

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Кислова Наталья Николаевна
Должность: Проректор по УМР и качеству образования
Дата подписания: 29.04.2021 18:08:09
Уникальный программный ключ:
52802513f5b14a975b3e9b13008093d5726b159bf6064f865ae65b96a966c035

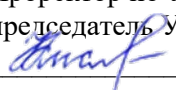
МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Самарский государственный социально-педагогический университет»

Кафедра информатики, прикладной математики и методики их преподавания

УТВЕРЖДАЮ

Проректор по УМР и КО,
председатель УМС СГСПУ
 Н.Н. Кислова

Программирование

рабочая программа дисциплины (модуля)

Закреплена за кафедрой **Информатики, прикладной математики и методики их преподавания**

Учебный план ФМФИ-61 7ПИЗ(5г)АБ.plx
Прикладная информатика

Квалификация **бакалавр**

Форма обучения **заочная**

Общая трудоемкость **28 ЗЕТ**

Часов по учебному плану 1008
в том числе: Виды контроля на курсах: в семестрах:
экзамены 1, 2

аудиторные занятия 112
самостоятельная работа 860
часов на контроль 36

Распределение часов дисциплины по семестрам

Курс	1		2		Итого	
	УП	РПД	УП	РПД	УП	РПД
Вид занятий						
Лекции	22	22	18	18	40	40
Лабораторные	34	34	30	30	64	64
Консультация перед экзаменом	4	4	4	4	8	8
В том числе инт.	20	20	16	16	36	36
Итого ауд.	60	60	52	52	112	112
Контактная работа	60	60	52	52	112	112
Сам. работа	462	462	398	398	860	860
Часы на контроль	18	18	18	18	36	36
Итого	540	540	468	468	1008	1008

Программу составил(и):

Пугач О.И.

При наличии обучающихся из числа лиц с ограниченными возможностями здоровья, которым необходим особый порядок освоения дисциплины (модуля), по их желанию разрабатывается адаптированная к ограничениям их здоровья рабочая программа дисциплины (модуля).

Рабочая программа дисциплины

Программирование

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА (уровень бакалавриата) (приказ Минобрнауки России от 12.03.2015г. №207)

составлена на основании учебного плана:

Прикладная информатика

утвержденного учёным советом вуза от 30.08.2016 протокол № 1.

Рабочая программа одобрена на заседании кафедры

Информатики, прикладной математики и методики их преподавания

Протокол от 28.08.2018 г. № 1

Зав. кафедрой Добудько Т.В.

Начальник УОП



Н.А. Доманина

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

Целью изучения дисциплины является изучение методов и приемов современного программирования и разработки приложений, формирование навыков разработки прототипов и законченных приложений с использованием современных языков объектно-ориентированного программирования (Java).

Задачи изучения дисциплины

– формирование готовности обучающихся работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия;

– развитие способности к самоорганизации и самообразованию в области профессиональной деятельности – проектировании, разработке и сопровождения информационных систем и приложений;

– формирование готовности обучающихся использовать нормативно-правовые документы, международные и отечественные стандарты в области информационных систем и технологий;

– формирование готовности обучающихся использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности.

Область профессиональной деятельности: системный анализ прикладной области, формализация решения прикладных задач и процессов информационных систем; разработка проектов автоматизации и информатизации прикладных процессов и создание информационных систем в прикладных областях; выполнение работ по созданию, модификации, внедрению и сопровождению информационных систем и управление этими работами.

Объектами профессиональной деятельности обучающихся, освоивших данную дисциплину, являются прикладные и информационные процессы, информационные технологии, информационные системы.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ОП: Б1.Б

2.1 Требования к предварительной подготовке обучающегося:

Содержание дисциплины базируется на материале:

Информатика (школьный курс)

2.2 Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

Программная инженерия

Производственная практика (практика по получению профессиональных умений и опыта профессиональной деятельности)

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

ОПК-1: способностью использовать нормативно-правовые документы, международные и отечественные стандарты в области информационных систем и технологий

Знать:

основные нормативно-правовые документы, международные и отечественные стандарты в области информационных систем и технологий

Уметь:

учитывать требования закона о защите персональных данных, отраслевые нормативы при разработке программного обеспечения.

Владеть:

ОК-6: способностью работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия

Знать:

соглашения и правила, общепринятые при разработке программного обеспечения на выбранном языке программирования (Java), принципы создания открытого программного обеспечения и взаимодействия разработчиков в данном процессе, социальные сети и другие ресурсы (github), обеспечивающие общение в сообществе программистов и правила общения на указанных ресурсах;

Уметь:

взаимодействовать с коллегами, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия, в рамках решения поставленной профессиональной задачи.

