Десятилетний педагогический эксперимент по обучению C++ программированию студентов МИИГАиК на основе геодезических и картографических задач

Заблоцкий В.Р. (МИИГАиК)
В 1976 г. окончил факультет почвоведения Московского государственного университета им. М.В. Ломоносова, по специальности «почвовед – агрохимик». После окончания университета работал во ВНИЦ «АИУС-агроресурсы». С 1999 г. работает в МИИГАиК, доцент. Кандидат биологических наук.

A ten-year pedagogical experiment on teaching C++ programming of MIIGAiK students based on geodetic and cartographic problems

Zablotskii V.R. (*Moscow State University of Geodesy and Cartography, Moscow, Russia)*
zablotskii@miigaik.ru

Аннотация. Разработана программа для студентов, изучающих C++ программирование на основе вводного учебного курса. Программа вычисляет обратный географический азимут некоторой линии от прямого географического азимута и сближения меридианов для точек начала и конца линии. Программа может использоваться в курсе информатики при изучении сложных логических выражений и для иллюстрации технологии процедурного программирования.

Summary. A computer program has been developed for students studying the C++ programming under elementary training course. The program computes the reverse geographic azimuth of line from the direct geographic azimuth and convergence of the meridians for start and end points of line. The developed program can be used in a computer science course to study the complex logical expressions and illustrate procedural programming technology.

Введение

В Московском университете геодезии и картографии обучение студентов младших курсов C++ программированию выполняется по новому, специально адаптированному для будущих картографов и геодезистов, учебному курсу. Разработка данного учебного курса продолжалась в течение более 10 лет. Первоначально были отобраны геодезические и картографические задачи, затем разработаны или адаптированы алгоритмы компьютерного решения этих задач. Основное требование к задачам заключалось в том, чтобы задачи были простыми и понятными для студентов младших курсов. Часто выбирались вычислительные задачи, с которыми студенты имеют дело на практических занятиях по геодезии и картографии. Далее для выбранных задач были написаны компьютерные программы, при этом стремились, чтобы геодезическое содержание программ в максимальной степени иллюстрировалось конструкциями языка программирования C++. Была определена последовательность, в которой студентам предлагаются компьютерные программы для изучения. Разработанный набор программ соответствует кругу изучаемых вопросов в курсе общей геодезии и картографии. В результате специального подбора программ и геодезических задач студенты, приходящие на занятия по программированию, понимают геодезическое содержание программ, благодаря тому, что они уже изучили эти вопросы в курсе общей геодезии.

Подчеркнем, что анализ учебных компьютерных программ и задания по программированию предлагаются студентам после того, как они познакомились с этими задачами в курсе геодезии. Поэтому вначале учебного курса по программированию на C++ студентам не предлагаются компьютерные программы, связанные с работой измерительных геодезических инструментов или с топографической съемкой. Программы на эти темами изучаются в конце учебного курса. Жесткие ограничения, конечно, отсутствуют, однако изменение последовательности программ потребует от преподавателя информатики дополнительных усилий для подготовки студентов и разъяснению геодезической сущности предлагаемых задач.

Процесс обучения студентов на семинарских занятиях включает запуск программы на персональном компьютере и последующий анализ инструкций программы. Для этой цели применяется интерактивная электронная доска, управляемая компьютером преподавателя. На доске визуализируется код программы, которую студенты должны освоить, может быть доработать, исправить синтаксические или логические ошибки, а затем скомпилировать и запустить. Для проверки работы программы используется готовый контрольный пример, позволяющий быстро определить справился студент с заданием или нет. В начале курса используются достаточно простые компьютерные программы с геодезическим содержанием, например, требуется вычислить горизонтальное расстояние между пунктами на карте, рассчитать высоту пункта или вычислить уклон линии ската и т.д. Когда полученный студентом и контрольный результат совпадают, студент переходит на второй этап изучения программы. С целью развития навыков C++ программирования и анализа кода студент выполняет устный анализ кода программы. Естественно, что в начале

семестра программы содержат всего несколько строк, потом количество строк кода возрастает. Как показывает практика, оптимальный размер учебных программ составляет 50-70 строк кода. Устный анализ кода подразумевает объяснение каждой инструкции кода и ответ на вопрос для чего нужна и что делает данная строка инструкции. Все компьютерные программы сопровождаются справочно-информационным материалом, поясняющим назначение каждой инструкции программы и предназначенным для самостоятельного анализа программы студентом.

