УДК 622.272:624.191.5

С.А. ХАРИН (д-р техн. наук, доц.)
Институт предпринимательства "Стратегия"

ПРИМЕНЕНИЕ ПАРАМЕТРИЗИРОВАННЫХ МЕТОДОВ РАЗРАБОТКИ ПРОГРАММ В ГОРНОМ ДЕЛЕ

Для автоматизации задач организации в горном деле представляет интерес разработка методов исследований, соответствующего программного обеспечения, которые позволили бы служить в качестве инструментов изучения вопросов шахтного строительства. Для проведения исследований разработан и проведен анализ ряда компьютерных программ на языке Java в интегрированной среде разработки модульных кроссплатформенных приложений Eclipse, использующих параметризированные методы.

Ключевые слова: метод, объект, класс, параметры, скорость, строительство, горные выработки

Проблема и ее связь с научными и практическими задачами. Динамичное функционирование горного производства является условием эффективного развития экономики. Проблема реконструкции производственных мощностей в условиях больших глубин разработки должна сопровождаться интенсивными усилиями в направлении исследований, направленных на совершенствование всех технологических процессов, которые требуют соответствующей автоматизации для обеспечения достоверности результатов.

Анализ исследований и публикаций. Анализ ранее опубликованных результатов исследований и современного состояния практики проектирования и строительства горных выработок указывает на необходимость более широкого использования компьютерных технологий для детального учета различных особенностей сооружения подземных объектов и обеспечения оптимальных параметров ведения работ.

Постановка задачи. Выполнить анализ различных видов программ, использующих параметризированные методы и конструкторы и разработать эффективное программное обеспечение для исследования вопросов организации проходки горных выработок

Изложение материала и результаты. В современных условиях является актуальной разработка методов исследований, соответствующего программного обеспечения, которые позволили бы служить в качестве инструментов изучения вопросов организации строительства. Рассмотрим решение задачи определения скорости строительства горных выработок с помощью следующих программ, написанных на языке Java (табл. 1, рис. 1-5).

Таблица 1 – Программы, используемые для расчета параметров организации проходки выработок

№	Вид программы
1	Программа, включающая метод внутри класса
2	Программа, включающая метод, возвращающий скорость
3	Программа, которая использует параметризованный метод
4	Класс использует параметризованный конструктор
5	Конструктор для разрешения конфликтов пространства имен

Программа, включающая метод внутри класса (рис. 1), характеризуется достаточной простотой. Рассмотрим две следующие строки программы: myboxl.volume () и mybox2.volume(). Первая строка включает метод volume () объекта myboxl. Таким образом, обращение к myboxl.volume () отображает скорость про-ходки выработки,

определенной переменной myboxl, а обращение к mybox2.volume ()отображает скорость проходки выработки, определенной переменной mybox2.

Когда выполняется myboxl.volume (), исполняющая система Java передает управление коду, определенному внутри метода volume ().После того как операторы внутри volume () выполнятся, управление возвращается в вызывающую подпрограмму, и работа продолжается со строки кода, следующей за вызовом. Используемый нами метод – это способ реализации подпрограмм в языке Java.

Другой способ реализации метода volume () состоит в том, чтобы он вычислял скорость проходки выработки и возвращал результат вызывающей программе (рис. 2). Рассмотрим строку vol = myboxl.volume (). При вызове метода volume () он помещается справа от оператора присваивания. Слева находится переменная vol, которая примет значение, возвращенное методом volume ().