Владеть:

ОК-7: способностью к самоорганизации и самообразованию

Знать:

основные массовые открытые онлайн-курсы в области программирования, возможности отраслевого повышения квалификации и систему сертификации в ИТ-отрасли;

Уметь:

самостоятельно планировать собственную работу и работу малой группы при разработке программного обеспечения; обеспечивать самообразование в области профессиональной деятельности;

Владеть:

ОПК-3: способностью использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности

Знать:

Синтаксис языка и структуру платформы Java; основные понятия объектно-ориентированного программирования (классы, объекты, полиморфизм, инкапсуляция, наследование, интерфейсы), основные графические библиотеки (spring, awt), стандартные классы и интерфейсы коллекций (массивы, списки, очереди, множества, map); основные функции математических библиотек Java, способы обращения к ним, основные шаблоны (паттерны) проектирования.

Уметь:

составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ.

Владеть:

В результате освоения дисциплины (модуля) обучающийся должен

3.1 Знать:

соглашения и правила, общепринятые при разработке программного обеспечения на выбранном языке программирования (Java), принципы создания открытого программного обеспечения и взаимодействия разработчиков в данном процессе, социальные сети и другие ресурсы (github), обеспечивающие общение в сообществе программистов и правила общения на указанных ресурсах;

основные массовые открытые онлайн-курсы в области программирования, возможности отраслевого повышения квалификации и систему сертификации в ИТ-отрасли;

основные нормативно-правовые документы, международные и отечественные стандарты в области информационных систем и технологий

Синтаксис языка и структуру платформы Java; основные понятия объектно-ориентированного программирования (классы, объекты, полиморфизм, инкапсуляция, наследование, интерфейсы), основные графические библиотеки (spring, awt), стандартные классы и интерфейсы коллекций (массивы, списки, очереди, множества, map); основные функции математических библиотек Java, способы обращения к ним, основные шаблоны (паттерны) проектирования.

3.2 Уметь:

взаимодействовать с коллегами, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия, в рамках решения поставленной профессиональной задачи.

самостоятельно планировать собственную работу и работу малой группы при разработке программного обеспечения; обеспечивать самообразование в области профессиональной деятельности;

учитывать требования закона о защите персональных данных, отраслевые нормативы при разработке программного обеспечения.

составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ.

3.3 Владеть:

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Интеракт.
	Раздел 1. Основы языка Java			
1.1	Особенности языка и платформы Java. /Лек/	1	2	1
1.2	Особенности языка и платформы Java. /Лаб/	1	2	1
1.3	Особенности языка и платформы Java. /Ср/	1	30	0
1.4	Интегрированные среды разработки /Лек/	1	2	1
1.5	Интегрированные среды разработки /Лаб/	1	2	1
1.6	Интегрированные среды разработки /Ср/	1	30	0
1.7	Встроенные типы данных. /Лек/	1	2	1