Печатание кода программы нельзя считать малополезным занятием, не вызывает же сомнение польза от подготовки конспекта изучаемой темы учебника. При печатании кода происходит запоминание написания ключевых слов, изучение формата конструкций языка программирования. Студент получает опыт работы со средой программирования, что является необходимым условием написания и запуска программ. При этом у студента накапливается опыт создания программ, редактирования и поиска ошибок в программе. Для поиска ошибок разработаны специальные эффективные схемы, в процессе печати кода и работы с кодом студент знакомится с ними. На семинарских занятиях можно предлагать студентам работать как с готовыми программами, так и с программами в которые искусственно добавлены ошибки. При выполнении таких учебных заданий, студенты тренируются в обнаружении и исправлении синтаксических ошибок. Такие задания направлены на изучение языка программирования, особенностей использования управляющий инструкций и т.д.

Обеспечение литературой - важный показатель развития учебной дисциплины и надо отметить, что сейчас имеется достаточно много учебников по программированию на языке C++, некоторые, наиболее популярные из них, представлены в [1-2]. Тем не менее, отсутствуют учебники, ориентированные на подготовку профильных инженеров, например картографов и геодезистов. Далее рассматривается учебная программа, предназначенная для студентов картографов и геодезистов, изучающих программирование в ходе вводного курса по программированию на языке C++. Программа демонстрирует использование сложных логических выражений в инструкции *if – else*.

Постановка задачи

Для разработки программы использовалась операционная система Microsoft Windows 10 на персональном компьютере. Также применялась среда программирования Code::Blocks и компилятор GCC C++.

Остановимся на геодезической постановке задачи. На рис.1 представлена схема, поясняющая задачу. Пусть требуется вычислить обратный истинный азимут направления, по заданному истинному азимуту направления A–B. Известно также сближение меридианов, проходящих через точку A – γ_A и точку B – γ_B, соответственно, начало и конец линии A–B.

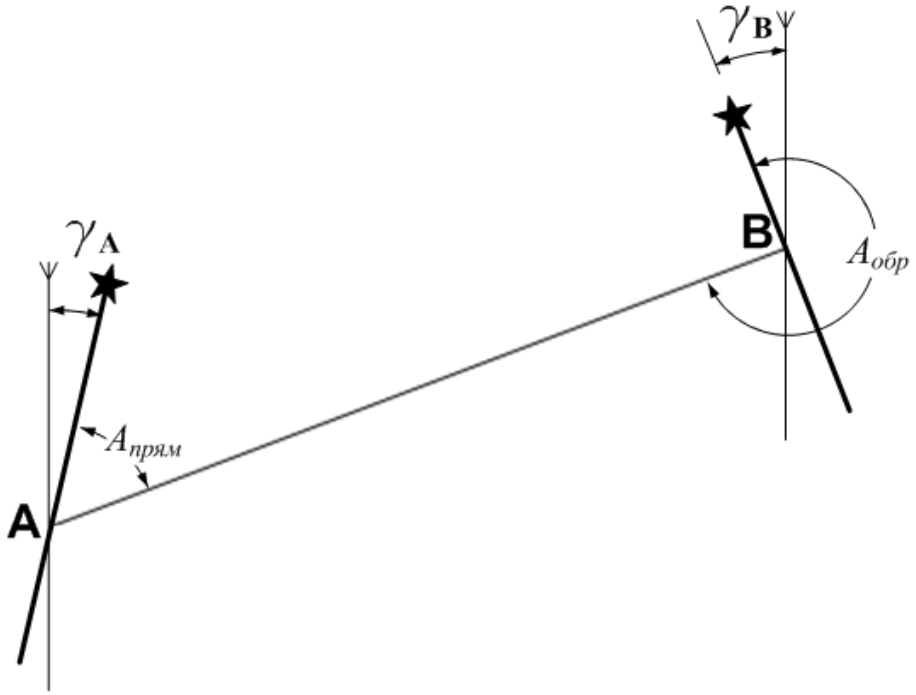


Рис. 1. Схема для определения обратного истинного азимута линии A–B

Как известно [3] для вычисления обратного истинного азимута направления B–A используется следующая формула:

$$A_{обр}^{ист} = A_{прям}^{ист} \pm 180^{o} + \gamma_B - \gamma_A , \quad (1)$$

где $A_{обр}^{ист}$ и $A_{прям}^{ист}$ – обратный и прямой истинный азимут направления.

В приведенной формуле знак «+» используется в случае, если истинный азимут направления находится в диапазоне от 0° до 180°, знак «–» в случае, если истинный азимут в диапазоне от 180° до 360°. Таким образом выбор схемы вычисления в программе удобно реализовать с помощью условной инструкции *if – else*.

Обсуждение результатов

Каждая из учебных программ, используемых в курсе «C++ для картографов и геодезистов» [4], имеет поясняющий текст, предназначенный студентам для самостоятельного изучения того, что делает и как работает компьютерная программа. Ниже представлен код разработанной программы. Рассмотрим код программы (рис.2).

```
01: #include <iostream>
02: using namespace std;
03:
04: int main(void)
05: {
06:   int  degrees, minutes;
07:   double  trueAzimuth,  inversTrueAzimuth;
08:   double convergenceOfMeridiansAtPointA;
09:   double convergenceOfMeridiansAtPointB;
10:
11:   cout<<"Введите истинный азимут направления A-B "
11:       <<"(градусы, пробел, минуты): ";
12:   cin >> degrees >> minutes;
13:   trueAzimuth = degrees * 60 + minutes;
14:   cout<<"Введите сближение меридианов в точке A "
14:       <<"(+/-)градусы, пробел, (+/-)минуты): ";
15:   cin >> degrees >> minutes;
16:   convergenceOfMeridiansAtPointA = degrees * 60 + minutes;
17:   cout<<"Введите сближение меридианов в точке B "
17:       <<"((+/-)градусы, пробел, (+/-)минуты): ";
18:   cin >> degrees >> minutes;
19:   convergenceOfMeridiansAtPointB = degrees * 60 + minutes;
20:
21:   if ((trueAzimuth >= 0) && (trueAzimuth < 180*60))
22:   {
23:    inversTrueAzimuth =  trueAzimuth + 180*60 +
23:                  convergenceOfMeridiansAtPointB –
23:                        convergenceOfMeridiansAtPointA;
24:    if(inversTrueAzimuth>=360*60) inversTrueAzimuth -= 360*60;
25:   }
26:   else
27:   {
28:    inversTrueAzimuth =  trueAzimuth - 180*60  +
28:                  convergenceOfMeridiansAtPointB –
28:                        convergenceOfMeridiansAtPointA;
29:   if(inversTrueAzimuth < 0)    inversTrueAzimuth += 360*60;
30:   }
31:   degrees = (int)inversTrueAzimuth/60;
32:   minutes = (inversTrueAzimuth - degrees*60);
33:
34:  cout <<"Обратный истинный азимут направления B-A: "
34:       << degrees << char(248) <<  minutes <<"'" <<endl;
35:
36:  return 0;
37:  }
```

