

## ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРОЕКТИРОВАНИЯ ГОРНОТЕХНИЧЕСКИХ СИСТЕМ

DOI: 10.26730/1999-4125-2020-5-47-54

УДК 528:634.958

### ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ, ИСПОЛЬЗУЮЩЕЙ DLL БИБЛИОТЕКИ УЧЕТА ПАРАМЕТРОВ РЕЖИМА ОПТИМИЗАЦИИ ПРОЦЕССОВ ГОРНЫХ РАБОТ

### INFORMATION SUPPORT OF A SYSTEM USING DLL LIBRARIES FOR RECORDING MINING OPTIMIZATION MODE PARAMETERS

**Шогенова Залина Асланбековна**,  
старший преподаватель, e-mail: shogenova.88@mail.ru  
**Shogenova Zalina Aslanbekovna**, Senior Lecturer,  
**Крымшокалова Джульетта Аbugалиевна**,  
кандидат физико-математических наук,  
старший преподаватель, e-mail: dzhuletta.krym@bk.ru  
**Krymshokalova Dzhuletta Abugalievna**, C. Sc. in Physics and Mathematics, Senior Lecturer  
**Шереужева Милана Артуровна**,  
студентка 3 курса, e-mail: milana.shereuzheva2001@mail.ru  
**Shereuzheva Milana Arturovna**, student

Кабардино-Балкарский государственный университет им. Х.М. Бербекова, 360004, Россия, г.  
Нальчик, ул. Чернышевского, 173  
Kabardino-Balkar State University. H.M. Berbekova, 360004, Russia, Nalchik, Chernyshevskogo, 173

#### **Аннотация:**

*Статья посвящена использованию DLL библиотек для оптимизации процессов горных работ. На современном этапе становления рыночной экономики страны основой функционирования и развития ее горнодобывающей промышленности является открытая форма добычи. Сегодня около 90% железной руды, до 60% руд цветных металлов и угля производится в России с использованием этого метода. Открытый карьер дает значительно лучшие технико-экономические показатели, чем подземный рудник.*

*Бурное развитие горных работ стало возможным благодаря достижениям горной науки и техники, в основу которых положены труды академиков Н. В. Мельникова, В. В. Ржевского, профессоров Е. Ф. Шешко, А. И. Арсентьева, В. С. Хохрякова, П. И. Токмакова и др.*

*Специфика горных работ заключается в том, что в подготовке информационного сопровождения производственных процессов, из которых складывается добыча полезных ископаемых, должны быть использованы весьма разнородные модели и алгоритмы их обработки. В двух крайних случаях это приводит либо к очень большому количеству узкоспециализированных программных продуктов и к необходимости иметь связующие программные компоненты; либо, другой вариант – суперсистема, охватывающая все информационные аспекты производственной жизни предприятия. В чистом виде эти варианты не найдены, первый из-за того, что даже программы, изначально ограниченные функционально, развивают и расширяют свои области действия до смежных областей, второй – из-за чрезвычайной сложности проектирования, настройки и обслуживания этой системы.*

**Ключевые слова:** информационное обеспечение, DLL библиотека, горные работы.

#### **Abstract:**

*The article is devoted to the use of DLL libraries to optimize mining processes. At the present stage of the formation of the country's market economy, the basis for the functioning and development of its mining industry is the open form of mining. Today, about 90% of iron ore, up to 60% of non-ferrous metal ores and coal are produced*

*in Russia using this method. An open pit gives significantly better technical and economic indicators than an underground mine.*

*The rapid development of mining became possible thanks to the achievements of mining science and technology, which are based on the works of Academicians N.V. Melnikov, V.V. Rzhovsky, professors E.F. Sheshko, A.I. Arsenyev, V.S. Khokhryakov, P. I. Tokmakova, and others.*

