67:           for j:=1 to M do                164:               if t[i]=TRUE then
68:             sum:=sum+sqrt(Yexp[i,j]-Y[i]); 165:                 begin
69:           S2[i]:=sum/(M-1);              166:                   str(B[i]:1:3,strt);
70:         end;                              167:                   if (B[i]>0) and (i<>1)
71:       maxS2:=S2[1];                      168:                     then strt:='+'+strt;
72:       for i:=2 to N do                    169:                   case i of
73:         if S2[i]>maxS2 then               170:                     1: strY:=strY+strt;
74:           maxS2:=S2[i];                  171:                     2: strY:=strY+strt+'X1';
75:       sumS2:=0;                           172:                     3: strY:=strY+strt+'X2';
76:       for i:=1 to N do                    173:                     4: strY:=strY+strt+'X3';
77:         sumS2:=sumS2+S2[i];              174:                     5: strY:=strY+strt+'X1X2';
78:       G_r:=maxS2/sumS2;                   175:                     6: strY:=strY+strt+'X1X3';
79:       if G_r<G_t then                     176:                     7: strY:=strY+strt+'X2X3';
80:         begin                            177:                     8: strY:=strY+strt+'X1X2X3';
81:           str(G_r:1:4,str1);              178:                   end;
82:           str(G_t:1:4,str2);              179:                 end;
83:           strG:='Gr=' +str1+'<Gt=' +str2; 180:               end;
84:         end;                              181:             end;
85:       else                               182:         assign(f,'report.txt');
86:         writeln('ERROR: INCREMENT M');    183:         rewrite(f);
87:         readkey;                          184:
88:         halt;                             185:
89:       end;                               186:       writeln(f,' X');
90:     end;                                  187:       for i:=1 to N do
91:     begin                                  188:       begin
92:       for j:=1 to N do                    189:         for j:=1 to N do
93:         write(f,X[i,j]:2);               190:           write(f,X[i,j]:2);
94:       end;                                  191:         writeln(f);
95:     end;                                  192:       end;
96:     halt;                                  193:
97:     end;                                  194:
195:     writeln(f,' Yexp');
196:     for i:=1 to N do
197:       begin
198:         for j:=1 to M do
199:           write(f,Yexp[i,j]:6:2);
200:         writeln(f);
201:       end;
202:     writeln(f,' Y');
203:     for i:=1 to N do
204:       for i:=1 to N do
205:         writeln(f,S2[i]:6:3);
206:       writeln(f,' S2');
207:       for i:=1 to N do
208:         writeln(f,S2[i]:6:3);
209:       writeln(f,' maxS2=',maxS2:1:3);
210:       writeln(f,' sumS2=',sumS2:1:3);
211:       writeln(f,' G_r=',G_r:1:4);
212:       writeln(f,' ',strG);
213:       writeln(f,' XT');
214:       for i:=1 to N do
215:         begin
216:           for j:=1 to N do
217:             write(f,XT[i,j]:2);
218:           writeln(f);
219:         end;
220:       writeln(f,' MULT1');
221:       for i:=1 to N do
222:         begin
223:           for j:=1 to N do
224:             write(f,MULT1[i,j]:2);
225:           writeln(f);
226:         end;
227:       writeln(f,' OBR');
228:       for i:=1 to N do
229:         begin
230:           for j:=1 to N do
231:             write(f,OBR[i,j]:6:3);
232:           writeln(f);
233:         end;
234:       writeln(f,' B');
235:       for i:=1 to N do
236:         begin
237:           for j:=1 to N do
238:             write(f,B[i,j]:7:3);
239:           writeln(f);
240:         end;
241:       writeln(f,' t_r');
242:       for i:=1 to N do
243:         write(f,t_r[i]:9:4);
244:       writeln(f,' t_t=',t_t:1:4);
245:       writeln(f,' t');
246:       for i:=1 to N do
247:         write(f,' ',t[i]);
248:       write(f,' ',strY);
249:       close(f);
250:       writeln(strY);
251:       writeln('Пауза записан в файл "report.txt"!');
252:       readkey;
253:     end.
