

деятельности технического заказчика. И это шаг 2 на пути к достижению совершенства в области организации строительства.

Штат строительной организации вряд ли будет органичен одним человеком. Технический заказчик – это тоже не один человек, а целая компания людей, взаимодействующих между собой, обменивающихся информацией для достижения определенной цели. Поэтому шаг 3: реализация менеджмента корпоративного знания. Менеджмент корпоративных знаний является важной частью системной поддержки процессов управления организациями и отраслями. В рамках этого направления особое значение имеет создание так называемой Онтологии организации. Она представляет собой модель понятий и взаимосвязей между ними, которая играет роль интеллектуально развитого тезауруса ПрО, обеспечивая:

- универсальность постановок задач;
- гармонизованность различных компонент средств автоматизации при заимствовании и наследовании информации по ходу аналитических процедур;
- эффективную коммуникацию между представителями разных профессиональных и ведомственных групп;
- возможность интеллектуального мониторинга положения дел в организации, базирующегося на комплексных моделях, включающих объекты, принадлежащие разным функциональным областям и рассматриваемые одновременно в разных функциональных ракурсах [3].

«Нет предела совершенству», и так можно до бесконечности делать шаги к достижению максимума эффективности деятельности технического заказчика. Но если претворить в жизнь названные 3 шага, то это уже будет неплохое начало на «пути к звездам».

Список литературы

1. Градостроительный кодекс Российской Федерации, Пункт 22 ст. 1. [Электронный ресурс] – Режим доступа: <http://www.gradkod.ru>.
2. Петренко Л.К., Манжиловская С.Е. Организация работ и управление реконструкцией [Электронный ресурс] // Интернет-журнал «Науковедение». – 2013. – № 3. [Электронный ресурс] – Режим доступа: <http://naukovedenie.ru/PDF/13trgsu313.pdf>.
3. Гаврилова Т. Онтологический подход к управлению знаниями при разработке корпоративных систем автоматизации [Электронный ресурс]. – Режим доступа: http://bigc.ru/theory/km/ontol_podhod_to_uz.php.
4. <http://ru.wikipedia.org/wiki/Инженерия>.

С++ ДЛЯ КАРТОГРАФОВ И ГЕОДЕЗИСТОВ. УЧЕБНАЯ ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ ПРОГРАММА «ЖЕНЕВСКАЯ ЛИНЕЙКА»

Заблоцкий В.Р.

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

Учебный практикум по программированию для студентов картографов и геодезистов обычно основывается на решении математи-

ческих задач, например, найти сумму сходящегося ряда с заданной точностью или вычислить площадь подынтегральной функции и не учитывает специфику геодезического направления подготовки. Нашей целью является разработка набора типовых учебных задач геодезического содержания [1, 2], которые могут использоваться в качестве домашних заданий и при выполнении лабораторных работ студентами геодезического направления подготовки.

В данной работе обсуждается объектно-ориентированная программа на С++, иллюстрирующая наследование. Наследование является фундаментальной концепцией объектно-ориентированного программирования и рассматривается на примере создания и использования обычной и женеvской линейек для измерения длин отрезков на аэрофотоснимке. Известно, что с помощью обычной линейки длина отрезка определяется с погрешностью 0,2 мм, женеvская линейка с лупой позволяет определять длину с погрешностью 0,05 мм. Тем не менее, линейки, как объекты программирования, близки по своему назначению, что позволяет использовать их в качестве примера на простое наследование. Класс женеvская линейка реализуется на основе родительского класса обычная линейка и наследует методы и свойства родительского класса. В программе моделируется техника измерения длин двух отрезков АВ и АС на аэрофотоснимке с помощью обычной и женеvской линейек.