1.8	Встроенные типы данных. /Лаб/	1	2	1
1.9	Встроенные типы данных. /Ср/	1	30	0
1.10	Константы и переменные. /Лек/	1	2	1
1.11	Константы и переменные. /Лаб/	1	2	1
1.12	Константы и переменные. /Ср/	1	40	0
1.13	Операторы ветвления. /Лек/	1	2	1
1.14	Операторы ветвления. /Лаб/	1	2	1
1.15	Операторы ветвления. /Ср/	1	30	0
1.16	Встроенный класс String. /Лек/	1	2	1
1.17	Встроенный класс String. /Лаб/	1	2	1
1.18	Встроенный класс String. /Ср/	1	40	0
1.19	Псевдослучайные числа. /Лек/	1	2	1
1.20	Псевдослучайные числа. /Лаб/	1	2	1
1.21	Псевдослучайные числа. /Ср/	1	40	0
1.22	Организация ввода и вывода данных. /Лек/	1	2	1
1.23	Организация ввода и вывода данных. /Лаб/	1	2	1
1.24	Организация ввода и вывода данных. /Ср/	1	45	0
1.25	Конструкции цикла в Java /Лек/	1	2	0
1.26	Конструкции цикла в Java /Лаб/	1	2	2
1.27	Конструкции цикла в Java /Ср/	1	45	0
1.28	Массивы. /Лек/	1	1	0
1.29	Массивы. /Лаб/	1	4	2
1.30	Массивы. /Ср/	1	45	0
1.31	Потоки данных и файлы /Лек/	1	1	0
1.32	Потоки данных и файлы /Лаб/	1	4	0
1.33	Потоки данных и файлы /Ср/	1	45	0
1.34	Исключительные ситуации /Лек/	1	1	0
1.35	Исключительные ситуации /Лаб/	1	4	0
1.36	Объекты и классы: основные понятия /Лек/	1	1	0
1.37	Объекты и классы: основные понятия /Лаб/	1	4	0
1.38	Объекты и классы: основные понятия /Ср/	1	46	0
	Раздел 2. Объектно-ориентированное программирование на Java			
2.1	Объекты и классы: реализация в языке Java /Лек/	2	1	0
2.2	Объекты и классы: реализация в языке Java /Лаб/	2	1	0
2.3	Объекты и классы: реализация в языке Java /Ср/	2	20	0
2.4	Средства управления жизнью объекта. /Лек/	2	1	0
2.5	Средства управления жизнью объекта. /Лаб/	2	1	0
2.6	Средства управления жизнью объекта. /Ср/	2	20	0
2.7	Работа с массивами объектов /Лек/	2	1	0
2.8	Работа с массивами объектов /Лаб/	2	1	0
2.9	Работа с массивами объектов /Ср/	2	20	0
2.10	Наследование в Java. /Лек/	2	1	0
2.11	Наследование в Java. /Лаб/	2	1	0
2.12	Наследование в Java. /Ср/	2	20	0
2.13	Абстрактные классы и интерфейсы /Лек/	2	1	0
2.14	Абстрактные классы и интерфейсы /Лаб/	2	1	0
2.15	Абстрактные классы и интерфейсы /Ср/	2	20	0
2.16	Пакеты. /Лек/	2	1	0
2.17	Пакеты. /Лаб/	2	1	0
2.18	Подсистема ввода вывода java.io. /Лек/	2	1	0
2.19	Подсистема ввода вывода java.io. /Лаб/	2	1	0
2.20	Подсистема ввода вывода java.io. /Ср/	2	20	0
2.21	Обработка исключений. /Лек/	2	1	0

2.22	Обработка исключений. /Лаб/	2	1	0
2.23	Обработка исключений. /Ср/	2	20	0
2.24	Классы Throwable и Exception: создание и использование. /Ср/	2	20	0
2.25	Обобщенное программирование (Java Generics). /Лек/	2	1	0
2.26	Обобщенное программирование (Java Generics). /Лаб/	2	1	0
2.27	Обобщенное программирование (Java Generics). /Ср/	2	20	0
2.28	Графическая подсистема Java. Классы AWT. /Лек/	2	1	0
2.29	Графическая подсистема Java. Классы AWT. /Лаб/	2	1	1
2.30	Графическая подсистема Java. Классы AWT. /Ср/	2	20	0
2.31	Обработка событий в графической подсистеме (Listeners). /Лаб/	2	1	1
2.32	Обработка событий в графической подсистеме (Listeners). /Ср/	2	20	0
2.33	Основные возможности графической системы Swing. /Лаб/	2	1	0
2.34	Основные возможности графической системы Swing. /Ср/	2	20	0
Раздел 3. Программы на Java в архитектуре клиент-сервер				
3.1	Многопоточное программирование на Java. /Лек/	2	1	1
3.2	Многопоточное программирование на Java. /Лаб/	2	1	1
3.3	Многопоточное программирование на Java. /Ср/	2	5	0
3.4	Класс Thread и интерфейс Runnable. /Лаб/	2	1	0
3.5	Класс Thread и интерфейс Runnable. /Ср/	2	5	0
3.6	Особенности написания многопоточных программ. /Лаб/	2	1	0
3.7	Особенности написания многопоточных программ. /Ср/	2	5	0
3.8	Организация пакетов стандартной библиотеки Java. /Лаб/	2	1	0
3.9	Организация пакетов стандартной библиотеки Java. /Ср/	2	5	0
3.10	Класс String и производительность в Java приложениях. /Лаб/	2	1	0
3.11	Класс String и производительность в Java приложениях. /Ср/	2	5	0
3.12	Перечисления Java(java.lang.Enum). /Лаб/	2	1	0
3.13	Перечисления Java(java.lang.Enum). /Ср/	2	5	0
3.14	Контейнеры и коллекции в Java. /Лаб/	2	1	0
3.15	Контейнеры и коллекции в Java. /Ср/	2	5	0
3.16	Итераторы /Лек/	2	1	1
3.17	Итераторы /Лаб/	2	1	1
3.18	Итераторы /Ср/	2	5	0
3.19	Технология коллективной разработки Java приложений. /Ср/	2	5	0
3.20	Автоматизация сборки и размещения Java приложений /Лаб/	2	1	1
3.21	Автоматизация сборки и размещения Java приложений /Ср/	2	5	0
3.22	Разработка прототипа и приложения (проект). /Ср/	2	5	0
3.23	Взаимодействие с базой данных /Лаб/	2	1	0
3.24	Взаимодействие с базой данных /Ср/	2	5	0
3.25	Оптимизация приложений /Ср/	2	5	0
Раздел 4. Разработка прототипов и приложений				
4.1	Апплеты. /Лек/	2	1	1
4.2	Апплеты. /Лаб/	2	1	1
4.3	Апплеты. /Ср/	2	5	0
4.4	Введение в программирование распределенных приложений. /Ср/	2	5	0
4.5	Интеграция с Web: Java сервлеты. /Лаб/	2	1	0
4.6	Интеграция с Web: Java сервлеты. /Ср/	2	5	0
4.7	Введение в ОС Android. /Лек/	2	1	1
4.8	Введение в ОС Android. /Лаб/	2	1	1
4.9	Введение в ОС Android. /Ср/	2	5	0
4.10	Основные View. /Лаб/	2	1	0
4.11	Основные View. /Ср/	2	10	0
4.12	Основные ViewGroup /Лаб/	2	1	0
4.13	Основные ViewGroup /Ср/	2	7	0