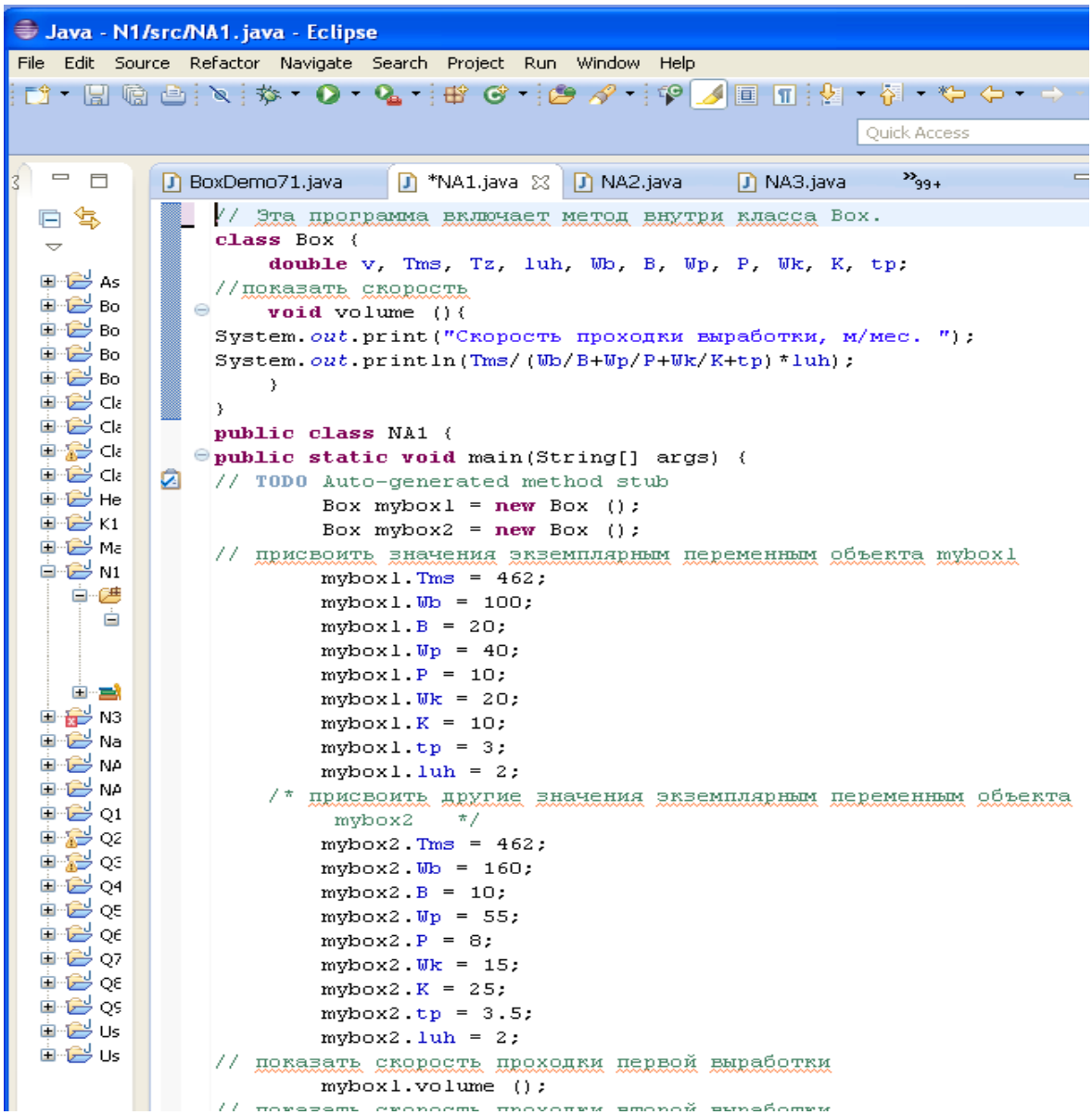

Рис. 1. Программа, включающая метод внутри класса

При использовании метода с параметрами происходит обобщение метода. Параметризованный метод может работать на множестве данных. Весьма целесо-

образно при определении скорости проходки выработок создать метод, который берет измерение скорости в свои параметры и устанавливает каждую переменную экземпляра. Эта концепция реализована следующей программой рис. 3. Очевидно, метод setDim () используется, чтобы установить скорость проходки каждой выработки.

При создании экземпляров иногда сложно инициализировать все переменные в классе. Поскольку требования инициализации являются достаточно общими, Java разрешает инициализацию объектов в момент их создания. Эта автоматическая инициализация выполняется с помощью конструктора. Конструктор инициализирует объект после его создания. Параметризованные конструкторы позволяют создавать объекты с разными данными. Следующая версия объекта (рис. 4) определяет конструктор, который устанавливает скорость проходки выработки с помощью своих параметров.