254:
255:
256:
257:
258:
259:
260:
261:
262:
263:
264:
265:
266:
267:
268:
269:
270:
271:
272:
273:
274:

```

Рис. 2. Программный код экологического приложения

X	Yexp	Y	S2
1 1 1 1 1 1 1 1	85.69 84.48 82.69	84.287	2.278
1 -1 1 1 -1 -1 1 -1	81.01 84.40 83.67	83.027	3.183
1 1 -1 1 -1 1 -1 -1	79.52 78.61 77.68	78.603	0.846
1 -1 -1 1 1 -1 -1 1	78.03 80.03 79.99	79.350	1.307
1 1 1 -1 1 -1 -1 -1	55.53 59.61 58.40	57.847	4.391
1 -1 1 -1 -1 1 -1 1	52.88 55.98 55.68	54.847	2.923
1 1 -1 -1 -1 -1 1 1	56.71 54.06 55.82	55.530	1.819
1 -1 -1 -1 1 1 1 -1	49.39 52.12 51.99	51.167	2.372

$B = (X^T \cdot X)^{-1} \cdot (X^T \cdot Y)$								$MULT1 = X^T \cdot X$																																																																																																																																																																																																											
$XT = X^T$ maxS2=4.391 sumS2=19.120 Критерий Кохрена G_r=0.2297 Gr=0.2297< Gt=0.5157								<table border="1"> <tr><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>8</td></tr> </table>								8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
8	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																						
0	8	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																						
0	0	8	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																						
0	0	0	8	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																						
0	0	0	0	8	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																						
0	0	0	0	0	8	0	0	0	0	0	0	0	0																																																																																																																																																																																																						
0	0	0	0	0	0	8	0	0	0	0	0	0	0																																																																																																																																																																																																						
0	0	0	0	0	0	0	8	0	0	0	0	0	0																																																																																																																																																																																																						
0	0	0	0	0	0	0	0	8	0	0	0	0	0																																																																																																																																																																																																						
0	0	0	0	0	0	0	0	0	8	0	0	0	0																																																																																																																																																																																																						
0	0	0	0	0	0	0	0	0	0	8	0	0	0																																																																																																																																																																																																						
0	0	0	0	0	0	0	0	0	0	0	8	0	0																																																																																																																																																																																																						
0	0	0	0	0	0	0	0	0	0	0	0	8	0																																																																																																																																																																																																						
0	0	0	0	0	0	0	0	0	0	0	0	0	8																																																																																																																																																																																																						

$OBR = (X^T \cdot X)^{-1}$								$MULT2 = X^T \cdot Y$							
0.125	0.000	0.000	0.000	0.000	0.000	0.000	0.000	544.657							
0.000	0.125	0.000	0.000	0.000	0.000	0.000	0.000	7.877							
0.000	0.000	0.125	0.000	0.000	0.000	0.000	0.000	15.357							
0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.000	105.877							
0.000	0.000	0.000	0.000	0.125	0.000	0.000	0.000	0.643							
0.000	0.000	0.000	0.000	0.000	0.125	0.000	0.000	-6.850							
0.000	0.000	0.000	0.000	0.000	0.000	0.125	0.000	3.363							
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.125	3.370							

$B = OBR \cdot MULT2$		t_r		
68.082	S2Y=2.390	215.7444	> t t	TRUE
0.985	S2Bi=0.100	3.1200	> t t	TRUE
1.920	SBi=0.316	6.0829	> t t	TRUE
13.235	Критерий	41.9389	> t t	TRUE
0.080	Стъюдента	0.2548	< t t	FALSE
-0.856	t_t=2.1199	2.7134	> t t	TRUE
0.420		1.3323	< t t	FALSE
0.421		1.3349	< t t	FALSE

$Y = 68.082 + 0.985X_1 + 1.920X_2 + 13.235X_3 - 0.856X_1X_3$

Рис. 3. Верификационный пример экологического приложения

С точки зрения программирования, в программном коде (рис. 2) объявлению массивов посвящены строки 10–20, заполнению массивов – строки 25–57, обработке массивов – строки 59–181, выводу массивов – строки 183–265.

С точки зрения экологического эксперимента, в программном коде (рис. 2) формированию матрицы планирования эксперимента посвящены строки 25–50, чтению из файла результатов эксперимента – строки 52–57, проверке однородности дисперсий по критерию Кохрена – строки 59–97, расчёту и проверке значимости коэффициентов уравнения регрессии по критерию Стьюдента – строки 99–181. В целях сокращения трудоёмкости программирования проверку адекватности модели по критерию Фишера можно опустить.

Результатом работы данного консоль-приложения является уравнение регрессии в безразмерной форме, а также числовой протокол всех расчётов, записанный файл (рис. 3).

Этап № 9. Если вышеуказанное консоль-приложение работает верно, то его программный код может быть использован для разработки второй части экологического приложения в IDE Lazarus (рис. 1 *e–f*), при этом обучающиеся реализуют не только вывод итоговой зависимости эффективности очистки воды от исследуемых факторов, но и представляют полученный результат в виде диаграммы влияния факторов и их взаимодействий.

Этап № 10. Анализ обучающимся полученной регрессионной модели и подготовка выводов с учётом интерпретации факторов и межфакторных взаимодействий.

ЗАКЛЮЧЕНИЕ

Экологическое образование обучающихся только тогда может быть высокоэффективным, когда различные аспекты его содержания раскрываются во взаимодействии учебных дисциплин. Представленная методология экологизации программирования направлена на преодоление экологического невежества, ведь, как известно, именно невежество, в смысле отсутствия или недостатка экологического образования, является причиной аварий, катастроф и бедствий, а в широком смысле – причиной глобального экологического кризиса. Это чрезвычайно важно, так как перед современными выпускниками технических вузов, которые ответственны за дальнейшее развитие техники и технологий, стоит задача по реализации этих технологий таким образом, чтобы минимизировать воздействие на окружающую среду.

Список литературы

Антонова Е. В. Исследование возможности применения скрытокристаллического графита для очистки вод от повышенного содержания железа // Проспект Свободный-2016: сборник матер. Междунар. конф. студ., асп. и мол. уч., посвященной Году образования в Содружестве Независимых Государств (Красноярск, 15-25 апреля 2016 г.). – Красноярск, 2016. – Зеленая металлургия. – С. 7–11.

Антонова Е. В., Бурцева Э. Н. Исследование возможности применения наноструктурированной композиции «алюминий – графит» для очистки отработанных вод от повышенного содержания железа // Проспект Свободный-2016: сборник матер. Междунар. конф. студ., асп. и мол. уч., посвященной Году образования в Содружестве Независимых Государств (Красноярск, 15-25 апреля 2016 г.). – Красноярск, 2016. – Зеленая металлургия. – С. 12–18.

Брындина Л. В., Полянский К. К. Математическое обоснование условий биосорбционной очистки белковосодержащих отработанных технологических вод // Вестник Томского государственного университета. – 2013. – Т. 18, Вып. 4. – С. 1466 – 1470.

Житенев Б. Н., Андреев С. В. Планирование многофакторного эксперимента на примере ионообменной очистки воды от нитратов // Вестник Брестского государственного технического университета. Водохозяйственное строительство, теплоэнергетика и геоэкология. – 2019. – № 2 (115). – С. 38–43.

Ищенко Е. П. Очистка нефтезагрязненных почв с использованием лузги подсолнечника: дис. ... канд. техн. наук: спец. 06.01.02 Мелиорация, рекультивация и охрана земель. – Самара: Самарский государственный технический университет, 2016. – 132 с.

Кузнецов С. И., Венгер Е. А., Мищенко Е. В., Куликова И. О. Абсорбционный метод нейтрализации сернистого ангидрида // Вестник Херсонского национального технического университета. – 2020. – № 2 (73). – С. 23–33.

Кулакова С. И., Подлипенская Л. Е., Мельничук Д. А. Организация и математическое планирование эксперимента. – Алчевск: ДонГТИ, 2021. – 121 с.

Тюльпинова Н. В. Алгоритмизация и программирование. – Саратов: Вузовское образование, 2019. – 200 с.

Тюльпинова Н. В. Технология алгоритмизации и программирования на языке Pascal. – Саратов: Вузовское образование, 2019. – 244 с.

Тюльпинова Н. В. Программный комплекс сквозной автоматизации имитационного моделирования бизнес-процессов // Наука Красноярья. – 2020. – Т. 9, № 1. – С. 184–198.

Тюльпинова Н. В., Ромашенкова А. А. Прогнозирование параметров качества процесса кислородной резки металла методами теории планирования // Современные материалы, техника и технологии. – 2020. – № 3 (30). – С. 71–77.

Холодов В. И. Планирование экспериментов в гидробиологических исследованиях. – Симферополь: Н. Оріанда, 2016. – 196 с.

Tyulpinova N. V. Methodology for introduction the environmental component into the structure and content of laboratory and practical classes of academic disciplines in programming for engineering profiles in technical university // Ekosistemy. 2023. Iss. 34. P. 208–215.

The regulatory framework of the national strategy of ecological education is described. The necessity of developing students' ability to assess the results and consequences of their activities from the point of view of non-inflation or minimization of harm to the environment is noted. Methodological and applied aspects of the formation of ecological culture of students in an interdisciplinary context are considered. The key role of programming in the organization of interdisciplinary connections between ecology, computer science and mathematics is noted. The link between programming and environmental experiment is revealed. The methodology of introducing an ecological component into the structure and content of laboratory and practical classes in the programming course is proposed. The emphasis is placed on the inclusion of students in cognitive research activities aimed at studying environmental problems and finding ways to solve them, on the transition from an explanatory and illustrative to an activity-based approach in teaching. An example of the design of an ecological application is presented. The corresponding program code and verification protocol are given. The pedagogical potential of programming in the fight against environmental ignorance and inability to see and feel the connection between one's well-being, health, safety and the processes taking place in the natural environment is indicated, since the basis of the universal culture of the third millennium is ecological culture and the view of oneself as the creator of a new ecological-like world based on interdisciplinary scientific knowledge.

Key words: ecology, programming, interdisciplinary communication, design of experiment.

Поступила в редакцию 02.12.22

Принята к печати 30.12.22