Рис. 2. Листинг программы для вычисления обратного азимута направления

Далее в тесте приводится справочно-информационный материал для изучаемой программы в том виде, в котором он раздается студентам для самостоятельного изучения. В строках 06 – 09 объявляются переменные. Целочисленные переменные *degrees* и *minutes* используются для градусов и минут прямого и обратного азимута направления. Переменные *trueAzimuth* и *inversTrueAzimuth* для хранения значений прямого и обратного истинных азимутов. Переменные *convergenceOfMeridiansAtPointA* и *convergenceOfMeridiansAtPointB* содержат гауссово сближение меридианов точек А и В. Начальные значения истинного азимута А–В, гауссова сближения меридианов для точки А и точки В вводятся с клавиатуры. В строке 13 вычисляется значения истинного азимута в угловых минутах. В строках 16 и 19 гауссово сближение меридианов также переводится в угловые минуты. Перевод градусов с дробной частью в минуты весьма удобен, поскольку позволяет выполнять арифметические операции (сложение и вычитание угловых величин) с помощью одного оператора (+ или –) и при этом не потерять точность вычисления в минутах. Если пользоваться градусами с дробной частью, то при обратном преобразовании градусов в минуты можно потерять одну минуту. Конечно, используя стандартную математическую функцию округления *round*, ошибку округления можно предотвратить, но использование стандартных функций – тема, изучаемая в курсе программирования после логических операторов и операторов отношения. Поэтому в данной программе функция округления не использовалась.

В строке 21 используется условная конструкция *if–else*, в которой в качестве условия используется сложное логическое условие, состоящее из нескольких выражений. Данное логическое условие находится внутри круглых скобок и представляет собой два выражения отношения соединенных оператором «логическое И». Например, чтобы программа проверяла условие «истинный азимут больше или равен 0°», используется отношение (*trueAzimuth >= 0*) и проверяла условие «истинный азимут меньше 180°» используется отношение (*trueAzimuth < 180*60*). Из этих двух отношений потребуется сконструировать сложное условие, объединяя два простых условия с помощью операции «логическое И» (&&). В результате такая конструкция будет иметь вид: (*trueAzimuth >= 0*)&&(*trueAzimuth < 180*60*). Истинность данного сложного условия будет выполняться, лишь тогда, когда каждая из двух его составных частей является истиной. Если выражение в круглых скобках является истинным, то для вычисления обратного истинного азимута используется формула (1) со сложением +180°.

В строке 24 записана сокращенная форма инструкции *if* и выполняется проверка «истинный обратный азимут больше или равен 360°» (здесь градусы также предварительно переводятся в угловые минуты). Если выражение *inversTrueAzimuth>=360*60* является истинным, то значение обратного истинного азимута уменьшается на 360°. При этом используется «составной оператор присваивания с вычитанием»: *inversTrueAzimuth –= 360*60*. Особенность такого оператора заключается в том, что наряду с присваиванием выполняется и арифметическая операция «вычитание». Такой составной оператор имеет вид (–=). Далее, строки 26 – 30 образуют ветвь *else*, которая определяет инструкции, выполняющиеся в случае, если истинный азимут больше 180°. В этом случае для вычисления обратного истинного азимута направления используется формула (1) с вычитанием –180°.

В строке 29 записана сокращенная форма инструкции *if* и выполняется проверка «истинный обратный азимут меньше 0°». Если выражение *inversTrueAzimuth < 0* является истинным, то значение обратного истинного азимута увеличивается на 360°. При этом используется «составной оператор присваивания со сложением»: *inversTrueAzimuth += 360*60*. В строках 31 – 32 от полученных угловых минут обратного истинного азимута выделяется целая часть градусов и находится значение целых минут. Результаты выводятся на экран. На этом работа программы заканчивается.

Тестирование программы

Выполним тестирование программы. Откомпилировав, и запустив программу, введем исходные значения из первых трех колонок таблицы. В колонке «Обратный истинный азимут линии» представлены результаты расчета программы.