Рассмотрим код программы. Определен класс линейка *Ruler*, содержащий общие методы: «провести отрезок прямой» *DrawAStraightLine* и «измерить длину отрезка» *MeasuringLineLenght*. Порождая класс женеvская линейка *GenevaRuler* из существующего класса *Ruler*, мы добавляем метод «измерить длину отрезка, используя лупу» *MeasuringLineLenghtUsingAMagnifyingGlass*. Определение методов двух классов приводится в строках 23–26, 27–35 и в строках 36–50 соответственно. В главной функции заданы три точки А, В и С на аэрофотоснимке, далее в строке 55 создается объект обычная линейка *ordinaryRuler*, а в строке 56 – объект женеvская линейка *myGenevaRuler*. Методом обычной линейки *ordinaryRuler.DrawAStraightLine* между точками А и В проводится отрезок (строка 58) и его длина измеряется методом *ordinaryRuler.MeasuringLineLenght*. Аналогичные действия выполняются с помощью женеvской линейки, в строке 63 проводится отрезок между точками А и С, затем измеряется длина отрезка. Результаты измерений выводятся на экран в консольное окно. На этом выполнение программы заканчивается.

```

01: #include <iostream>
02: using namespace std;
03:
04: class Ruler {
05:     protected:
06:         float accuracyOfLineMeasurement;
07:     public:
08:         Ruler(float accuracy){
09:             accuracyOfLineMeasurement = accuracy;
10:         }
11:         void DrawAStraightLine(char pointOne,char pointTwo);
12:         float MeasuringLineLenght(char pointOne,char pointTwo);
13: };
14:
15: class GenevaRuler : public Ruler {
16:     public:
17:         GenevaRuler(float accuracy):Ruler(accuracy){
18:             accuracyOfLineMeasurement = accuracy;
19:         }
20:         float MeasuringLineLenghtUsingAMagnifyingGlass(char
20:             pointOne,char pointTwo);
21: };
22:
23: void Ruler::DrawAStraightLine(char pointOne,char pointTwo)
24: {
25:     cout<<"Линия "<<pointOne<<pointTwo<<"проведена на аэрофотоснимке"
26: }
27: float Ruler::MeasuringLineLenght(char pointOne,char pointTwo)
28: {
29:     float lenght;
30:     cout <<"Определите длину линии " << pointOne << pointTwo
31:         <<" по обычной линейке и введите это значение:";
32:     cin >> lenght;
33:
34:     return lenght;
35: }
36: float GenevaRuler::MeasuringLineLenghtUsingAMagnifyingGlass(char
36:     pointOne, char pointTwo)
37: {
38:     float x1,x2,lenght;
39:     cout <<"Определите с помощью лупы отсчет для точки " << pointOne
40:         <<" и введите это значение:"; cin >> x1;
41:     cout <<"Определите с помощью лупы отсчет для точки " << pointTwo
42:         <<" и введите это значение:"; cin >> x2;
43:
44:     if (x2 > x1)
45:         lenght = x2 - x1;
46:     else
47:         lenght = x1 - x2;
48:
49:     return lenght;
50: }
51: int main()
52: {
53:     char point[3] = {'A','B','C'}; //Даны три точки на аэрофотоснимке
54:
55:     Ruler ordinaryRuler(0.2);
56:     GenevaRuler myGenevaRuler(0.05);
57:
58:     ordinaryRuler.DrawAStraightLine(point[0],point[1]);
59:
60:     cout <<"Длина линии " << point[0] << point[1] <<" равна: "
61:         << ordinaryRuler.MeasuringLineLenght(point[0],point[1])
62:
63:     myGenevaRuler.DrawAStraightLine(point[0],point[2]);
64:
65:     cout <<"\nДлина линии " << point[0] << point[2] <<" равна: "
66:         << myGenevaRuler.MeasuringLineLenghtUsingAMagnifyingGlass
66:         (point[0],point[2]);
67:
68:     return 0;
69: }

```

Одно из преимуществ наследования состоит в том, что исключается дублирование программного кода внутри программы. Это видно на примере строки 63, где объект женеvская линейка вызывает метод *DrawAStraightLine*, определенный в родительском классе.

Выводы

Разработана учебная объектно-ориентированная программа на языке C++ для студентов, обучающихся в вузе геодезического профиля. В программе реализованы два класса – обычная и женеvская линейка на основе наследования, и моделируется техника измерения длин

отрезков с помощью созданных линейек. Данная программа подчеркивает преимущества объектно-ориентированного программирования на основе задачи определения длин отрезков на аэрофотоснимке.

Список литературы

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