МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ С.А. ЕСЕНИНА»

Утверждаю:
Декан физико-математического
факультета
Н.Б. Федорова
«30» августа 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

АЛГОРИТМИЗАЦИЯ И ОСНОВЫ СТРУКТУРНОГО ПРОГРАММИРОВАНИЯ

Уровень основной профессиональной образовательной программы: **бакалавриат**

Направление подготовки: **02.03.03 Математическое обеспечение и администрирование информационных систем**

Направленность (профиль) подготовки: **Администрирование информационных систем**

Форма обучения: **очная**

Срок освоения ОПОП: **нормативный срок освоения 4 года**

Факультет: **физико-математический**

Кафедра: **Информатики, вычислительной техники и методики преподавания информатики**

Рязань, 2019

ВВОДНАЯ ЧАСТЬ

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения учебной дисциплины «Алгоритмизация и основы структурного программирования» является формирование у обучающихся профессиональных компетенций в процессе изучения алгоритмизации и основ структурного программирования для последующего применения в учебной и практической деятельности.

Задачи дисциплины:

- систематизация, формализация и расширение знаний по основам информатики, приобретенные в школе;
- привитие навыков алгоритмического мышления, культуры алгоритмизации и нисходящего структурного программирования;
- формирование теоретической базы и практических умений и навыков для решения задач на компьютере в императивных системах программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВУЗА

2.1. Дисциплина Б1.В.03.01 «Алгоритмизация и основы структурного программирования» относится к модулю Программирование IV части, формируемой участниками образовательных отношений, Блока 1.

2.2. Для изучения данной дисциплины необходима дисциплина предшествующего уровня образования:

– *Информатика и информационные технологии*.

2.3. Перечень последующих дисциплин, для которых необходимы знания, умения и навыки, формируемые данной дисциплиной:

– *Основы программирования*.

2.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения основной профессиональной образовательной программы

Изучение данной дисциплины направлено на формирование у обучающихся общепрофессиональных (ОПК) компетенций

№ п/п	Код и содержание компетенции	Код и наименование индикатора достижения компетенции	Перечень планируемых результатов обучения по дисциплине В результате изучения дисциплины обучающиеся должны:		
			Знать	Уметь	Владеть
1	*2*	*3*	*4*	*5*	*6*
1.	ПКВ-1. Способен разрабатывать алгоритмы решения поставленных задач, реализовывать и отлаживать программные коды с использованием различных технологий программирования	ПКВ-1.1. Способен составлять формализованные описания и алгоритмы решений поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов	• понятие алгоритма и его основные свойства; • основные понятия алгоритмической системы: представление информации в виде данных, система команд исполнителя, алгоритмический язык исполнителя; • основные средства записи и типы алгоритмов; • алгоритмические структуры, их основные свойства и приемы использования	• при решении стандартных задач профессиональной деятельности конкретизировать и описать основные понятия, результаты других научных дисциплин, данные и их типы средствами записи алгоритмов и программ; • записывать алгоритмы на языке схем; • анализировать структуру алгоритмов; • описывать алфавит и основные синтаксические конструкции языков Питон и Паскаль с помощью нормальных форм Бэкуса-Наура	• навыками составления алгоритмов решений поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов
2.		ПКВ-1.2. Способен осуществлять выбор технологии, языка и системы программирования для написания программных кодов решения задач в различных предметных областях	• этапы решения задачи на компьютере на основе парадигмы императивного программирования; • цели и составные части технологии нисходящего структурного проектирования алгоритмов и программ	• разрабатывать и записывать алгоритмы и программы на языках Питон и Паскаль в соответствие с технологией нисходящего структурного проектирования	• навыками выбора технологии, языка и системы программирования для написания программных кодов решения задач в средах, по крайней мере, двух систем программирования

3.		ПКВ-1.3. Способен разрабатывать, реализовать, отлаживать программный код в соответствии с техническим заданием, осуществлять рефакторинг программного кода	• составные части алгоритмического языка программирования (алфавит, синтаксис, семантика) и способы их описания; • концепцию типов данных; • структуры программ на языках программирования Питон и Паскаль; • основные понятия алгоритмических языков программирования и их классификацию: идентификаторы, данные и их типы, операции, стандартные функции, выражения, операторы; • синтаксис и семантику основных операторов языков программирования Питон и Паскаль (присваивания, ввода-вывода, ветвления, циклов), способы реализации алгоритмических структур средствами языков программирования; • основные синтаксические конструкции, используемые в языках программирования Питон и Паскаль для описания и обработки данных типа массив; • прием флаг	• разрабатывать и записывать программы на языках Питон и Паскаль в соответствие с принципом структурности; • при решении учебной задачи конкретизировать и описать основные понятия, данные и их типы, организовать ввод и вывод, реализовать алгоритмические структуры средствами, по крайней мере, двух языков программирования; • разрабатывать алгоритмы и программы обработки массивов	• навыками разработки, реализации, отладки программного кода в соответствии с техническим заданием, осуществлять рефакторинга программного кода в средах, по крайней мере, двух систем программирования.