Таблица

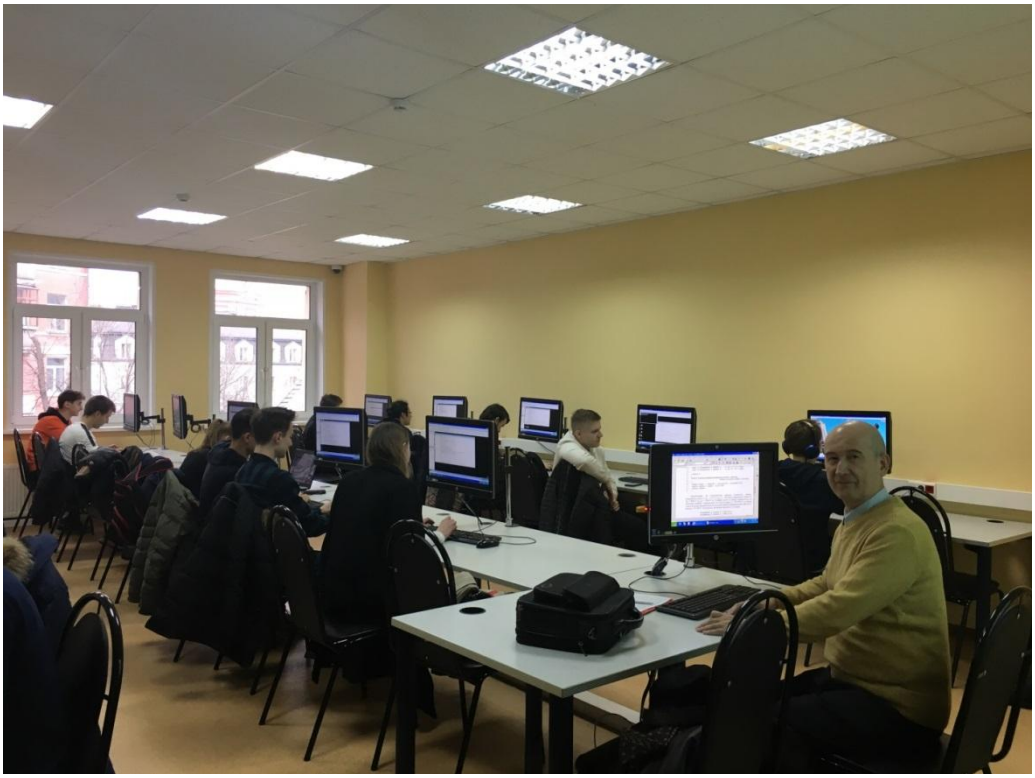
Результаты тестирования программы

Истинный азимут линии	*Зональное сближение меридианов для точки А*	*Зональное сближение меридианов для точки В*	*Обратный истинный азимут линии*
197°15'	+0°27'	–1°20'	15°28'
98°53'	–1°39'	+1°12'	281°44'
179°50'	+1°20'	+2°50'	1°20'

Данные представленные в таблице являются контрольным примером, который должен выполнить студент, запустив рассмотренную выше программу. Взглянув на полученные результаты, преподаватель может быстро определить считает ли правильно программа или нет. Контрольный пример также является важной этапом в продвижении студента по учебным заданиям.

Выводы

Рассмотрены вопросы разработки и применения нового учебного курса «программирование на С++ для студентов картографов и геодезистов» в МИИГАиК. Особенность курса заключается в реализации процесса обучения студентов программированию параллельно с курсом общей геодезии. Разработана и отлажена учебная программа и контрольный пример. Программа вычисляет обратный истинный азимут от прямого истинного азимута, гауссова сближения меридианов для начальной и конечной точки линии направления.



Доцент Заблоцкий В.Р. на практическом занятии по информатике со студентами факультета картографии и геоинформатики МИИГАиК (2019 г.)

Программа может быть использована в качестве учебного материала в курсе С++ программирования для иллюстрации условных инструкций (в сокращенной и полной форме) и сложных логических выражений, составленных из логических операторов и операторов сравнения. Подробно описывается код программы и поясняется назначение использованных инструкций.

Список литературы

1. *Либерти Дж.* Освой самостоятельно С++. 10 минут на урок. М.: ООО «И.Д. Вильямс», 2004. 352 с.
2. *Дейтел Х., Дейтел П.* Как программировать на С++. М.: Изд. «Бином-Пресс», 2008 1386 с.
3. *Киселев М.И., Михелев Д.Ш.* Геодезия: учебник для студ. сред. проф. образования М.: Изд-во Академия, 2008. 384 с.
4. *Заблоцкий В.Р.* Обучение языку С/С++ на основе программирования учебных геодезических задач. Сборник статей по итогам международной научно-технической конференции, посвященной 230-летию основания МИИГАиК, выпуск 2, ч.1, М.: МИИГАиК, 2009, с.199-202.