

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Ярославский государственный университет им. П.Г. Демидова

Кафедра информационных и сетевых технологий

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

«_18_» мая 2020 г.

Рабочая программа дисциплины
«Скриптовые языки программирования»

Направление подготовки
02.03.02 Фундаментальная информатика и информационные технологии

Профиль
«Информатика и компьютерные науки»

Квалификация выпускника
Магистр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 27 апреля 2020 г.,
протокол № 9

Программа одобрена НМК
факультета ИВТ
протокол № 7 от
17 мая 2020 г.

Ярославль

1. Цели освоения дисциплины

Целями дисциплины «Скриптовые языки программирования» являются освоение подходов к программированию на скриптовых языках программирования, таких как Python, R и JavaScript. Данный курс вырабатывает у студентов алгоритмическое мышление, умение применять основные концепции и классические алгоритмы современной информатики и эффективно решать возникающие задачи на практике.

2. Место дисциплины в структуре ОП магистратуры

Дисциплина «Скриптовые языки программирования» является факультативной дисциплиной ОП магистратуры.

Для освоения данной дисциплиной студенты должны обладать знаниями по математике и информатике в объеме школьной программы, проявлять настойчивость, целеустремленность и инициативу в процессе обучения.

Полученные в рамках дисциплины «Скриптовые языки программирования» знания необходимы для развития алгоритмического мышления, развития навыков решения сложных задач, изучения профильных курсов по программированию.

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП магистратуры

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП 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
Профессиональные компетенции		
ПК-4 Способность принимать эффективные проектные решения в условиях неопределенности и риска	ПК-4.1. обладает способностью принимать эффективные проектные решения в условиях неопределенности и риска ПК-4.2. обладает способностью использовать и развивать методы научных исследований и инструментария в области проектирования и управления информационными системами в прикладных областях ПК-4.3 демонстрирует способность анализировать данные и оценивать требуемые знания для решения нестандартных задач с использованием математических методов и методов компьютерного моделирования	Знать: – основные структуры данных в языках Python, R и JavaScript; – управляющие конструкции языков Python, R и JavaScript. Уметь: – разрабатывать тесты для предлагаемых решений; – использовать лучшие практики программирования для решения прикладных задач; Владеть навыками: – программирования на языке Python, R и JavaScript.

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 акад. часа.

№ 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 промежуточно й аттестации (по семестрам)
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1	Раздел 1. Основы программирования на языке Python	5	5		14	1		16	Индивидуальн ое задание №1 Зачет
2	Раздел 2. Основы объектно- ориентированного программирования на языке Python	5	5		13	1		16	Индивидуальн ое задание №2 Зачет
3	Раздел 3. Основы программирования на языке R	5	4		14	2		16	Индивидуальн ое задание №3 Зачет
4	Раздел 4. Основы программирования на языке JavaScript	5	4		13	3		16,7	Индивидуальн ое задание №4 Зачет
									Зачет
	Всего за 4 семестр		18		54	7		64,7	
	Всего		18		54	7		64,7	

Содержание разделов дисциплины:

Раздел 1. Основы программирования на языке Python.

1.1. Арифметические выражения и числовые типы данных, переменные, логические операторы и условный переход.

1.2. Локальные и глобальные переменные, основы обработки исключений, циклы, работа со строками.

1.3. Основные структуры данных: списки, словари и кортежи.

Раздел 2. Основы объектно-ориентированного программирования на языке Python.

2.1. Определение классов в Python.

2.2. Ссылки на объекты.

2.3. Наследование в Python.

Раздел 3. Основы программирования на языке R.

- 3.1. Обзор и история языка R.
 - 3.2. Типы данных в R.
 - 3.3. Управляющие структуры в R – циклы, условный оператор.
 - 3.4. Функции в R.
 - 3.5. Профилирование программ в R.
- Раздел 4. Основы программирования на языке JavaScript.
- 4.1. Язык сценариев JavaScript.
 - 4.2. Операторы и функции.
 - 4.3. Объекты, строки, числа и массивы.
 - 4.4. Объектная модель документа.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются:

- для формирования текстов материалов для промежуточной и текущей аттестации
- программы Microsoft Office, издательская система LaTeX;
- компиляторы с высокоуровневых языков программирования;
- для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

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

а) основная:

1. Доусон М. Програмируем на Python. СПб.: Питер, 2015. – 416 с.
2. Р. И. Кабаков «R в действии. Анализ и визуализация данных на языке R» // ДМК Пресс, 2014. 580 с.
3. Крокфорд Д. JavaScript: сильные стороны. / Д. Крокфорд; [пер. с англ. А. Лузгана] - СПб.: Питер, 2013. - 173 с.

б) дополнительная:

1. Саммерфильд М. Python на практике. ДМК Пресс, 2014. – 338 с.

2. Лутц М. Python: карманный справочник. Вильямс, 2015. – 320 с.
3. С. Мостицкий, В. Шитиков «Статистический анализ и визуализация данных с помощью R» // ДМК Пресс, 2015. 496 с.

в) ресурсы сети «Интернет»

1. Электронная библиотека учебных материалов ЯрГУ
(http://www.lib.uni Yar.ac.ru/opac/bk_cat_find.php).
2. Электронно-библиотечная система «Юрайт»(<https://urait.ru/>).
3. Электронно-библиотечная система «Лань»(<https://e.lanbook.com/>).
4. Timus Online Judge. Архив задач с проверяющей системой (acm.timus.ru).

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Аудитории, оборудованные для проведения лекций, практических занятий и консультаций, фонд библиотеки, компьютерная техника.

Автор(ы) :

Зав. кафедрой информационных и сетевых технологий, к.ф.-м.н. Д.Ю. Чалый

**Приложение №1 к рабочей программе дисциплины
«Скриптовые языки программирования»
Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

1.1. Контрольные задания и иные материалы, используемые в процессе текущей аттестации

Задания для самостоятельной работы

Задания по теме № 1. «Основы программирования на языке Python»

1. Приведенный ниже код содержит синтаксические ошибки. Исправьте их. После исправления ошибок программа должна напечатать два числа. Первое из них 1.09888451159. Приведите второе число (в ответ запишите число округленное до тысячных).

```
def project_to_distance(point_x point_y distance):  
    dist_to_origin = math.square_root(pointx ** 2 + pointy ** 2)  
    scale == distance / dist_to_origin  
    print point_x * scale, point_y * scale
```

```
project-to-distance(2, 7, 4)
```

2. Каким будет значение переменной count после выполнения следующей программы?

```
count = 0  
def square(x):  
    global count  
    count += 1  
    return x**2  
  
print(square(square(square(3))))
```

3. Какие из имен в следующей программе встречаются в глобальном контексте, а какие в локальном?

```
a = 3  
b = 6  
  
def f(a):  
    c = a + b  
    return c
```

4. Каким будет значение переменной istr после выполнения следующей программы?

```
astr = "Hello Bob"  
istr = 0  
try:  
    istr = int(astr)  
except:  
    istr = -1
```

5. Что напечатает следующий код:
print(len('apple'*5))

6. Что делает следующая программа:
- ```
fhand = open("myfile.txt")
x = 0
for line in file:
 x = x + 1
print(x)
```
7. Чему может равняться значение переменной s, чтобы следующая программа что-нибудь вывела на экран?
- ```
x = (4, 2, 9)
if s > x:
    print("Bigger!")
```
8. Что напечатает следующая программа?
- ```
stuff = dict()
stuff['pear'] = 'tasty'
print(stuff['candy'])
```
9. Приведите пример входных данных, для которых программа выведет значение 5?
- ```
s = input("Введите строку:")
words = dict()
m = 0
for w in s.split():
    c = words.get(w, 0)
    if c >= m:
        m = c + 1
    words[w] = c + 1

print(m)
```

Задания по теме №2. «Основы объектно-ориентированного программирования на языке Python»

1. Разработайте набор классов для моделирования карточной колоды и отдельной карты.
2. Разработайте класс для моделирования отдельной персоны и нескольких персон, которые в совокупности составляют семью.
3. Разработать класс, который моделирует банковский счет и типовые операции по этому счету.

Задания по теме №3. «Основы программирования на языке R»

Пусть вам дан набор данных, в котором содержатся данные о загрязнении воздуха. Набор содержит следующие измеряемые характеристики: дата и время измерения, уровень сульфатов в воздухе (в мкг/м³), уровень нитратов в воздухе (в мкг/м³). При этом набор данных состоит из нескольких сотен файлов, по одному файлу для каждого места измерения. Задачи:

1. Реализовать функцию, позволяющую вывести среднее загрязнение по любому выбранному веществу и набору мест измерения.

Для каждого файла вывести количество сделанных измерений, в которых измерено и значение нитрата, и значение сульфата.

Задания по теме №4. «Основы программирования на языке JavaScript»

Запрограммируйте с использованием JavaScript приложение, реализующее адресную книгу.

Список заданий к зачету

Зачет выставляется по результатам тестового задания и краткого собеседования со студентом после его проверки. Тестовое задание является упрощенным заданием по сравнению с заданием для самостоятельной работы.

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Общепрофессиональные компетенции						
ПК-4	Индивидуальное задание №1,2,3,4. Зачет	1-4	Знать: – основные структуры данных в языках Python, R и JavaScript; – управляющие конструкции языков Python, R и JavaScript. Уметь: – разрабатывать тесты для предлагаемых решений; – использовать лучшие практики программирования для решения прикладных задач; Владеть навыками: – программирования на языке Python, R и JavaScript.	1. Знать основные структуры данных современной информатики и их использование в языках Python, R, JavaScript. 2. Уметь программировать основные алгоритмы современной информатики на Python, R, JavaScript .	1. Уметь создавать оригинальные структуры данных для решения типовых задач и писать для них код на Python, R, JavaScript.	1. Проявлять значительные творческие способности в программировании на языках Python, R, JavaScript.

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объёме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;

- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка. Для дисциплин, изучаемых в течение нескольких семестров, оценка может выставляться не только по окончании ее освоения, но и в промежуточных семестрах. Вид оценки («отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «зачтено», «незачтено») определяется рабочей программой дисциплины в соответствии с учебным планом.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Приложение №2 к рабочей программе дисциплины «Скриптовые языки программирования»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Скриптовые языки программирования» являются лабораторные работы, причем в достаточно большом объеме. Это связано с тем, что для программирования на скриптовых языках необходима практика. По большинству тем предусмотрены практические занятия, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и отработка практических навыков программирования.

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 на лекциях и практических занятиях, при необходимости по наиболее трудным темам проводятся дополнительные консультации. Основная цель решения задач – помочь усвоить основные конструкции языков программирования и приемов написания работающего кода.

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

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков работы с аппаратом скриптовых языков, в течение обучения проводятся мероприятия текущей аттестации в виде зачета. Также проводятся консультации (при необходимости) по разбору заданий для самостоятельной работы, которые вызвали затруднения.

В конце обоих семестров изучения дисциплины студенты сдают зачет. Зачеты выставляются по итогам тестирования и краткого собеседования по его результатам.

Освоить вопросы, излагаемые в процессе изучения дисциплины «Скриптовые языки программирования» самостоятельно студенту крайне сложно. Это связано со сложностью изучаемого материала и большим объемом курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет по материалу дисциплины студенту очень сложно.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу. К таким можно отнести следующие издания:

1. Доусон М. Програмируем на Python. СПб.: Питер, 2015. – 416 с.
2. Р. И. Кабаков «R в действии. Анализ и визуализация данных на языке R» // ДМК Пресс, 2014. 580 с.
3. Крокфорд Д. JavaScript: сильные стороны. / Д. Крокфорд; [пер. с англ. А. Лузгана] - СПб.: Питер, 2013. - 173 с.

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. Информационная система "Единое окно доступа к образовательным ресурсам" (<http://window.edu.ru/library>).

Целью создания информационной системы "Единое окно доступа к образовательным ресурсам" (ИС "Единое окно ") является обеспечение свободного доступа к интегральному каталогу образовательных интернет-ресурсов и к электронной библиотеке учебно-методических материалов для общего и профессионального образования.

Информационная система "Единое окно доступа к образовательным ресурсам" создана по заказу Федерального агентства по образованию в 2005-2008 гг. Главной разработчик проекта - Федеральное государственное автономное учреждение Государственный научно-исследовательский институт информационных технологий и телекоммуникаций (ФГАУ ГНИИ ИТТ "Информика") www.informika.ru.

ИС "Единое окно" объединяет в единое информационное пространство электронные ресурсы свободного доступа для всех уровней образования в России. Разделы этой системы:

- [Электронная библиотека](#) – является крупнейшим в российском сегменте Интернета хранилищем полнотекстовых версий учебных, учебно-методических и научных материалов с открытым доступом. Библиотека содержит более 30 000 материалов, источниками которых являются более трехсот российских вузов и других образовательных и научных учреждений. Основу наполнения библиотеки составляют электронные версии учебно-методических материалов, подготовленные в вузах, прошедшие рецензирование и рекомендованные к использованию советами факультетов, учебно-методическими комиссиями и другими вузовскими структурами, осуществляющими контроль учебно-методической деятельности.

- Интегральный [каталог](#) образовательных интернет-ресурсов содержит представленные в стандартизированной форме метаданные внешних ресурсов, а также содержит описания полнотекстовых публикаций электронной библиотеки. Общий объем каталога превышает 56 000 метаописаний (из них около 25 000 - внешние ресурсы). Расширенный поиск в "Каталоге" осуществляется по названию, автору, аннотации, ключевым словам с возможной фильтрацией по тематике, предмету, типу материала, уровню образования и аудитории.

- Избранное. В разделе представлены подборки наиболее содержательных и полезных, по мнению редакции, интернет-ресурсов для общего и профессионального образования.

- Библиотеки вузов. Раздел содержит подборки сайтов вузовских библиотек, электронных каталогов библиотек вузов и полнотекстовых электронных библиотек вузов.

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.