ОСНОВНАЯ ЧАСТЬ
1. ОБЪЕМ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

Вид учебной работы		**Всего часов**	**Семестры**
			№ 1
			часов
1		2	4
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)		**84**	**84**
В том числе:			
Лекции (Л)		34	34
Лабораторные работы (ЛР)		50	50
Самостоятельная работа студента (всего)		**96**	**96**
В том числе:			
Изучение литературы и других источников		26	26
Подготовка к выполнению лабораторных работ		34	34
Подготовка к защите лабораторных работ		34	34
Контроль		**36**	**36**
Вид промежуточной аттестации	**зачет (З)**		
	экзамен (Э)		**+**
ИТОГО: общая трудоемкость	**часов**	**216**	**216**
	зач. ед.	**6**	**6**

2. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ
2.1. Содержание разделов дисциплины

№ семестра	№ раздела	Наименование раздела учебной дисциплины	Содержание раздела в дидактических единицах
1	2	3	4
1	1	Алгоритмическая система и ее составные части	Алгоритмическая система как совокупность средств и понятий. Понятие алгоритма и исполнителя алгоритма. Представление информации в виде данных: понятие о вводе/выводе; входные, выходные и промежуточные данные. Система команд исполнителя. Константы, переменные, операции, функции и выражения. Понятие типа данных. Алгоритмический язык исполнителя. Принципы фон Неймана как основа структуры персонального компьютера. Представление в компьютере целых чисел. Представление в компьютере действительных чисел. Средства записи алгоритмов. Схемы алгоритмов
1	2	Основные понятия и простейшие средства алгоритмических языков программирования	Программирование как этап решения задачи на компьютере. Понятие языка программирования высокого уровня. Понятие об исполнителе программы на языке высокого уровня. Составные части языка высокого уровня: алфавит, синтаксис, семантика. Метаязыки описания конструкций языков программирования. Расширенные Бэкуса-Наура формы, примеры метаформул. Структуры программ и концепции типов данных в алгоритмических языках на примере языков Python и PascalABC. Оператор как логически завершенная

			конструкция алгоритмического языка программирования. Различные классификации операторов: исполняемые и неисполняемые, простые и составные. Различные классификации типов данных: простые и структурированные, изменяемые и неизменяемые, стандартные и определяемые пользователем. Основные простые типы данных в языках Python и Pascal (кроме логического): идентификаторы стандартных типов, диапазоны, операции и функции, синтаксис типа диапазон. Особенности целочисленной и действительной арифметики в системах программирования (СП) Python и PascalABC. Синтаксис разделов программы на языке PascalABC, понятие порядкового типа в языке Pascal. Синтаксис и семантика оператора присваивания, знакомство с операторами ввода/вывода в языках Python и Pascal. Общая характеристика СП Python и PascalABC: основное меню, окна редактирования и результатов исполнения программы, диалоговые окна, встроенная система помощи. Написание, ввод, отладка и тестирование программ линейной структуры в СП Python и PascalABC
1	3	Технология нисходящего структурного проектирования. Разработка разветвлящихся алгоритмов и программ в соответствии с принципом структурности	Цели и составные части технологии нисходящего структурного проектирования. Взаимосвязь принципов нисходящего проектирования, модульности и структурности. Алгоритмические структуры как основа структурного кодирования. Базовые алгоритмические структуры: следование и развилка. Вложение (суперпозиция) алгоритмических структур как основной прием принципа структурности. Присоединение алгоритмических структур. Правила пунктуации в программе на алгоритмическом языке. Составной оператор в языке Pascal. Логический тип данных в алгоритмических языках Python и Pascal. Операции отношения и логические операции. Алгоритмические структуры, реализующие ветвления. Организация ветвлений средствами алгоритмических языков: синтаксис и семантика условных операторов в языках Python и Pascal, оператора выбора в языке Pascal. Понятие о приеме программирования “флаг”. Примеры алгоритмов и программ разветвляющейся структуры. Разработка схем алгоритмов разветвляющейся структуры. Написание, ввод, отладка и тестирование программ разветвляющейся структуры в СП Python и PascalABC
1	4	Разработка циклических алгоритмов и программ в соответствии с принципом структурности	Базовая алгоритмическая структура цикл с предусловием. Цикл с постусловием в языке Pascal. Проблема зацикливания. Цикл с параметром. Синтаксис и семантика операторов циклов в языках Python и Pascal. Особенности системной организации циклов с параметром в СП Python и PascalABC. Рекомендации по программированию циклов. Примеры циклических алгоритмов и программ. Разработка схем алгоритмов циклической структуры. Написание, ввод, отладка и тестирование программ циклической структуры в СП Python и PascalABC

