

$$\mu(b_i) = \exp \left\{ -\frac{b_i - b_i^{(0)^2}}{2\sigma_i^2} \right\}, \quad (3)$$

где  $b_i^{(0)}$  модальные значения (ядра) нечетких чисел  $b_i$ ,  $\sigma_i^2$  – квадрат коэффициента концентрации.

Решения этой задачи ННП начинают из построения области допустимых решений (ОДР). Из графической точки зрения такая область представляет собой пересечения нечетких полуплоскостей (или нечетких прямых), которые определяются системой (2).

На этом же графике строим семейство целевых функций (1). Так как в рассматриваемом случае ОДР представляет собой нечеткий выпуклый многогранник, то в задачах такого типа нечеткая точка экстремума является вершиной такого нечеткого многогранника. Таким образом, искомое экстремальное решения, которое графически соответствует координатам нечеткой точки пересечения двух определенных нечетких прямых, можно найти путем общего решения системы двух нечетких уравнений, что соответствуют этим предельным нечетким прямым, то есть в общем случае нужно найти нечеткое решение нечеткой системы линейных алгебраических уравнений (НСЛАУ) [5].

Выразим решения НСЛАУ (2) через параметры задачи по формулам Крамера:

$$x_k = \frac{\det A_k}{\det A}, k = 1, 2, \dots, m, \quad (4)$$

где  $\det A$  – определитель матрицы  $A = (a_{ij})$ ,  $\det A_k$  – определитель матрицы, который получается при замене  $k$ -го столбца матрицы  $A$  столбцом свободных членов ( $b_i$ ).

В рассматриваемом случае нечеткими являются только свободные члены  $b_i$ . Значения переменных  $x_k$ , что вычисляются по формуле (4), запишем, раскрывая определители  $\det A_k$  по элементам  $k$ -го столбца:

$$x_k = \frac{\sum_{i=1}^m A_{ik} b_i}{\det A} = \frac{\sum_{i=1}^m A_{ik}}{\det A} b_i = \sum_{i=1}^m d_{ik} b_i, \quad (5)$$

где  $d_{ik} = \frac{A_{ik}}{\det A}$ ;  $A_{ik}$  – адьюнкта элемента  $a_{ik}$  матрицы  $A$ ,  $k = 1, 2, \dots, n$ .

Теперь можно получить функции принадлежности компонентов нечеткого решения задачи [5]:

$$\mu(x_k) = \mu \left( \sum_{i=1}^m d_{ik} b_i \right) = \exp \left\{ -\frac{(x_k - m_k)^2}{2D_k} \right\};$$

$$m_k = \sum_{i=1}^m d_{ik} b_i^{(0)};$$

$$D_k = \sum_{i=1}^m d_{ik}^2 \sigma_i^2, k = 1, 2, \dots, m.$$

Подставив полученные значения нечетких координат точек экстремума в выражения для

целевой функции (1), можно получить нечеткие значения искомой функции.

Если все неравенства нечетких ограничений преобразовать в уравнения, то получим нечеткую задачу на условный экстремум, которая может быть решена методом множителей Лагранжа.

В данной работе изложены достаточно простые графоаналитические методы определения нечетких координат нечетких экстремальных точек путем нахождения нечеткого решения НСЛАУ. Полученные в работе результаты в дальнейшем могут быть использованы для геометрической интерпретации задач нечеткого целочисленного программирования и теории нечетких игр.

#### Список литературы

1. Барішевський С.О. Основи теорії точкових нечітких множин // Праці: Таврійський державний агротехнологічний університет. – Мелітополь: ТДАТУ, 2012. – Вип. 4. – Т. 52. – С. 141–144.
2. Барішевський С.О. Точкові нечіткі множини та їх відображення // Праці: Таврійський державний агротехнологічний університет. – Мелітополь: ТДАТУ, 2012. – Вип. 4. – Т. 54. – С. 3–8.
3. Барішевський С.О. Основи теорії точечних нечетких множеств: Алгебраические и топологические аспекты // Праці: Таврійський державний агротехнологічний університет. – Мелітополь: ТДАТУ. – Вип. 4. – Т. 57. – С. 22–27.
4. Барішевський С.О. Аксиоматичні основи евклідової нечіткої планіметрії: аксіоми нечіткого простору // Сучасні проблеми моделювання: зб. наук. праць / МДПУ ім. Б. Хмельницького; гол. ред. кол. А.В. Найдиш. – Мелітополь: Видавництво МДПУ ім. Б. Хмельницького, 2014. – Вип. 1. – С. 13–16.
5. Раскин Л.Г., Серая О.В. Нечеткая математика. Основы теории. Приложения. – Х.: Парус, 2008. – 352 с.
6. Берштейн Л.С., Боженюк А.В. Нечеткие графы и гиперграфы. – М.: Научный мир, 2005. – 256 с.

#### ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++ ДЛЯ КАРТОГРАФОВ И ГЕОДЕЗИСТОВ УЧЕБНАЯ ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ ПРОГРАММА «НИВЕЛИРНАЯ РЕЙКА»

Заблоцкий В.Р.

Московский государственный университет геодезии и картографии, Москва, e-mail: v-r-zablotskii@ya.ru

Обсуждается программа для студентов, изучающих программирование на C++ и знакомых с основами общей геодезии. В учебно-методической литературе, к сожалению, отсутствуют объектно-ориентированные программы, связанные с задачами картографии и геодезии. Представлены лишь программы на основе функционального программирования [1]. Цель работы заключалась в разработке программы, которая моделирует технику определения превышения одной точки местности над другой с помощью геометрического нивелирования. Рассмотрим содержательную постановку задачи. В выбранных точках местности размещаются вертикально нивелирные рейки, между ними устанавливается нивелир. Наводя зрительную трубу нивелира на рейки, берут отсчеты.

Сначала берут отсчет по черной стороне задней рейки, затем рейку поворачивают красной стороной и берут отсчет по красной стороне. Затем трубу наводят на переднюю рейку и снова берут отсчеты по черной и красной стороне рейки. Превышение передней точки вычисляют как разность двух отсчетов и из двух измерений находят среднее значение.

Последовательность, описанных выше шагов выполнения геометрического нивелирования, представлена в программе «НИВЕЛИРНАЯ РЕЙКА». В ней реализован класс нивелирная рейка *LevelingStaff*, содержащий общие методы: «установить рейку в контрольной точке» *SetUpStaffAtControlPoint* и «повернуть рейку красной стороной к прибору» *RotateTheStaffRedFace*, а также частные элементы данных – целочислен-

ные переменные, *longOfStaff* для хранения длины рейки и *zeroOfRedFace* – для начального отсчета на пятке красной стороны рейки. Определение, указанных выше методов класса, приводится вне класса в строках 19–24 и 25–28. Отметим также, что в строках 29–36 находится определение функции «направить трубу прибора на рейку и взять отсчет» *PointTelescopeToStaffAndGetReading*. Главная функция определяется в строках 37–65. В строке 42 создаются две нивелирные рейки и конструктору передаются два начальных значения: длина рейки – 3 м и начальный отсчет на пятке красной стороны рейки – 4787. С помощью функций библиотеки времени выполнения *strcpy* устанавливаются начальные значения символьного массива *model* равное «Двусторонняя деревянная рейка».

```

1: #include <iostream>
02: #include <string>
03: using namespace std;
04:
05: class LevelingStaff {
06:     public:
07:         LevelingStaff(int longOfStaff, int zeroOfRedFace)
08:         {
09:             LevelingStaff::longOfStaff = longOfStaff;
10:             LevelingStaff::zeroOfRedFace = zeroOfRedFace;
11:             strcpy(model, "Двусторонняя деревянная рейка");
12:         }
13:         void SetUpStaffAtControlPoint(char []);
14:         void RotateTheStaffRedFace(char []);
15:     private: char model [30];
16:             int longOfStaff;
17:             int zeroOfRedFace;
18: };
19: void LevelingStaff::SetUpStaffAtControlPoint(char string[])
20: {
21:     cout << string << " рейка установлена вертикально на контрольной "
22:         "точке и повернута черной стороной к прибору."
23:         << endl;
24: };
25: void LevelingStaff::RotateTheStaffRedFace(char string[])
26: {
27:     cout << string << " рейка повернута красной стороной к прибору.";
28: }
29: int PointTelescopeToStaffAndGetReading(char string[])
30: {
31:     int h;
32:     cout << string << " рейка: Введите отсчет по рейке ";
33:     cin >> h;
34:     return h;
35: }
36: }
37: int main(void)
38: {
39:     int valueFromBlackFaceOfStaff1, valueFromBlackFaceOfStaff2;
40:     int valueFromRedFaceOfStaff1, valueFromRedFaceOfStaff2;
41:
42:     LevelingStaff Staff1(3,4787),Staff2(3,4787) ;
43:
44:     Staff1.SetUpStaffAtControlPoint("Задняя");
45:     Staff2.SetUpStaffAtControlPoint("Передняя");