4.14	Стили и темы (практика). /Лек/	2	2	1
4.15	Стили и темы (практика). /Ср/	2	10	0
4.16	Списки в UI. /Ср/	2	10	0
4.17	Управление Activity. /Лаб/	2	1	1
4.18	Управление Activity. /Ср/	2	10	0
4.19	Взаимодействие с подсистемами хранения данных /Лаб/	2	1	1
4.20	Взаимодействие с подсистемами хранения данных /Ср/	2	10	0
4.21	Фреймворки Java для работы с многопоточностью в Android /Лек/	2	2	1
4.22	Фреймворки Java для работы с многопоточностью в Android /Лаб/	2	1	1
4.23	Фреймворки Java для работы с многопоточностью в Android /Ср/	2	6	0
4.24	Отправка сообщений (BroadcastReceiver). /Ср/	2	6	0
4.25	/Экзамен/	1	18	0
4.26	/Экзамен/	2	18	0
	Консультация перед экзаменом /КонсЭ/	1	4	0
	Консультация перед экзаменом /КонсЭ/	2	4	0

5. Оценочные и методические материалы по дисциплине (модулю)

5.1. Содержание аудиторной работы по дисциплине (модулю)

Лекции				
№	Тема лекции	часы	Содержание	Литература
1	Основы языка Java	28		
1.1	Особенности языка и платформы Java.	2	Особенности языка и платформы Java. Классификация программ по типу исполнения (компилируемые, интерпретируемые, исполняемые на виртуальных машинах). Виртуальная машина Java. JIT-компиляция. Создание простейшей программы на Java, её компиляция в байт-код и запуск.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.2	Интегрированные среды разработки.	2	Интегрированные среды разработки.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.3	Встроенные типы данных.	2	Встроенные типы данных. Способы задания литералов различных типов. Хранение данных в памяти ЭВМ. Приведение типов (явное и автоматическое).	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.4	Константы и переменные.	2	Константы и переменные. Оператор присваивания. Порядок действий (приоритет операторов). Арифметические операторы. Операторы инкремента и декремента.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.5	Операторы ветвления.	2	Операторы ветвления. Условный оператор. Минимизация количества проверок. Операторы ветвления. Оператор множественного выбора. Его сравнение с условным оператором.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.6	Встроенный класс String.	2	Встроенный класс String. Строковые операции. Работа со строками.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.7	Псевдослучайные числа.	2	Встроенный класс Math. Псевдослучайные числа.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

1.8	Организация ввода и вывода данных.	2	Организация ввода и вывода данных. Класс Scanner.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.9	Конструкции цикла в Java	2	Цикл типа «n раз». Цикл типа «пока» (с пред- и постпроверкой условия).	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.10	Массивы.	2	Массивы. Способы объявления и инициализации массивов. