для культурного досуга, они хуже учатся, чаще болеют из-за недосмотра родителей.

Нередко такая семья испытывает отрицательное отношение со стороны окружающих, особенно соседей, которых раздражают некомфортные условия существования рядом (нарушение спокойствия, тишины, особенно если ребенок-инвалид с задержкой умственного развития или его поведение негативно влияет на здоровье детского окружения). Окружающие часто уклоняются от общения и дети с ОВ практически не имеют возможности полноценных социальных контактов, достаточного круга общения, особенно со здоровыми сверстниками. Общество не всегда правильно понимает проблемы таких семей, и лишь небольшой их процент ощущает поддержку окружающих. В связи с этим родители не берут детей с ОВ в театр, кино, зрелищные мероприятия и т.д., тем самым, обрекая их с рождения на полную изоляцию от общества. В последнее время родители с аналогичными проблемами налаживают между собой контакты.

Семьи выбирают инклюзивные школы для того, чтобы их дети могли расширить свое общение с типично развивающимися сверстниками, а также получить возможность общаться с другими родителями и учителями. Высокая положительная оценка со стороны педагогов школ усилий семьи в воспитании ребенка с ограниченными возможностями служит разработке механизмов эффективного сотрудничества семьи и школы. Для этого, прежде всего, необходим эмоциональный контакт, доверие, уважение и принятие, поддержка родителей и учет их мнения. Педагоги оказывают конкретную практическую помощь семьям, воспитывающим детей с ограниченными возможностями здоровья, при оформлении инвалидности и приобретении специальных средств для обучения детей (тифло- и сурдотехника). Иначе говоря, педагог служит связующим звеном между детьми и взрослыми, обеспечивает атмосферу социально-психологического благоприятствия в образовательном учреждении, привлекает родителей и общественность к организации и проведению социально значимых мероприятий.

Основной целью работы с семьёй является создание благоприятных условий для личностного развития всех членов семьи (физического, социального, духовно-нравственного, интеллектуального), оказание комплексной социально-психологической помощи, а также защита ребенка и его окружения от негативного воздействия различных факторов на личностное развитие.

Кроме того, цель педагогов заключается в том, чтобы не только продуктивно сотрудничать с отдельными семьями, но находить новаторские способы взаимодействия всех семей. В инклюзивных школах учителя разрабатывают механизмы эффективного партнерства школы и семьи, которые связывают семьи и школу и, в конечном счете, помогают всем детям учиться и добиваться успеха. В отличие от многих программ сотрудничества, которые ориентированы на нужды семей со средним и высоким достатком, работники инклюзивных школ принимают во внимание все семьи и используют методы работы, позволяющие семьям с любым доходом участвовать в жизни школы.

Среди таких методов можно отметить:

– поддержку института семьи в школьной системе;

– посредничество между семьями и различными социальными учреждениями;

– предоставление семьям информации для возможности осознанного выбора;

– использование лексики, свободной от профессионального жаргона и понятной родителям;

– помощь с проездом членов семьи к месту проведения школьных мероприятий;

– проведение мероприятий и собраний недалеко от школы и места проживания семей;

– альтернативные варианты работы с ребенком;

– организацию мероприятий, при которой дети не отделены от родителей во время их проведения;

– проведение мероприятий, ориентированных на интересы и нужды родителей;

– встреча с родителями в удобное для них время, например, вечером, на выходных или утром.

Инвалиды являются частью нашего общества. Отношение к ним определяет уровень культуры и социального развития. Равнодушие и жестокость по отношению к инвалидам ведут к духовной деградации всего общества. Гармонично организованная работа педагогов образовательных учреждений с семьями, имеющими детей с ОВЗ – гарантия успеха образовательного и воспитательного процесса, адаптации детей – инвалидов в детском коллективе.

Список литературы

1. Янсон Ульф. Инклюзия как принцип организации образования. – М., 2011.

2. Загуменнов Ю.Л. Инклюзивное образование: создание равных возможностей для всех учащихся // Минская школа сегодня. – 2008. – № 6.

3. Зайцев Д.В. Социальная интеграция детей-инвалидов. – Саратов: Научная книга, 2003.

C++ ДЛЯ СТУДЕНТОВ КАРТОГРАФОВ И ГЕОДЕЗИСТОВ: УЧЕБНАЯ ПРОГРАММА «ОПРЕДЕЛЕНИЕ ВЫСОТЫ ТОЧКИ НА КАРТЕ»

Заблоцкий В.Р.

Московский государственный университет геодезии и картографии, Москва,
e-mail: V.R.Zablotskii@yandex.ru, zablotskii@freemail.ru

Обсуждается учебная программа по программированию на C++ для студентов, обучающихся в геодезическом вузе. Данная программа предназначена для изучения таких понятий языка программирования C/C++ как типы переменных,

объявление и инициализация переменных, использование арифметических операторов. Нашей целью является разработка набора типовых учебных задач с геодезическим содержанием [1–18], которые могут использоваться преподавателями и студентами, обучающимися по специальностям картографии и геодезии в качестве домашних заданий и при выполнении учебного практикума. Задача данной работы заключалась в разработке программы, иллюстрирующей определение высотной отметки по карте, на примере точки, лежащей между горизонталями с разными высотами.

Сделаем замечание, касающееся вопроса наименования переменных. Конечно, удобно, если имя переменной несет смысловую нагрузку, которое раскрывает ее назначение в программе. Например, в разработанной программе используются переменные с именами *verticalInterval*, *contourInterval*, когда речь идет о высоте сечения рельефа горизонталями и о кратчайшем расстоянии между горизонталями. Другими содержательными именами переменных являются следующие: переменная с именем *lowerContourLine*, используется для хранения отметки младшей горизонтали, переменная *greaterContourLine* – для хранения отметки старшей горизонтали. Еще две переменные *distanceToLowerContourLine* и *distanceToGreaterContourLine* предназначены для хранения значения расстояния от младшей горизонтали до выбранной точки и значения расстояния от старшей горизонтали до этой точки.

Рассмотрим геодезическую постановку задачи. Определение высоты точки по карте – часто встречающаяся задача при работе с топографическими картами. Как известно, если точка лежит между двумя горизонталями, то ее отметка определяется интерполированием. Для этого через интересующую точку проводят линию между двумя смежными горизонталями, называемую «заложением» и определяют расстояние от меньшей горизонтали до точки и для контроля расстояние от точки до большей горизонтали. Затем вычисляют высоту точки по формулам:

$$H_1 = H_{\min} + h \cdot \frac{a}{d}; \quad H_2 = H_{\max} - h \cdot \frac{b}{d};$$

$$H_{\text{ср}} = \frac{H_1 + H_2}{2},$$

где $H_{\min}$ и $H_{\max}$ – отметки меньшей и большей горизонталей; h – высота сечения рельефа горизонталями; d – заложение горизонталей; a – расстояние от точки до меньшей горизонтали; b – расстояние от точки до большей горизонтали; $H_{\text{ср}}$ – среднее значение высоты искомой точки.

Программа, представленная ниже, иллюстрирует расчет среднего значения высоты точки в задаче определения отметок точек местности по топографической карте. Пусть, например искомая точка находится между младшей горизонталью *lowerContourLine* со значением 222 м и старшей горизонталью *greaterContourLine* со значением 224 м и пусть далее заложение горизонталей *contourInterval* = 5.0 мм, расстояние от точки до меньшей горизонтали *distanceToLowerContourLine* = 1.0 мм, расстояние от точки до большей горизонтали *distanceToGreaterContourLine* = 4.0 мм.

```
01:  #include <iostream>
02:  using namespace std;
03:
04:  int main (void)
05:  {
06:   int lowerContourLine = 222, greaterContourLine = 224;
07:   int verticalInterval;                        // сечение рельефа, м
08:   float contourInterval = 5.0;                 // заложение, мм
09:   float distanceToLowerContourLine = 1.0;
10:   float distanceToGreaterContourLine = 4.0;
11:
12:   float pointHeight_1, pointHeight_2, pointHeight;
13:
14:   verticalInterval = greaterContourLine - lowerContourLine;
15:
16:   pointHeight_1 = lowerContourLine + verticalInterval*
16:                       distanceToLowerContourLine/contourInterval;
17:   pointHeight_2 = greaterContourLine - verticalInterval*
17:                       distanceToGreaterContourLine/contourInterval;
18:
19:   pointHeight = ( pointHeight_1 +  pointHeight_2 )/2;
20:
21:   cout <<"Высота точки: " << pointHeight <<" м" << endl;
22:
23:   return 0;
24:  }
```

Рассмотрим код программы. В программе используется переменные, как с плавающей точкой, так и целочисленные. В строке 06 переменные типа *int lowerContourLine* и *greaterContourLine* инициализируются значением 222 и 224 соответственно. В строке 07 объявляется переменная *verticalInterval* типа *int* для хранения значения сечения рельефа. В строках 08 переменная *contourInterval* типа *float* инициализируются начальным значением 5,0 мм. Далее в строках 09–10 переменная *distanceToLowerContourLine* инициализируется значением 1,0 мм, а переменная *distanceToGreaterContourLine* значением 4,0 мм. Данные величины определяются на карте с помощью линейки и представляются в миллиметрах возможно с дробной частью, поэтому тип переменных для них определен как *float*. Поскольку значение высоты точки может быть дробным числом, поэтому переменные *pointHeight_1*, *pointHeight_2* и *pointHeight* имеют тип *float*. В строке 14 вычисляется высота сечения рельефа, переменной *verticalInterval* присваивается значение выражения *greaterContourLine – lowerContourLine*. Затем в строке 16 вычисляется высота точки *pointHeight_1* от младшей горизонтали и для контроля в строке 17 высота точки *pointHeight_2* от старшей горизонтали. Выражения в строках 16 и 17 компилятор читает слева направо и поскольку арифметический оператор «сложение» имеет меньший приоритет по сравнению с арифметическими операторами «умножения» и «деления», то вначале выполняется умножение и деление, а потом сложение или вычитание.