*The specificity of mining operations lies in the fact that in the preparation of information support for production processes that make up the extraction of minerals, very heterogeneous models and algorithms for their processing should be used. In two extreme cases, this leads either to a very large number of highly specialized software products and to the need to have middleware components. Or, another extreme option is a supersystem that covers all information aspects of the production life of an enterprise.*

**Key words:** *information support, DLL library, mining operations.*

Основным направлением технического перевооружения горных работ за рубежом в последнее десятилетие стало широкое распространение высокопроизводительного оборудования буровых станков с поперечником долота до 450 мм, трудовых экскаваторов с ковшем вместимостью до 26 м<sup>3</sup>, автосамосвалов грузоподъемностью до 310 м<sup>3</sup>, разного вспомогательного оборудования [1].

Задачами автоматизации являются обеспечение высокой эффективности производства, безопасности работы за счет автоматического контроля основных параметров оборудования и опасных факторов (газа, отсутствие проветривания), предупреждение опасных режимов работы оборудования и персонала, автоматические сигнализация, блокировки, регулирование и управление.

Эффективность внедрения автоматизации зависит от поточности операций в технологии производства, наличия комплексной механизации, правильной эксплуатации аппаратуры и других компонентов. При выборе технологии, способа каждого процесса необходимо проанализировать исходные данные: состояние и свойства горных работ, характеристики их разработки, условие залегания месторождения, гидрогеологические и климатические условия и производительность карьера и т.п.

Основная задача состоит в умелом использовании знаний для решения самостоятельных реальных технологических задач, возникающих на производстве. В процессе работы имеется возможность расширения знаний путем изучения передового опыта горных предприятий и литературных источников.

Ведущими производственными процессами горных работ являются подготовка горных пород к выемке, выемочно-погрузочные работы, перемещение горной массы, отвалообразование вскрышных пород, складирование добытого полезного ископаемого. Правильный выбор технологии, способа процессов горных работ и горнотранспортного оборудования во многом определяет высокую производительность и эффективность разработки месторождения [2].

Информационные технологии предназначены для решения различных типов проблем, возникающих на любом этапе добычи, в первую очередь для информационных услуг для всех работников предприятий, связанных с принятием управленческих решений. Здесь информация обычно представляется в форме регулярных или специальных управленческих отчетов и содержит информацию о прошлом, настоящем и возможном будущем компании. Автоматизация офиса включает в себя организацию и поддержку коммуникационных процессов как внутри производства, так и во внешней среде на основе компьютерных сетей и других современных средств передачи и работы с информацией. Широко используются информационные технологии при проектировании в виде компьютерной графики, моделирования процессов и инженерных расчетов [3].

В горном деле современная система использования информационных технологий представляет собой комплекс со следующими основными подсистемами обеспечения [4]:

– информационное обеспечение – система классификации информации, технологическая схема обработки данных, нормативно-справочная информация, система документооборота, создание различного вида документации;

– организационное обеспечение – совокупность мер и мероприятий, регламентирующих функционирование системы управления, наличие связи между структурами предприятия;

– техническое обеспечение – комплекс используемых в системе технических средств, включающий ЭВМ и средства связи;

– математическое обеспечение – совокупность методов, правил, математических моделей и алгоритмов решения задач;

– программное обеспечение – совокупность программ, необходимых на всех этапах деятельности предприятия.

В настоящее время существует множество программных продуктов, обеспечивающих информационные технологии обработки

Таблица. Значения параметра модуля инициализации библиотеки

Dll_process_attach	Указывает, что библиотека подключена к вызывающему
Dll_process_detach	Указывает, что библиотека отсоединена от вызывающего процесса
Dll_thread_attach	Указывает, что процесс создает новый поток
Dll_thread_detach	Указывает, что процесс завершает поток

различного рода информации [5].

Важнейшей особенностью горного производства является его естественная, природная база. Разработка месторождений полезных ископаемых, их добыча всегда требовали использования совокупности многих технических приемов, которые прошли многовековой путь развития от горного искусства до современной технологии горного производства.