1	5	Ввод с клавиатуры и вывод на монитор средствами алгритмических языков	Понятие о текстовом режиме монитора, текущее положение курсора. Синтаксис и семантика ввода с клавиатуры в СП Python и PascalABC. Особенности ввода с клавиатуры символьной и строковой информации в СП PascalABC. Синтаксис функции PRINT в Python. Организация форматированного вывода в Python: синтаксис, символы форматирования, основные форматы для вывода значений строкового и числового типа, семантика. Синтаксис и семантика оператора вывода на монитор в СП PascalABC. Форматированный вывод текстовой и числовой информации на монитор в СП PascalABC. Диалоговые программы. Дружественность интерфейса с пользователем. Примеры организации форматированного вывода на монитор. Разработка схем алгоритмов, написание, ввод, отладка и тестирование программ с форматированным выводом на монитор в СП Python и PascalABC
1	6	Структурированный тип данных массив	Массивы в программах на языке Python, примеры. Синтаксис определения типа массив в языке Pascal. Рекомендации по описанию переменной типа массив и понятие полной совместимости типов в языке Pascal, примеры. Элемент массива (переменная с индексами). Основные алгоритмы и программы обработки числовых массивов: поиск максимума (минимума), поиск элементов с заданными свойствами, сортировка методом пузырька, модифицированный алгоритм сортировки методом пузырька. Примеры алгоритмов и программ обработки массивов, в том числе с использованием приема “флаг”. Разработка схем алгоритмов обработки массивов. Написание, ввод, отладка и тестирование программ обработки массивов в СП Python и PascalABC

2 2. Перечень лабораторных работ

ЛР вводная. Знакомство с системами программирования Python и PascalABC. Ввод, отладка и тестирование простейших программ

Лабораторная работа №1. Написание, ввод, отладка и тестирование программ линейной структуры

Лабораторная работа №2. Написание, ввод, отладка и тестирование алгоритмов и программ разветвляющейся структуры

Лабораторная работа №3. Написание, ввод, отладка и тестирование разветвляющихся алгоритмов и программ с использованием структуры выбор

Лабораторная работа №4. Написание, ввод, отладка и тестирование циклических алгоритмов и программ

Лабораторная работа №5. Написание, ввод, отладка и тестирование циклических алгоритмов и программ целочисленной арифметики

Лабораторная работа №6. Написание, ввод, отладка и тестирование циклических алгоритмов и программ с форматированным выводом на монитор

Лабораторная работа №7. Написание, ввод, отладка и тестирование алгоритмов и программ обработки числовых массивов

3. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТА

Самостоятельная работа осуществляется в объеме 96 часов.
Видами СРС являются:

- изучение литературы и других источников;
- подготовка к выполнению лабораторной работы;
- подготовка к защите лабораторной работы.

Формами текущего контроля успеваемости являются:

- защита лабораторной работы.

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (см. фонд оценочных средств)

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1. Основная литература

№ п/п	Автор (ы), наименование, место издания и издательство, год
1	2
1.	Шелудько, В.М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули / В.М. Шелудько ; Министерство науки и высшего образования РФ, Южный федеральный университет, Институт компьютерных технологий и информационной безопасности. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 108 с. : ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=500060 (дата обращения: 30.08.2019)
2.	Комлев, Н.Ю. Самоучитель игры на Паскале. ABC и немного Турбо : практическое пособие / Н.Ю. Комлев. – Москва : СОЛОН-ПРЕСС, 2013. – 256 с. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=227109 (дата обращения: 30.08.2019)

5.2. Дополнительная литература

п/п	Автор (ы), наименование, место издания и издательство, год
1	*2*
1	Северенс, Ч. Введение в программирование на Python / Ч. Северенс. – 2-е изд., испр. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 231 с. : схем., ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=429184 (дата обращения: 30.08.2019)
2	Непейвода, Н. Н. Стили и методы программирования [Электронный ресурс] / Н. Н. Непейвода. - Москва : Интернет-Университет Информационных Технологий, 2005. - 320 с. - Режим доступа: http://biblioclub.ru/index.php?page=book&id=233198 (дата обращения: 30.08.2019).
3	Информатика и программирование. Алгоритмизация и программирование [Текст]: учебник для студ. учреждений высш. проф. образования/ [Н.И.Парфилова, А.В.Пруцков, А.Н.Пылькин, Б.Г.Трусов]; под ред. Б.Г.Трусова.-М.: Издательский центр «Академия», 2012.-336 с.

5.3. Базы данных, информационно-справочные и поисковые системы:

1. BOOR.ru [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://www.book.ru (дата обращения: 30.08.2019).
2. East View [Электронный ресурс] : [база данных]. – Доступ к полным текстам статей научных журналов из сети РГУ имени С.А. Есенина. – Режим доступа: http:/dlib.eastview.com (дата обращения: 30.08.2019).
3. Moodle [Электронный ресурс] : среда дистанционного обучения / Ряз. гос. ун-т. – Рязань, [Б.г.]. – Доступ, после регистрации из сети РГУ имени С.А. Есенина, из любой точки, имеющей доступ к Интернету. – Режим доступа: http://e-learn2.rsu.edu.ru/moodle2 (дата обращения: 30.08.2019).
4. Znanium.com [Электронный ресурс] : [база данных]. – Доступ к полным текстам по паролю. – Режим доступа: http://znanium.com (дата обращения: 30.08.2019).
5. «Издательство «Лань» [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://e-lanbook.com (дата обращения: 30.08.2019).

6. Университетская библиотека ONLINE [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://www.biblioclab.ru (дата обращения: 30.08.2019).

7. Юрайт [Электронный ресурс] : электронная библиотека. – Доступ к полным текстам по паролю. – Режим доступа: http://www.biblio-online.ru (дата обращения: 30.08.2019).

8. Труды преподавателей [Электронный ресурс] : коллекция // Электронная библиотека Научной библиотеки РГУ имени С.А. Есенина. – Доступ к полным текстам по паролю. – Режим доступа: http://dspace.rsu.edu.ru/xmlui/handle/123456789/3 (дата обращения: 30.08.2019).

5.4. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины

1. eLIBRARY.RU [Электронный ресурс] : научная электронная библиотека. – Режим доступа: http://elibrary.ru/defaultx.asp, свободный (дата обращения: 30.08.2019).

2. Единая коллекция цифровых образовательных ресурсов [Электронный ресурс] : федеральный портал. – Режим доступа: http://school-collection.edu.ru/, свободный (дата обращения: 30.08.2019).

3. Единое окно доступа к образовательным ресурсам [Электронный ресурс] : федеральный портал. – Режим доступа: http://window.edu.ru/, свободный (дата обращения: 30.08.2019).

4. Интернет Университет Информационных технологий. [Электронный ресурс] : сайт. – Режим доступа: http://www.intuit.ru/, свободный (дата обращения 30.08.2019).

5. Петров Д.Н. Парадигмы программирования. [Электронный ресурс] : сайт. – Режим доступа: http://dnpetrov.narod.ru/, свободный (дата обращения 30.08.2019).

6. Портал естественных наук. [Электронный ресурс] : сайт. – Режим доступа: http://e-science11.ru, свободный (дата обращения 30.08.2019).

7. Портал для программистов и администраторов информационных систем. [Электронный ресурс] : сайт. – Режим доступа: http://www.coderpost.net/, свободный (дата обращения 30.08.2019).

8. Российский общеобразовательный портал [Электронный ресурс] : образовательный портал. – Режим доступа: http://www.school.edu.ru/, свободный (дата обращения: 30.08.2019).

9. Сервер Информационных Технологий [Электронный ресурс] : сайт. – Режим доступа: http://citforum.ru/, свободный (дата обращения 30.08.2019).

10. Сайт программирования. [Электронный ресурс] : сайт. – Режим доступа: http://www.cyberguru.ru/, свободный (дата обращения 30.08.2019).

11. Федеральный центр информационно-образовательных ресурсов [Электронный ресурс] : Единое окно доступа к образовательным ресурсам. – Режим доступа: http://fcior.edu.ru, свободный (дата обращения: 30.08.2019).

5.5. Периодические издания

1. Компьютерные и информационные науки. Доступ: Киберленинка [Электронный ресурс] : научная электронная библиотека. – Режим доступа: https://cyberleninka.ru/article/c/computer-and-information-sciences, свободный (дата обращения: 30.08.2019).
2. Электротехника, электронная техника, информационные технологии. Доступ: Киберленинка [Электронный ресурс] : научная электронная библиотека. – Режим доступа: https://cyberleninka.ru/article/c/electrical-electronic-information-engineering, свободный (дата обращения: 30.08.2019).
3. Архив научных статей из журнала «Программирование» [Электронный ресурс]. – Режим доступа: naukarus.com/j/programmirovanie, свободный (дата обращения: 30.08.2019).

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Требования к аудиториям (помещениям, местам) для проведения занятий: специализированные лекционные аудитории, оборудованные видеопроекционным оборудованием для презентаций.

6.2. Требования к оборудованию рабочих мест преподавателя и обучающихся: видеопроектор, ноутбук, переносной экран, для проведения демонстраций, рабочие места обучающихся оснащены ПК с доступом в Интернет.

6.3. Требования к специализированному оборудованию отсутствуют

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Вид учебных занятий	Организация деятельности студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии. Уделить внимание следующим понятиям: *этапы решения задачи на компьютере; основные понятия алгоритмической системы; принцип структурности; алгоритмические структуры; вложение; структурный анализ; алфавит, синтаксис и семантика языка программирования; программа для компьютера; идентификатор; переменная; данные и их типы, операция, стандартная функция, выражение; простой и структурированный оператор*.

Лабораторная работа	В соответствии с запланированным на самостоятельную работу временем составить схемы алгоритмов и программы решения соответствующего варианта учебной задачи. Согласовать заранее составленные схемы и программы с преподавателем, ведущим занятие. Тексты программ должны содержать короткие комментарии, отражающие тему и номер лабораторной работы, номер варианта, фамилию студента, связь тех или иных переменных с условием задачи, а также комментарии, отражающие этапы решения задачи. Ввод с клавиатуры и вывод на монитор числовых данных должны сопровождаться краткими текстовыми сообщениями. Ввести тексты программ в компьютер, осуществить отладку и тестирование программ, при наличии нескольких программ решения одной и той же задачи добиться с учетом возможных погрешностей одинаковых результатов и продемонстрировать работоспособность программ преподавателю. Оформить отчет о лабораторной работе с указанием фамилии студента, номера лабораторной работы и номера варианта. Оформленная работа также должна содержать полный текст задания, схему алгоритма (кроме л/р № 1), тексты отлаженных на компьютере программ с комментариями по всем переменным. После текста программы на конкретном языке программирования должны быть отражены результаты ее выполнения из окна результатов в том виде, в котором они были выведены системой программирования. В случае ветвящихся алгоритмов и программ необходимо отразить несколько вариантов выполнения программы с различными наборами исходных данных, соответствующих различным ветвям исполнения алгоритма. Защитить оформленную лабораторную работу, продемонстрировав теоретические и практические знания, умения и навыки по соответствующей теме, возможные варианты схем решения задачи, структуры алгоритмов которых отличны от структур оформленных схем.
Подготовка к экзамену (зачету)	При подготовке к экзамену (зачету) необходимо ориентироваться на конспекты лекций, рекомендуемую литературу, типовые практические задания и др.

8. ТРЕБОВАНИЯ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ УЧЕБНОГО ПРОЦЕССА

1. Операционная система Windows Pro (договор №Tr000043844 от 22.09.15г)
2. Антивирус Kaspersky Endpoint Security (договор №02-ЗК-2019 от 15.04.2019г.)
3. Офисное приложение LibreOffice (свободно распространяемое ПО)
4. Система программирования Python (свободно распространяемое ПО)
5. Система программирования PascalABC (свободно распространяемое ПО)
6. Архиватор 7-zip (свободно распространяемое ПО);
7. Браузер изображений Fast Stone Image Viewer (свободно распространяемое ПО);
8. PDFридер Foxit Reader (свободно распространяемое ПО);
9. Медиа проигрыватель VLC mediaplayer (свободно распространяемое ПО);
10. Запись дисков ImageBurn (свободно распространяемое ПО);
11. DJVU браузер DjVu Browser Plug-in (свободно распространяемое ПО)
12. Запись дисков ImageBurn (свободно распространяемое ПО)
13. DJVU браузер DjVu Browser Plug-in (свободно распространяемое ПО