```

```

46:
47:   valueFromBlackFaceOfStaff1 =
48:       PointTelescopeToStaffAndGetReading ("Задняя");
49:   Staff1.RotateTheStaffRedFace ("Задняя");
50:   valueFromRedFaceOfStaff1 =
51:       PointTelescopeToStaffAndGetReading ("Задняя");
52:
53:   valueFromBlackFaceOfStaff2 =
54:       PointTelescopeToStaffAndGetReading ("Передняя");
55:   Staff2.RotateTheStaffRedFace ("Передняя");
56:   valueFromRedFaceOfStaff2 =
57:       PointTelescopeToStaffAndGetReading ("Передняя");
58:
59:   int h1,h2;
60:   h1 = valueFromBlackFaceOfStaff1 - valueFromBlackFaceOfStaff2;
61:   h2 = valueFromRedFaceOfStaff1 - valueFromRedFaceOfStaff2;
62:   cout <<"Среднее превышение h = " << (h1+h2)/2. << endl;
63:
64:   return 0;
65: }

```

Поскольку все элементы члены – класса отмечены как имеющие тип *private*, то для доступа к ним должны использоваться интерфейсные функции, которые в данной версии программы не предусмотрены. Далее в строках 44 и 45 рейки устанавливаются в контрольных точках, причем рейка *Staff1* определяется как задняя, а рейка *Staff2* как передняя. Затем, в строках 47–48, вызывается метод класса, чтобы направить трубу прибора на рейку и взять отсчет *PointTelescopeToStaffAndGetReading*, величина отсчета сохраняется в переменной *valueFromBlackFaceOfStaff1*. В строке 49 вызывается метод, чтобы повернуть рейку *Staff1* красной стороной *RotateTheStaffRedFace* к трубе нивелира. Далее в строках 50–51 выполняется отсчет по красной стороне рейки, и значение сохраняется в переменной *valueFromRedFaceOfStaff1*. Код в строках 53–57 повторяет действия, рассмотренные ранее для задней рейки, но теперь они выполняются для передней рейки. Затем в строке 60 вычисляется превышение точки (целочисленная переменная *h1*) по черной стороне рейки, а в строке 61 по контрольной стороне рейки. Среднее значение величины превышения рассчитывается и выводится на экран в строке 62. На этом выполнение программы заканчивается.

**Выводы.** Разработана учебная объектно-ориентированная программа для практических занятий по программированию на C++. В программе реализован класс нивелирная рейка, на его основе создаются два объекта – задняя и передняя рейка и моделируется техника геометрического нивелирования для определения превышения одной точки местности над другой. Данная программа подчеркивает преимущества объектно-ориентированного программирования на основе задачи геометрического нивелирования.

#### Список литературы

1. Заблоцкий В.Р. Фам Суан Хоан Программирование учебных геодезических задач в среде BORLAND C++ BUILDER 6 (консольные приложения) // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2008. – № 4. – С. 81–89.

#### ОЦЕНКА ДИНАМИЧЕСКИХ ХАРАКТЕРИСТИК МАТЕРИАЛОВ И МЕХАНИЧЕСКИХ СИСТЕМ ЭКСПЕРИМЕНТАЛЬНЫМИ МЕТОДАМИ

<sup>1</sup>Захезин А.М., <sup>1</sup>Колосова О.П.,

<sup>1</sup>Воителев П.Ю., <sup>2</sup>Пакулев М.В.

<sup>1</sup>Южно-Уральский государственный университет,  
Челябинск, e-mail: yuri\_pryad@mail.ru;

<sup>2</sup>СоюзТехГаз

Сотрудники кафедры «Теоретическая механика» к.т.н., доц. Захезин А.М., к.т.н., доц. Малышева Т.В., к.т.н., доцент Колосова О.П., к.т.н. Иванов Д.Ю., Воителев П.Ю. Южно-Уральского государственного университета разработали рабочие чертежи и собрали аппаратно-компьютерный комплекс лабораторных работ по курсам теоретической и прикладной механики с применением виброизмерительной аппаратуры фирм «Роботрон» и «Брюль и Кьер», соединенной по каналам связи через многоканальный синхронный регистратор с ЭВМ, использующей современные пакеты прикладных программ MATLAB и MATCAD, ПО «Атлант». Отдельные работы разработаны совместно с М.В. Пакулевым «СоюзТехГаз».

Лабораторные работы проводятся вузовско-академической лабораторией «Диагностика машин» УрО АН, в курсе преподавания теоретической и прикладной механики для отдельных специальностей Южно-Уральского государственного университета. Цикл лабораторных работ [1] позволяет студентам