Обратим внимание на особенность оператора деления, которая имеется в C/C++. Например, выполнив в программе такую арифметическую операцию как 9/4 мы не получим ожидаемый результат 2.25, вместо этого результатом будет целое число 2. Этот пример иллюстрирует целочисленное деление, такое деление, при котором отбрасывается дробная часть результата. Чтобы получить результат с дробной частью, об этом надо сообщить компилятору. Для этого делимое или делитель (либо и то и другое) должно быть представлено числом с дробной частью, в результате ответ также представляется числом с дробной частью. Если и делимое, и делитель представлено целым числом, то в результате получается целое число. В приведенном выше примере можно использовать запись 9./4, которая «говорит» компилятору, что число 9.0 является числом с плавающей точкой, тогда результатом деления тоже будет число с плавающей точкой. Можно более наглядно указать тот факт, что делимое принадлежит к вещественным числам, записав его в виде 9.0.

Поскольку все арифметические операторы, в том числе «умножить» и «разделить» выполняются над однотипными операндами, то в выражении *verticalInterval*distanceToLowerContourLine* вначале происходит выравнивание типов. Компилятор C++ сначала преобразует операнд к типу большого операнда, в нашем случае к типу *float*, и только потом выполняется арифметическая операция. Таким образом, сначала значение переменной *verticalInterval* преобразуется в тип *float*, затем выполняется умножение, и результат умножения *verticalInterval*distanceToLowerContourLine* дает число с плавающей точкой. Далее выполняется «без проблем» деление, так как и делимое и делитель имеют тип *float*. И наконец, в строке 16 выполняется сложение, причем значение переменной *lowerContourLine* тоже расширяется до типа *float*. Аналогичным образом происходит расширение значения переменных в строке 17. Далее в строке 19 вычисляется высота точки, как среднее значение двух измерений и полученный результат – отметка точки *pointHeight* выводится на экран. Если вы откомпилируете и запустите данную программу, на экране компьютера появится следующий фраза: «Высота точки 222.4 м».

Выводы

Разработана учебная программа на языке C++ для студентов, обучающихся в геодезическом вузе. В программе демонстрируется вычисление средней высоты точки в задаче определения отметки точки местности по топографической карте. Данная программа иллюстрирует структурное программирование на примере вычисления высоты точки, лежащей между двумя разноименными горизонталями.

Список литературы

1. Заблоцкий В.Р. Особенности преподавания информатики в вузе геодезического профиля на современном этапе // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2015. – № 6. – С. 119–125.

2. Заблоцкий В.Р. Программирование на языке C++ для картографов и геодезистов. Учебная программа «Буссоль» с множественным наследованием // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2016. – № 1. – С. 105–107.

3. Заблоцкий В.Р. Программирование на языке C++ для картографов и геодезистов. Учебная программа со структурой «Топографическая карта» // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2016. – № 3. – С. 105–107.

4. Заблоцкий В.Р. C++ для студентов картографов и геодезистов: учебная объектно-ориентированная программа «Перегрузка функции» // Международный журнал экспериментального образования. – 2016. – № 10–1. – С. 20–22.

5. Заблоцкий В.Р. Программирование на языке C++ для картографов и геодезистов: учебная объектно-ориентированная программа «Нивелирная рейка» // Международный журнал экспериментального образования. – 2015. – № 5–1. – С. 89–91.

6. Заблоцкий В.Р. C++ для картографов и геодезистов: учебная объектно-ориентированная программа «Женевская линейка» // Международный журнал экспериментального образования. – 2015. – № 10–1. – С. 25–26.

7. Заблоцкий В.Р. C++ для картографов и геодезистов: учебная программа «Коллимационная погрешность» с условной if-else инструкцией // Международный журнал экспериментального образования. – 2015. – № 12–1. – С. 25–26.

8. Заблоцкий В.Р. C++ для картографов и геодезистов: учебная программа «Преобразование угла из радианной меры в градусную» с инструкцией цикла // Международный журнал экспериментального образования. – 2016. – № 6–1. – С. 25–27.

9. Заблоцкий В.Р. С++ для картографов и геодезистов: учебная программа «Уклон ската», иллюстрирующая инструкцию цикла // Международный журнал экспериментального образования. – 2016. – № 4–3. – С. 462–463.

