


МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 23 » мая _____ 2023 г.

Рабочая программа дисциплины
«Системы программирования»

Направление подготовки
01.03.02 Прикладная математика и информатика

Направленность (профиль)
«Искусственный интеллект»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 21 апреля 2023 г.,
протокол № 8

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
28 апреля 2023 г.

Ярославль

1. Цели освоения дисциплины

Результаты изучения дисциплины «» востребованы при освоении дисциплин модулей: «Машинное обучение и анализ данных» и «Искусственный интеллект», преддипломной практике и выпускной квалификационной работе.

2. Место дисциплины в структуре образовательной программы бакалавриата (магистратуры, специалитета)

Дисциплина «Системы программирования» согласно учебному плану входит в модуль «Алгоритмизация и программирование» и реализуется в 4 семестре. Изучается на основе знаний, полученных при изучении дисциплин модулей «Аппаратное и программное обеспечение компьютера», «Современные цифровые технологии».

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы бакалавриата (магистратуры, специалитета)

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ОПК-2. Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач.	ИОПК2.1 Осуществляет выбор и адаптацию математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач.	Самостоятельно и грамотно осуществляет выбор и адаптацию математических методов и систем программирования для разработки и реализации алгоритмов решения прикладных задач, в том числе в новой или нестандартной ситуации.
ОПК-5. Способен разрабатывать алгоритмы и компьютерные	ИОПК5.1 Обладает знаниями в области алгоритмизации и программирования.	Демонстрирует глубокие знания в области алгоритмизации и программирования.

программы, пригодные для практического применения.	ИОПК5.2 Демонстрирует умение выбрать и обосновать выбор языка и среды программирования для разработки компьютерных программ.	Умеет выбирать и обосновать выбор языка и среды программирования для разработки компьютерных программ, в том числе в новой или нестандартной ситуации.
	ИОПК5.3 Владеет навыками разработки алгоритмов и компьютерных программ, пригодных для практического применения.	Полностью самостоятельно и верно разрабатывает алгоритмы и компьютерные программы, пригодные для практического применения.
Профессиональные компетенции		
ПК-1. Способен проектировать компьютерное программное обеспечение.	ИПК1.1 Разрабатывает и изменяет архитектуру компьютерного программного обеспечения.	Демонстрирует глубокое знание и понимание принципов построения и видов архитектуры компьютерного программного обеспечения, типовых решений, библиотек программных модулей, шаблонов, классов объектов, используемых при разработке компьютерного программного обеспечения, методов и средств проектирования компьютерного программного обеспечения. Полностью верно и самостоятельно использует существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения, применяет методы и средства проектирования компьютерного программного обеспечения.
	ИПК1.2 Проектирует структуру данных, баз данных и программных интерфейсов. согласовывает сроки выполнения поставленных задач.	Глубоко знает, полностью верно и самостоятельно умеет применять методы и средства проектирования баз данных и программных интерфейсов.

	<p>ИПК1.3 Разрабатывает техническую документацию на компьютерное программное обеспечение с использованием существующих стандартов, оценивает и согласовывает сроки выполнения поставленных задач.</p>	<p>Демонстрирует свободное владение стандартами в области разработки компьютерного программного обеспечения. Полностью верно и самостоятельно разрабатывает техническую документацию, оценивает и согласовывает сроки выполнения поставленных задач.</p>
<p>ПК-9. Способен оценить качество разрабатываемого программного обеспечения путем проверки соответствия продукта заявленным требованиям, сбора и передачи информации о несоответствиях.</p>	<p>ИПК9.1 Демонстрирует умение определять и описывать тестовые случаи на основе требований, заявленных к программному обеспечению.</p>	<p>Глубоко знает и понимает классификацию видов и типов тестирования программного обеспечения, техники проектирования и комбинаторики тестов, тестовые данные, обеспечивающие проверку безопасности программного обеспечения. Полностью верно и самостоятельно применяет техники проектирования тестов, анализирует тестовые случаи на предмет полноты учета покрытия, документирует тесты, разрабатывает скрипты и/или программные модули для автоматизации тестирования программного обеспечения, в том числе для проверки информационной безопасности разрабатываемого программного обеспечения.</p>
	<p>ИПК9.2 Проводит тестирование по разработанным тестовым случаям, осуществляет сбор информации о несоответствиях заявленным</p>	<p>Самостоятельно и полностью верно выполняет начальные настройки для проведения тестирования, необходимые виды тестирования.</p>

<p>ИПК9.3 Анализирует результаты тестирования и дает оценку качеству разрабатываемого программного обеспечения.</p>	<p>Демонстрирует глубокое знание и понимание типов дефектов программного обеспечения, их классификации и статистики возникновения. Полностью верно и самостоятельно определяет уровень критичности дефектов программного обеспечения, составляет отчеты об анализе результатов тестирования программного обеспечения.</p>
---	---

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единиц, 144 акад. часов.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости Форма промежуточной аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1	Объектно-ориентированное программирование.	4	6	0	6	0	0	14	
2	Классы в С#.	4	10	0	12	0	0	24	
3	Наследование и полиморфизм.	4	8	0	12	0	0	22	
4	Обобщения.	4	6	0	10	0	0	14	
	ИТОГО		30	0	40	0	0	74	Экзамен

Содержание разделов дисциплины:

Раздел 1. Объектно-ориентированное программирование.

Объектная модель: основные элементы объектной модели, отношения между объектами и классами. Принципы объектно-ориентированного программирования: абстрагирование; инкапсуляция; наследование; полиморфизм.

Раздел 2. Классы в С#.

Особенности платформы .NET. Отличия С# от С. Поля класса. Методы класса: основные понятия; рекурсивные методы; конструкторы и деструкторы.

Перезагрузка операций. Свойства и индексы.

Функциональные типы в С#: делегаты; события.

Раздел 3. Наследование и полиморфизм.

Наследование. Виртуальные функции и абстрактные классы. Создание иерархии исключений.

Раздел 4. Обобщение.

Основные понятия. Уточнения, используемые в обобщениях. Обобщенные интерфейсы.

Обобщенные методы. Обобщенные делегаты

5. Образовательные технологии, в том числе технологии электронного обучения и дистанционные образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

1. MozillaFirefox

6. Перечень лицензионного и (или) свободно распространяемого программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине

2. ОС семейства MicrosoftWindows
3. LibreOffice
4. Microsoft Office

7. Перечень современных профессиональных баз данных и информационных справочных систем, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)

В процессе осуществления образовательного процесса используются:

- Windows 7 в качестве основной операционной системы;
- Интегрированная среда разработки Microsoft Visual Studio для подготовки демонстрационных примеров, выполнения лабораторных работ и учебных упражнений;
- OfficeStd 2013 RUS OLP NL Acdmc 021-10232 для формирования инструкций по выполнению учебных упражнений, а также текстов материалов для промежуточной и текущей аттестации.

8. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», рекомендуемых для освоения дисциплины

а) основная литература

1. Ржевский, С. В. Математическое программирование : учебное пособие / С. В. Ржевский. — Санкт-Петербург : Лань, 2022. — 608 с. — ISBN 978-5-8114-3853-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/206993> 2.
Кривцов, А. Н. Алгоритмизация и программирование. Основы программирования на C/C++ : учебное пособие / А. Н. Кривцов, С. В. Хорошенко. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2020. — 202 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/180057>

б) дополнительная литература

1. Юрьева, А. А. Математическое программирование : учебное пособие / А. А. Юрьева. — 2-е изд., испр. и доп. — Санкт-Петербург : Лань, 2022. — 432 с. — ISBN 978-5-8114-1585-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/212210>
2. Рацеев, С. М. Программирование на языке Си : учебное пособие для вузов / С. М. Рацеев. — Санкт-Петербург : Лань, 2022. — 332 с. — ISBN 978-5-8114-8585-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/193320>