Природная база каждого месторождения характеризуется совокупностью признаков, называемых горно-геологическими условиями. Эти условия определяют специфику горной технологии и, поскольку они изменяются в очень широком диапазоне, существует большое многообразие типов горнотехнологических процессов.

Таким образом, при разработке месторождения имеют дело с вероятностным характером горно-геологических факторов, и это обстоятельство определяет вероятностный характер протекания всего производственного процесса добычи полезного ископаемого, включая горнотехнологические процессы.

Информационные технологии широко используются в дизайне в виде компьютерной графики, моделирования процессов и инженерных расчетов. Программный код библиотеки в формате dll имеет структуру, которая в большинстве языков программирования одинакова. Любая dll-библиотека должна включать в себя:

- модуль, поддерживающий очистку и инициализацию dll-библиотек;
- модули и функции, которые реализуют решения проблем в своей предметной области;
- требуемую структуру документации.

Модуль инициализации находится в форме функции или процедуры, которая принимает входные параметры, один из которых имеет значение, указывающее модификатор вызова. Значения, которые может принимать этот параметр, перечислены ниже.

Модуль инициализации возвращает параметр со значением true, если процесс успешен, или false, если во время инициализации произошла ошибка. Если параметр неверен, все дополнительные действия завершены [6].

Перед использованием библиотечных модулей необходимо связать библиотеку с программным обеспечением. Подключение может

быть сделано двумя способами. Первый – объявить отдельную внешнюю функцию или процедуру, а второй – динамически реализовать объявление с использованием функций Win32 API. Но все равно внутренние модули библиотеки будут загружены в память только при запуске приложения. Чтобы загрузить dll-библиотеку, операционная система должна найти и подгрузить ее. Стандартный поиск операционной системы осуществляется в следующих разделах:

- местоположение приложения, которое запрашивало доступ к библиотеке;
- текущий каталог;
- местоположение системы на диске, обычно адрес каталога следующий: C: \ Windows \ System32;
- расположение подраздела системы поддержки 16-битных программ;
- расположение самой операционной системы Windows;
- пути, указанные в переменной среде Path.

Для того, чтобы найти нужную библиотеку, необходимо просматривать разделы выше. Порядок просмотра может быть и другим, если библиотека не найдена системой - появится сообщение об ошибке поиска. Использование динамических библиотек дает следующие преимущества:

- экономить память и уменьшить загрузку. При использовании динамических библиотек несколько приложений обращаются к одной и той же библиотеке одновременно. В этом случае для использования библиотеки необходимо сохранить только одну копию библиотеки в системной памяти. При использовании статических библиотек операционная система должна загружать код библиотеки в память для каждой программы, которая нуждается в этой библиотеке;
- экономия дискового пространства. В случае параллельного использования библиотеки DLL многими программами на диске сохраняется только одна копия этой библиотеки. Напротив, при использовании статических библиотек для каждой программы создается отдельная копия этой библиотеки;
- удобство и простота смены и обновления dll-библиотек. Если вам нужно внести изменения или корректировки в dll-библиотеку, то после внесения изменений в библиотеку вам не придется снова компилировать или связывать приложение,

```

//Соединения
public class Soedin
{
    /*
     * Значения интенсивности отказов соединений при эксплуатации
     */
    //public static double Soedin_lmbd_e(double lmbd_b)
}

```

Рис. 1. Пример реализации класса  
Fig. 1. Example of class implementation

```

/*
 * среднее квадратическое отклонение (СКО) емкости ХИТ для
 * новых условий
 * (стр 582)
 */
public static double HimIstToka_sigma(double sigma_t, double Cj, double Ct, double Bt, double Bxp, double Bc)
{
    return (sigma_t * (Bt * Cj / Ct) / (Bxp * Bc));
}

```

Рис. 2. Пример описания public-метода  
Fig. 2. An example of a description of a public method

```

protected void Primer(double a, double b, double c)
{
    int i = (int)a;
    float j = (float)b;
    j = (float)(i + j) / (float)c;
}

```

Рис. 3. Пример описания protected метода  
Fig. 3. An example of a protected method description

```

/*
 * Математические модели для расчета эксплуатационной
 * интенсивности отказов отдельных групп (типов)
 * знаковинтезирующих индикаторов
 * Вакуумные накаливаемые, газоразрядные,
 * электролюминесцентные, жидкокристаллические,
 * сегнетокерамические
 * модель 1 (стр 225)
 */
private static double ZnSintezIndic_lmbd_e_VGEJS1(double lmbd_b, double Ke, double Kpr)
{
    return (lmbd_b * Ke * Kpr);
}

```

Рис. 4. Пример описания private-метода  
Fig. 4. An example of a private method description

которое использует эту библиотеку. Однако при использовании объектного кода статических библиотек вам придется перекомпилировать программу или ссылку снова в случае даже небольших изменений;

– обеспечение долгосрочной поддержки выпущенных библиотек. Обеспечивает простое обновление и добавление библиотечных функций для расширения возможностей или поддержки новых устройств;

```
/*
 * Математические модели для расчета эксплуатационной
 * интенсивности отказов отдельных групп (типов)
 * знаковосинтезирующих индикаторов
 * Вакуумные люминесцентные
 * 1 модель (стр 225)
 */
internal static double ZnSintezIndic_lmld_e_VL1(double lmbd_b, double Kt, double Ke, double Kpr)
{
    return (lmbd_b * Kt * Ke * Kpr);
}
```

*Рис. 5. Пример описания internal-метода*  
*Fig. 5. An example of describing an internal method*

```
/*
 * Расчет эксплуатационной интенсивности отказов
 * знаковосинтезирующих индикаторов, находящихся в
 * режиме ожидания
 * Для неподвижных объектов
 * Вакуумные накаливаемые, вакуумные люминесцентные,
 * газоразрядные, электролюминесцентные, жидкокристаллические,
 * сегнетокерамические
 * 2 вид (стр 225)
 */
protected internal double ZnSintezIndic_lmld_eh_VGEJS_N02(double lmbd_hsg, double Kusl, double Kpr)
{
    return (lmbd_hsg * Kusl * Kpr);
}
```

*Рис. 6. Пример описания protected internal метода*  
*Fig. 6. An example of a protected internal method description*

– поддержка многоязычных программ. Использование разных языков программирования при разработке программы не мешает использованию библиотеки, если язык программы и язык библиотеки используют одинаковые стандарты для функций вызова;

– упрощенное создание международных версий приложений. Простота использования библиотек DLL облегчает интернационализацию программы, если, например, она загружает все строковые ресурсы, используемые программой, в библиотеку DLL [9];

– единственным большим недостатком можно считать тот факт, что программа, которая использует библиотеку DLL, не может иметь полную функциональность в ее отсутствие.

Если приложение использует функцию из одной библиотеки или библиотека использует функцию из другой библиотеки, получается зависимость, вследствие которой приложение или библиотека становится зависимой, она теряет самодостаточность. В результате ошибки возникают во время следующих событий:

– соответствующая библиотека была обновлена до новой версии;

– внесены изменения в зависимую библиотеку DLL;

– соответствующий файл DLL был заменен более ранней версией;

– соответствующий файл DLL не был найден системой или был удален с компьютера.

Эти действия обычно называют конфликтами DLL. Если обратная совместимость не обеспечена, программа не будет успешно запущена. Такие действия называются конфликтами DLL. Программа может быть успешно запущена и нормально использоваться только при наличии обратной совместимости. Ассортимент программного обеспечения, предлагаемого на рынке информационных технологий, настолько широк, что обычному пользователю часто бывает трудно выбрать правильную программу или систему программ, которые явно удовлетворяли бы его потребностям. Это особенно актуально для специализированного программного обеспечения, созданного исключительно и предназначенного

```

public static double ZnSintezIndic_Kp(double Iprsr0, double Iprsr, double tp0, double tp, double Ea, double m)
{
    return (Math.Pow(Iprsr / Iprsr0, m) * Math.Exp(Ea /
        (8.617 * Math.Pow(10.0, -5.0))) * (1 / (tp0 + 273) - 1 / (tp + 273)));
}

```

Рис. 7. Пример описания формулы внутри оператора return  
Fig. 7. An example of describing a formula inside a return statement

для узкого круга пользователей. В этом смысле горнодобывающие компании являются отличным примером использования конкретных компьютерных программ, технологий и оборудования, где просто невозможно использовать общие решения для разных производственных условий. Основной причиной этого является изменчивость среды, котировки цен на бирже за соответствующий минерал и т. д., то есть это постоянная динамика объекта исследования.

Для реализации модуля математического аппарата системы расчета надежности современных радиоэлектронных средств в среде разработки Microsoft Visual Studio создается проект библиотеки классов, в результате компиляции которого получается dll-библиотека. Пример описания класса на языке C# в среде разработки приведен на Рис. 1.

Перед каждым классом идет комментарий, разъясняющий тип радиоэлектронных элементов, методы расчета надежности которого данный класс реализует. Ключевое слово public означает, что доступ к классу не ограничен. Внутри класса описываются его методы. Методы класса – это практически то же самое, что и функции, только доступ к ним реализуется через объект класса или непосредственно через класс, если метод статичный. Модификаторы доступа к методам класса могут быть нескольких видов:

– Public. Public-методы не ограничивают доступ к себе, позволяя использовать себя как внутри класса, так и за его пределами, например в «теле» основной программы. Пример описания public-метода приведен на Рис. 2.

– Protected. Protected-методы имеют ограничения для доступа. Эти методы возможно использовать только внутри класса, а также внутри классов-потомков. Классы-потомки – это классы, которые наследовались от данного класса – класса-родителя. Пример protected-метода приведен на Рис. 3.

– Private. Private-методы и свойства доступны только внутри своего класса. За пределами класса, а также в классах наследниках получить доступ к этому методу или свойству невозможно. Пример private-метода приведен на Рис.4.

– Internal. Internal-методы доступны для использования только внутри одной сборки

проекта. Пример Internal-метода приведен на Рис. 5.

– Protected internal. Данный модификатор разрешает доступ к методу из любого кода в сборке, а также из наследованного класса другой сборки. Доступ из другой сборки должен осуществляться в пределах объявления класса, производного от класса, в котором объявлен защищенный внутренний элемент, и должен происходить через экземпляр типа производного класса. Пример описания protected internal-метода приведен на Рис. 6.

Для реализации доступа к методам без создания объекта класса используется ключевое слово static. Этот модификатор активно используется в данной библиотеке классов, так как для доступа к расчету по необходимой формуле нерационально создавать объект класса, содержащего все расчеты для заданного типа радиоэлектронных элементов.

Для реализации расчетов активно используется встроенная в среду разработки библиотека Math. Данная библиотека предоставляет константы и статические методы для тригонометрических, логарифмических и иных общих математических функций.

При использовании методов данной библиотеки реализуются все необходимые формулы внутри методов. Сами формулы описываются внутри оператора return, который сразу по результату вычисления подает на выход метода результат вычисления. Пример подобной конструкции приведен на Рис. 7.

Соответственно, в итоге получается библиотека классов, содержащих методы для расчета всех необходимых формул. Далее происходит компиляция проекта, в результате которого создается файл библиотеки в формате dll.

Рассмотренный вариант использования библиотек DLL для учета параметров режима оптимизации процесса добычи – очень универсальный метод, позволяющий создавать программные системы с широким спектром приложений. На этапе предварительной добычи систему можно использовать для сбора, хранения, обработки и просмотра данных. На этапе проектирования система может использоваться для подготовки технико-экономического обоснования и оценки воздействия на окружающую среду, а также для выбора идеальных схем расположения скважин..

## СПИСОК ЛИТЕРАТУРЫ

1. Гаврилов П. Д., Гимельшейн Л. Я., Медведев А. Е. Автоматизация производственных процессов. Учебник для вузов. М.: Недра, 1985. 215 с.
2. Культин Н. Microsoft Visual C# в задачах и примерах / Н. Культин. - СПб.: БХВ-Петербург, 2009. 314 с.
3. Мак-Дональд М. WPF: Windows Presentation Foundation в .NET 4.0 с примерами на C# 2010 / Мак-Дональд М. – Изд.: Вильямс, 2011. 1020 с.
4. Нурлыбаев М. А. Автоматизированные системы управления технологическими процессами на карьерах / Учебное пособие для вузов. М.:Недра, 1985. 197 с.
5. Острейковский В. А. Теория надежности. М.: Высшая школа, 2003. 463 с.
6. Пучков Л. А., Федунец Н. И., Потресов Д. К. Автоматизированные системы управления в горнодобывающей промышленности / Учебник для вузов. М.: Недра, 1987. 285с.
7. Руссинович М. Внутреннее устройство Microsoft Windows. СПб.: Питер, 2013. 800 с.
8. Рихтер Д. Windows для профессионалов. СПб.:Питер, 2001. 752 с.
9. Саймон Р. Microsoft Windows API. Справочник системного программиста. Изд.: DiaSoft, 2004. 1217 с.
10. Фролов А. Операционная система Microsoft Windows 3.1 для программиста. М.: Диалог-МИФИ, 1993. 284 с.
11. Шилдт Г. C# 4.0 полное руководство. Изд.: Вильямс, 2011. 1056 с.
12. Троелсен Э. Pro C# 2010 and the .NET 4 Platform Пятое издание. Изд.: Вильямс, 2011. 1392 с.
13. Ватсон Б. C# 4.0 на примерах / Б. Ватсон.- СПб.: БХВ-Петербург, 2011. – 674 с.
14. Нейгел К. C# 4.0 и платформа .NET 4 для профессионалов / К. Нейгел, Б. Ивьен, Д. Глинн, К. Уотсон, М. Скиннер. – Изд.: Диалектика, 2011. – 1440 с.
15. Шилдт Г. C# 4.0 полное руководство / Г. Шилдт. – Изд.: Вильямс, 2011. – 1056 с.
16. Иванченко А. Н., Шайда А. Ю. Программирование приложений с использованием библиотек Южно-Российский государственный политехнический университет (НПИ) имени М. И. Платова, Редакционно-издательский отдел ЮРГПУ (НПИ) 2016. – 85с.
17. Евтушенко, С. И. Автоматизация и роботизация строит: Учебное пособие / С. И. Евтушенко, А. Г. Булгаков, В. А. Воробьев. - М.: Риор, 2018. - 752 с.
18. Деревяшкин, И. В. Гидромеханизация открытых горных работ. Гидромониторно-землесосные комплексы. Учебное пособие. Гриф МО РФ / И. В. Деревяшкин. - М.: ИНФРА-М, 2016. - 679 с.
19. Ржевский, В. В. Открытые горные работы. Производственные процессы. Книга 1 / В. В. Ржевский. - Москва: СИНТЕГ, 2016. - 122 с.
20. Чемезов, Е. Н. Безопасность ведения открытых горных работ. Учебное пособие. Гриф УМО вузов России / Чемезов Е. Н. - М.: Вузовская книга, 2008. - 912 с.
21. Егурнов, Г. Открытые горные работы / Г. Егурнов. - Москва: Гостехиздат, 2014. - 572 с.

## REFERENCES

1. Gavrilov P. D., Gimel'shejn L. Ya., Medvedev A. E. Avtomatizaciya proizvodstvennyh processov. Uchebnik dlya vuzov. M.: Nedra, 1985. 215 s.
2. Kul'tin N. Microsoft Visual C# v zadachah i primerah / N. Kul'tin. - SPb.: BHV-Peterburg, 2009. 314 s.
3. Mak-Donal'd M. WPF: Windows Presentation Foundation v .NET 4.0 s primerami na C# 2010 / Mak-Donal'd M. – Izd.: Vil'yams, 2011. 1020 s.
4. Nurlybaev M. A. Avtomatizirovannye sistemy upravleniya tekhnologicheskimi processami na kar'erah /Uchebnoe posobie dlya vuzov. M.:Nedra, 1985. 197 s.
5. Ostrejkovskij V. A. Teoriya nadezhnosti. M.: Vysshaya shkola, 2003. 463 s.
6. Puchkov L. A., Fedunec N. I., Potresov D. K. Avtomatizirovannye sistemy upravleniya v gornodobyvayushchej promyshlennosti /Uchebnik dlya vuzov. M.: Nedra, 1987. 285s.
7. Russinovich M. Vnutrennee ustrojstvo Microsoft Windows. SPb.: Piter, 2013. 800 s.
8. Rihter D. Windows dlya professionalov. SPb.:Piter, 2001. 752 s.
9. Sajmon R. Microsoft Windows API. Spravochnik sistemnogo programmista. Izd.: DiaSoft, 2004. 1217 s.
10. Frolov A. Operacionnaya sistema Microsoft Windows 3.1 dlya programmista. M.: Dialog-MIFI, 1993. 284 s.
11. Shildt G. C# 4.0 polnoe rukovodstvo. Izd.: Vil'yams, 2011. 1056 s.

12. Troelsen. E. Pro C# 2010 and the .NET 4 Platform Pyatoe izdanie. Izd.: Vil'yams, 2011. 1392 s.
13. Watson B. S# 4.0 na primerah / B. Watson.- SPb.: BHV-Peterburg, 2011. – 674 s.
14. Neigel K. C# 4.0 i platforma .NET 4 dlya professionalov / K. Neigel, B. Iv'en, D. Glinn, K. Uotson, M. Skinner. – Izd.: Dialektika, 2011. – 1440 s.
15. Shildt G. C# 4.0 polnoe rukovodstvo / G. SHildt. – Izd.: Vil'yams, 2011. – 1056 s.
16. Ivanchenko A. N., Shajda A. YU. Programirovanie prilozhenij s ispol'zovaniem bibliotek. Yuzhno-Rossijskij gosudarstvennyj politekhnicheskij universitet (NPI) imeni M. I. Platova, Redakcionno-izdatel'skij otdel YURGPU (NPI) 2016. – 85s.
17. Evtushenko, S. I. Avtomatizaciya i robotizaciya stroit: Uchebnoe posobie / S. I. Evtushenko, A. G. Bulgakov, V. A. Vorob'ev. - M.: Rior, 2018. - 752 c.
18. Derevyashkin, I. V. Gidromekhanizaciya otkrytyh gornyh rabot. Gidromonitorno-zemlesosnye komplekсы. Uchebnoe posobie. Grif MO RF / I. V. Derevyashkin. - M.: INFRA-M, 2016. - 679 c.
19. Rzhetskij, V. V. Otkrytye gornye raboty. Proizvodstvennye processy. Kniga 1 / V. V. Rzhetskij. - Moskva: SINTEG, 2016. - 122 c.
20. Chemezov, E. N. Bezopasnost' vedeniya otkrytyh gornyh rabot. Uchebnoe posobie. Grif UMO vuzov Rossii / Chemezov E. N. - M.: Vuzovskaya kniga, 2008. - 912 c.
21. Egurnov, G. Otkrytye gornye raboty / G. Egurnov. - Moskva: Gostekhizdat, 2014. - 572 c.

Поступило в редакцию 24.11.2020

Received 24 November 2020