10. Заблоцкий В.Р., Зеленков В.В. Учебная компьютерная программа «ТЕОДОЛИТ». Часть 1. Вычисление горизонтальных углов // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2009. – № 4. – С. 90–100.

11. Заблоцкий В.Р., Фам Суан Хоан. Учебная компьютерная программа «ТЕОДОЛИТ». Часть 2. Использование указателей для создания журнала угловых измерений // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2011. – № 5. – С. 107–113.

12. 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, вып. 2, ч. 1. – М.: МИИГАиК, 2009. – С. 199–202.

13. 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.

14. Заблоцкий В.Р., Васякин С.А. Применение программы «Калькулятор» в решении учебных геодезических задач // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2004. – № 5. – С. 10–34.

15. Заблоцкий В.Р. Базовые понятия языка С/С++. Методическое пособие по курсу «Информатика и программно-алгоритмические языки». – М.: Изд-во МИИГАиК, 2004. – 56 с.

16. Лениченко В.Н., Заблоцкий В.Р. Программирование на С и С++: учебное пособие. – М.: Изд-во МИИГАиК, 2009. – 31 с.

17. Заблоцкий В.Р., Шевчук Д.А. Методические указания по курсу «Информатика» (для самостоятельного изучения программы Калькулятор). – М.: Изд-во МИИГАиК, 2003. – 27 с.

18. Журкин И.Г., Заблоцкий В.Р., Степанов С.А. Компьютерное тестирование студентов первого курса по дисциплине «Информатика и программно-алгоритмические языки» // Известия высших учебных заведений. Геодезия и аэрофотосъемка. – 2006. – № 4. – С. 167–185.

СИСТЕМА ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В КОЛЛЕДЖЕ

Лаптева Е.А., Чеканин И.М.

ФГБОУ ВО «Волгоградский государственный медицинский университет», Волгоград, e-mail: elenadanik@yandex.ru

Внедрение Федеральных государственных образовательных стандартов выдвигает новые требования к организации образовательного процесса, в том числе и к объективным системам оценки уровня знаний и умений студентов, компетенций выпускников на основе чётких согласованных критериев. Оценка качества подготовки студентов и выпускников должна включать результаты их текущей, промежуточной и итоговой государственной аттестации. Однако образовательные стандарты не содержат дидактического инструментария, позволяющего измерить и оценить результаты обучения.

Обоснование и выбор критериев, измерение и оценка уровней сформированности приобретённых выпускниками компетенций представляют собой сложную задачу в профессиональном образовании. Известно, что фонды оценочных средств должны разрабатываться, утверждаться и применяться в соответствии со следующими представлениями о ключевых ценностях оценивания. Оценивание должно быть: валидным (объекты оценки должны соответствовать поставленным целям учебной дисциплины); надёжным (необходимо использовать единообразные согласованные критерии или стандарты); справедливым (разные студенты должны иметь равные возможности добиться успеха); развивающим (фиксировать, что могут студенты и как им улучшить свои результаты); эффективным (выполнимым, не забирать много времени у преподавателей и студентов).

Сегодня функция оценивания не сводится только к выявлению недостатков освоения учебных программ, а рассматривается главным образом как средство критического анализа образовательного процесса, предполагающего определение направлений его улучшения.

Современные требования к результатам освоения программ подготовки специалистов среднего звена предусматривают обязательное внедрение компетентностного подхода, центральным понятием которого является компетенция. В отличие от знаний, умений, навыков, предполагающих действие по аналогии с образцом, компетенция предусматривает наличие опыта самостоятельной деятельности на основе универсальных знаний.

Для того чтобы установить уровень компетентности выпускника медицинского колледжа, необходимо обозначить средства оценивания, которые должны выявлять как содержательный, так и деятельностный компонент подготовленности студента. В условиях компетентностного подхода это предполагает наличие необходимого набора объективных и качественных критериев. При этом методы оценки должны быть направлены на измерение освоенных целостных компетенций, а не отдельных знаний и умений. Можно выделить основные методы оценки, используемые при модульном обучении:

– составление портфолио практического обучения студента, в структуру портфолио могут входить: результаты защиты курсовых и научно-исследовательских работ; участие в выставках, конкурсах; результаты обучения на курсах; результаты практики; сведения об участии в олимпиадах, конкурсах, конференциях, учебных семинарах; спортивные и иные достижения;

– сбор образцов деятельности обучающихся, демонстрирующих освоение ими требуемых компетенций;

– индивидуальные и групповые проекты;

– обязательная публичная защита курсовых работ;

– применение в учебном процессе ситуационных и практикоориентированных заданий;

– использование кейс-измерителей, при разработке которых целесообразно предлагать не только подробное описание ситуации, но и перечень вопросов;