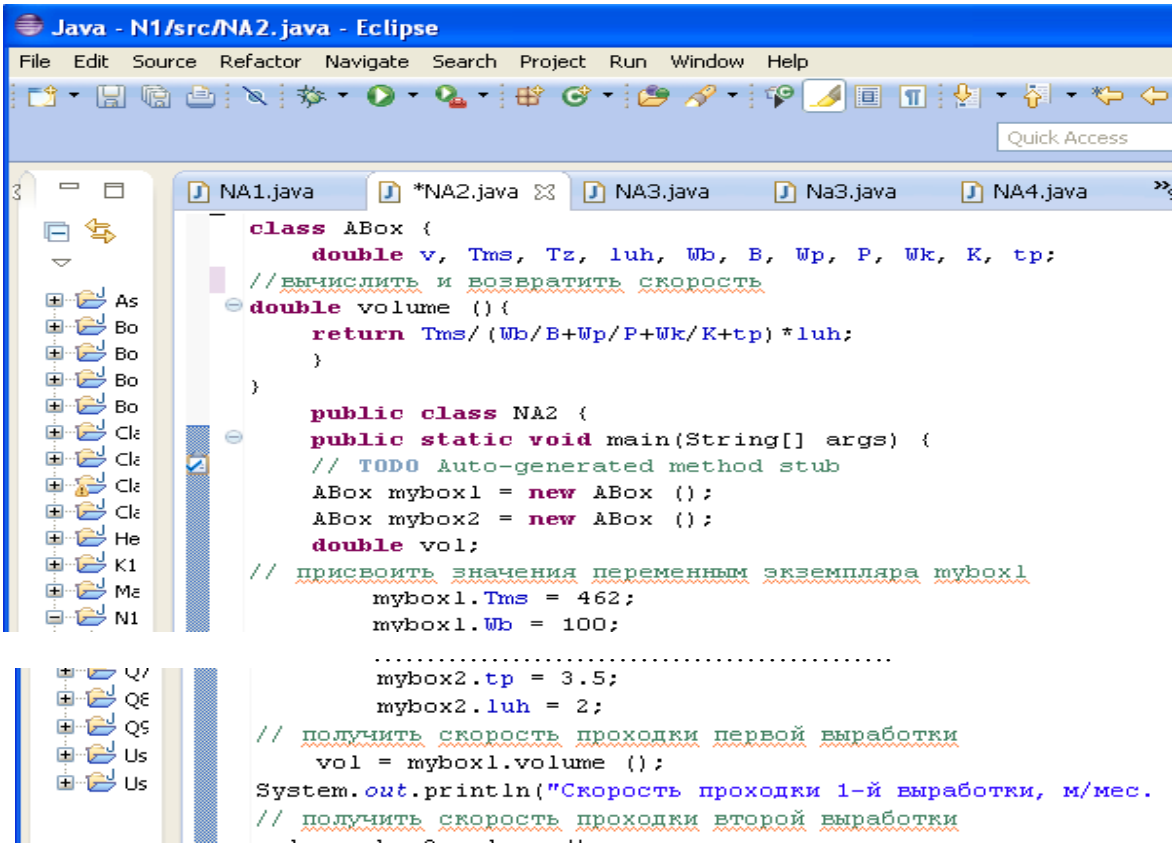

Рис. 2. Программа, включающая метод, возвращающий скорость

В ряде случаев у метода возникает необходимость обращаться к объекту, который его вызвал. Для этого Java определяет ключевое слово this.

Его можно использовать внутри любого метода, чтобы сослаться на текущий объект. То есть this - это ссылка на объект, метод которого был вызван. Как известно, в Java недопустимо объявление двух локальных переменных с одним и тем же именем внутри той же самой или включающей области действия идентификаторов. Когда локальная переменная имеет такое же имя, как переменная экземпляра, локальная переменная скрывает переменную экземпляра.

Поскольку this позволяет обращаться прямо к объекту, это можно применять для разрешения любых конфликтов пространства имен, которые могли бы происходить между экземплярными и локальными переменными. Ниже представлена

другая версия программы расчета скорости проходки выработки (рис. 5), которая использует, например, Tms и luh для имен параметров и затем применяет this, чтобы получить доступ к переменным экземпляра, с теми же самыми именами, например, this.Tms = Tms или this.luh = luh.

Анализ сопоставимых по выполняемым задачам фрагментов программ расчета скоростей проходки выработок по числу знаков отражается рис. 6. Очевидно, что наименьшим их числом характеризуется 5 вариант программы, представленный конструктором с ключевым словом this. Вместе с тем такие программы подвержены опасности совершения программистом ошибки во время инициализации переменных. По этому фактору выглядит приемлемой весьма простая программа версии 1, включающая метод внутри класса, которая не слишком сильно превышает по числу знаков версию 5.

```
// Эта программа использует параметризованный метод.
class Box {
double Tms, luh, Wb, B, Wp, P, Wk, K, tp;
// вычислить и возвратить скорость
double volume() {
return (Tms /(Wb/B+Wp/P+Wk/K+tp))*luh;
}
// установить параметры
void setDim(double aTms, double aluh, double aWb, double aB, doubl
    Tms = aTms;
    luh  = aluh;
    Wb = aWb;
    B = aB;
    Wp = aWp;
    P = aP;
    Wk = aWk;
     K = aK;
     tp = atp;
 }
}
public class Na3 {
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Box mybox1 = new Box();
    Box mybox2 = new Box();
    double vol;
    // инициализировать каждую выработку
    mybox1.setDim(462, 2, 100, 20, 40,   10, 20, 10, 3);
    mybox2.setDim(462, 2, 160, 10, 55,   8,   25, 15, 3.5);
    // получить скорость 1-й выработки
    vol = mybox1.volume();
    System.out.println("Скорость проходки 1-й выработки, м/с " +
    // получить скорость 2-й выработки
```