3. Гунько, А. В. Программирование : учебно-методическое пособие / А. В. Гунько. — Новосибирск : НГТУ, 2019. — 74 с. — ISBN 978-5-7782-3961-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/152231>
4. Касторнов, А.Ф. Программирование на языке Паскаль : учебное пособие для вузов / Касторнов А.Ф., Касторнова В.А., Козлов О.А. - Нижний Новгород : НГПУ им. К.Минина, 2012. - 152 с.
5. Павловская, Т.А. C#: программирование на языке высокого уровня : учебное пособие для вузов. - СПб. : Питер, 2013. - 432с.
6. Немнюгин, С.А. TurboPascal: Программирование на языке высокого уровня : учебник для вузов / Немнюгин С.А. - 2-е изд. - СПб. : Питер, 2007. - 543 с.

в) ресурсы сети «Интернет»

- 1 Интерактивная доска.
- 2 <http://www.ois.org.ua/spravka/mat/index.htm> - электронная библиотека по математике.
- 3 <http://eqworld.ipmnet.ru/ru/library.htm>- учебно-образовательная физико-математическая библиотека.
- 4 <http://www.exponenta.ru/>- образовательный математический сайт.

**Приложение № 1 к рабочей программе дисциплины
«Системы программирования»**

**Фонд оценочных средств
для проведения текущего контроля успеваемости
и промежуточной аттестации студентов
по дисциплине**

1. Типовые контрольные задания и иные материалы,
используемые в процессе текущего контроля успеваемости

Примерные задания для лабораторных работ:

Задание 1. Создание класса

Напишите программу, создающую класс и несколько объектов данного класса.

1. Создайте класс `planet`, который содержит информацию о планете солнечной системы, включающий следующие атрибуты: название, диаметр, массу и расстояние от Солнца в тысячах километров в закрытой части класса. Включите в класс метод, который возвращает расстояние от Солнца в милях и метод, выводящий всю информацию о планете на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
2. Создайте класс `file`, описывающий файл на диске. Закрытыми элементами класса будут: имя файла, размер в битах, атрибуты файла, дата и время создания. Включите в класс метод, который вычисляет размер файла в байтах, килобайтах и мегабайтах, а также метод вывода всех данных о файле на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
3. Создайте класс `card`, который поддерживает каталог библиотечных карточек. Этот класс содержит название книги, имя автора, выданное на руки число экземпляров, дату выдачи книги читателю и дату возврата книги в библиотеку, в закрытой части класса. Включите в класс методы подсчета количества дней, которые книга находится на руках читателя, и вывода всех данных о книге на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
4. Создайте класс `nomenclature`, описывающий товары на складе магазина. Закрытыми элементами класса будут: название товара, оптовая цена, розничная наценка и количество товаров на складе. Включите в класс методы подсчета возможного чистого дохода при продаже этого товара и вывода всех данных о товаре на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
5. Создайте класс `persona`, который содержит информацию о ФИО человека, дате рождения, поле и адресе человека закрытой части класса. Включите в методы подсчета количества дней, оставшихся до следующего дня рождения и вывода данных о человеке на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
6. Создайте класс `soft`, который содержит информацию об установленном программном обеспечении. Закрытыми элементами класса будут: название программы, разработчик, занимаемый объем, дата завершения лицензии. Включите в класс методы подсчета количества дней до завершения лицензии и вывода всех данных об установленном

программном обеспечении на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.

7. Создайте класс `car`, содержащий информацию об автомобилях. Закрытыми элементами класса будут: название модели, изготовитель, количество лошадиных сил, расход топлива на 100 км и массу автомобиля. Включите в класс методы подсчета количества топлива, необходимого для того, чтобы проехать расстояние в 1000 км и вывода всех данных об автомобиле на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
8. Создайте класс `worker`, содержащий информацию о сотрудниках предприятия. Закрытыми элементами класса будут: ФИО сотрудника, табельный номер, название отдела, должность и дата принятия. Включите в класс методы расчета стажа сотрудника (количество лет, месяцев и дней) и вывода всех данных о сотруднике на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
9. Создайте класс `fluid`, содержащий информацию о жидких веществах. Закрытыми элементами класса будут: название вещества, цвет, запах, плотность жидкости. Включите в класс методы расчета массы жидкости в одном кубическом метре и вывода всех данных о жидкости на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.
10. Создайте класс `country`, который содержит информацию о названии, форме правления, численности населения и площади страны в закрытой части класса. Включите в класс методы расчета плотности населения страны и вывода всех данных о стране на экран. Для инициализации и удаления объектов класса используйте конструкторы и деструкторы.

Задание 2. Иерархия классов

Напишите программу, в которой создается иерархия классов. Покажите использование виртуальных методов.

Перечень классов:

1. студент, преподаватель, персона, заведующий кафедрой;
2. служащий, персона, рабочий, инженер;
3. рабочий, кадры, инженер, администрация;
4. двигатель, двигатель внутреннего сгорания, дизель, турбореактивный двигатель;
5. журнал, книга, печатное издание, учебник;
6. тест, экзамен, выпускной экзамен, испытание;
7. место, область, город, мегаполис;
8. квитанция, накладная, документ, чек;
9. автомобиль, поезд, транспортное средство, экспресс;
10. млекопитающие, парнокопытные, птицы, животное.

Образцы вопросов для защиты лабораторных работ:

1. Приведите определения понятий «класс» и «объект», изучив материал лекции и официальную документацию разработчика среды программирования.

2. Изучите материалы лекции и официальную документацию разработчика среды программирования и приведите пример объявления класса.
3. Для чего используются поля класса?
4. Изучите материалы лекции и официальную документацию разработчика среды программирования и приведите пример описания полей класса.
5. Как присвоить значение полю?
6. Для чего предназначены методы класса?
7. Изучите материалы лекции и официальную документацию разработчика среды программирования и приведите пример объявления метода класса.
8. Как осуществляется доступ к методам?
9. Изучите материалы лекции и официальную документацию разработчика среды программирования и приведите пример конструктора класса.
10. Сколько конструкторов может иметь класс?
11. Как описать конструктор?
12. Для чего используются деструкторы?
13. Сколько деструкторов может иметь класс?
14. Изучите материалы лекции и официальную документацию разработчика среды программирования и приведите пример деструктора класса.

2. Список вопросов и (или) заданий для проведения промежуточной аттестации

Вопросы к экзамену:

1. Основные элементы объектной модели.
2. Отношения между объектами и классами.
3. Абстрагирование.
4. Инкапсуляция.
5. Наследование.
6. Полиморфизм.
7. Особенности платформы .NET.
8. Классы с C#, поля класса.
9. Указание области видимости: public; private; protected; internal.
10. Статические поля.
11. Константы.
12. Методы класса: синтаксис; тип возвращаемого значения; список формальных параметров.
13. Модификаторы доступа к методам: public, private, protected, internal, static.
14. Методы с переменным числом параметров.
15. Определение выходных параметров метода.
16. Вызов метода.
17. Рекурсивные методы.
18. Конструкторы.

19. Деструкторы.
20. Перегрузка методов.
21. Перегрузка операторов.
22. Свойства класса.
23. Индексаторы класса.
24. Функциональные типы в C#, делегаты.
25. Функциональные типы в C#, события.
26. Наследование.
27. Виртуальные функции.
28. Абстрактные классы.
29. Создание иерархии исключений.
30. Обобщения, основные понятия.
31. Уточнения, используемые в обобщениях.
32. Обобщенные интерфейсы.
33. Обобщенные методы.
34. Обобщенные делегаты.

Уровни оценки компетенций следующие: базовый – 55-69 баллов, повышенный – 70-100 баллов.

Преподаватель проводит систематический контроль знаний студентов, ориентируясь на перечень вопросов для проведения зачета/экзамена.

Критерии оценки лабораторных работ / практических занятий / самостоятельной работы студента (от 0 до 10 баллов):

- **9-10 баллов** выставляется студенту, если работа выполнена самостоятельно и полностью верно; представлен отчет, содержащий результаты выполнения заданий работы и ответы на контрольные вопросы; студент анализирует результаты, полученные в ходе выполнения работы, делает выводы.
- **7-8 баллов** выставляется студенту, если работа выполнена самостоятельно, в целом правильно, но имеются некоторые неточности в выполнении заданий или ответах на контрольные вопросы; представлен отчет, содержащий результаты выполнения заданий и ответы на контрольные вопросы; студент анализирует результаты, полученные в ходе выполнения работы, делает выводы.
- **5-6 баллов** выставляется студенту, если работа выполнена самостоятельно, в целом правильно, но имеются некоторые неточности в выполнении заданий или ответах на контрольные вопросы; представлен отчет, содержащий результаты выполнения заданий лабораторной работы и ответы на контрольные вопросы; студент испытывает затруднения при проведении анализа результатов, полученных в ходе выполнения лабораторной работы, и формулировке выводов.
- **3-4 балла** выставляется студенту, если студент не до конца справился с заданием, не совсем верно ответил на контрольные вопросы, однако оформил отчет по результатам работы.

- **1-2 балла** выставляется студенту, если студент не до конца справился с заданием, не совсем верно ответил на контрольные вопросы, не оформил отчет по результатам работы.
- **0 баллов** выставляется студенту, если студент не справился с заданием, неверно ответил на представленные вопросы.

Ответ на зачете/экзамене оценивается исходя из 40 баллов (максимум). Билет содержит теоретический вопрос и практическое задание, преподаватель может задавать дополнительные вопросы. Полный ответ на основной вопрос оценивается максимум в 20 баллов, предполагает свободное изложение (не чтение) всего необходимого материала, ответы студента на уточняющие вопросы, если они есть. Правильный ответ на дополнительный вопрос оценивается максимум в 5 баллов. Правильное выполнение практического задания оценивается в 20 баллов.

Шкала оценивания компетенций:

Оценка в 100-балльной шкале	Оценка в 5-ти балльной шкале	Уровень сформированности компетенций
0-54 баллов	неудовлетворительно (не зачтено)	недостаточный
55-69 баллов	удовлетворительно (зачтено)	базовый
70-85 баллов	хорошо (зачтено)	повышенный
86-100 баллов	отлично (зачтено)	

Приложение № 2 к рабочей программе дисциплины «Системы программирования»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Системы программирования» являются традиционные лекции, которые включают в себя большое количество примеров кода, что позволяет сделать материал лекций более наглядными, улучшает информативность и понимание изучаемого курса.

По большинству тем предусмотрены лабораторные работы, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и получение навыков разработки и отладки программных приложений. Выполнение этих лабораторных работ, а также выполнение индивидуальных заданий позволяет не только понять и закрепить теоретический материал, но и приобрести навык анализа предметной области и создания прикладных приложений на языке С.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются и обсуждаются на лекциях и лабораторных занятиях. Основная цель решения задач – помочь усвоить фундаментальные понятия и основы разработки и отладки программных приложений. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной при выполнении лабораторных работ или из учебной литературы.

Большое внимание должно быть уделено самостоятельной домашней работе. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и лабораторных занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков программирования на языке С, в течение всего периода обучения проводятся консультации и разбор самостоятельного выполнения индивидуальных заданий.

В конце семестра студенты сдают экзамен. Экзамен проводится в письменной форме. Задания позволяют проверить знание синтаксиса языка, умение понимать чужой код и, конечно, умение построить и реализовать на С свой алгоритм.

Освоить вопросы, излагаемые в процессе изучения дисциплины "Основы программирования" самостоятельно первокурснику достаточно сложно. Хорошее усвоение материала подразумевает не только умение составлять и реализовывать алгоритмы, но и понимание механизмов взаимодействия программы с операционной системой, способов хранения данных и доступа к ним, в том числе с использованием косвенной адресации. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Не в меньшей степени это относится и к лабораторным занятиям, поскольку одной из важнейших целей данного учебного курса является формирование именно практических навыков программирования. Так что без упорных и

регулярных занятий в течение семестра сдать экзамен по итогам изучения дисциплины студенту будет сложно.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 8 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» (www.biblioclub.ru) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека «Книгообеспеченность» (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд

научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.