Рис. 3. Программа использует параметризованный метод

```
/* класс Box использует параметризованный конструктор для инициали
class Box {
double Tms, luh, Wb, B, Wp, P, Wk, K, tp;
// конструктор класса Box.
Box(double aTms, double aluh, double aWb, double aB, double aWp,
Tms = aTms;
luh  = aluh;
Wb = aWb;
B = aB;
Wp = aWp;
P = aP;
Wk = aWk;
K = aK;
tp = atp;
}

// вычислить и возвратить скорость
double volume() {
return (Tms /(Wb/B+Wp/P+Wk/K+tp))*luh;
}
}
public class NA4 {
public static void main(String[] args) {
// TODO Auto-generated method stub
// объявить, распределить и инициализировать Box-объекты
    Box mybox1 = new Box(462, 2, 100, 20, 40,   10, 20, 10, 3);
    Box mybox2 = new Box(462, 2, 160, 10, 55,   8,   25, 15, 3.5).
    double vol;

// получить скорость 1-й выработки
vol = mybox1.volume ();
System.out.println("Скорость 1-й выработки, м/с " + vol);
```

Рис. 4. Класс Box использует параметризованный конструктор

```
class Box {
double Tms, luh, Wb, B, Wp, P, Wk, K, tp;

// конструктор класса Box.
Box(double Tms, double luh, double Wb, double B, double Wp, doubl
this.Tms = Tms;
this.luh  = luh;
this.Wb = Wb;
this.B = B;
this.Wp = Wp;
this.P = P;
this.Wk = Wk;
this.K = K;
this.tp = tp;
}

// вычислить и возвратить скорость
double volume() {
return (Tms /(Wb/B+Wp/P+Wk/K+tp))*luh;
}
}
public class NA5 {
public static void main(String[] args) {
// TODO Auto-generated method stub

Box mybox1 = new Box(462, 2, 100, 20, 40, 10, 20, 10, 3);
Box mybox2 = new Box(462, 2, 160, 10, 55, 8, 25, 15, 3.5);
double vol;

// получить скорость 1-й выработки
vol = mybox1.volume ();
System.out.println("Скорость 1-й выработки, м/с " + vol);
```

Рис. 5. Конструктор для разрешения конфликтов пространства имен

```
class Box {
double Tms, luh, Wb, B, Wp, P, Wk, K, tp;

// конструктор класса Box.
Box(double Tms, double luh, double Wb, double B, double Wp, doubl
this.Tms = Tms;
this.luh  = luh;
this.Wb = Wb;
this.B = B;
this.Wp = Wp;
this.P = P;
this.Wk = Wk;
this.K = K;
this.tp = tp;
}

// вычислить и возвратить скорость
double volume() {
return (Tms /(Wb/B+Wp/P+Wk/K+tp))*luh;
}
}
public class NA5 {
public static void main(String[] args) {
// TODO Auto-generated method stub

Box mybox1 = new Box(462, 2, 100, 20, 40, 10, 20, 10, 3);
Box mybox2 = new Box(462, 2, 160, 10, 55, 8, 25, 15, 3.5);
double vol;

// получить скорость 1-й выработки
vol = mybox1.volume ();
System.out.println("Скорость 1-й выработки, м/с " + vol);
```

Рис. 6. Анализ фрагментов программ по числу знаков

Выводы и направления дальнейших исследований

В ходе исследования разработан и проведен анализ ряда компьютерных программ для автоматизации вопросов изучения организации шахтного строительства на языке Java в интегрированной среде Eclipse, использующих параметризированные методы.

Надійшла до редакції 31.03.2014

С.А. Харін

ЗАСТОСУВАННЯ ПАРАМЕТРИЗОВАНИХ МЕТОДІВ РОЗРОБКИ ПРОГРАМ У ГІРНИЧІЙ СПРАВІ

Для автоматизації завдань організації в гірничій справі становить інтерес розробка методів досліджень, відповідного програмного забезпечення, які дозволили б служити в якості інструментів вивчення питань шахтного будівництва. Для проведення досліджень розроблено та проведено аналіз ряду комп'ютерних програм мовою Java в інтегрованому середовищі розробки модульних кроссплатформенних додатків Eclipse, що використовують параметризовані методи.

Ключові слова: метод, об'єкт, клас, параметри, швидкість, будівництво, гірничі виробки

S.A. Kharin

APPLICATION DEVELOPMENT PARAMETRIZED METHOD PROGRAMS IN MINING

To automate the tasks of the organization in the mining of interest to develop research methods appropriate software that would serve as tools for studying questions of mine construction. For research designed and analyzed a number of computer programs in Java integrated development environment modular cross-platform applications, Eclipse, use parameterized methods.

Keywords: method, object, class, parameters, speed, construction, mine workings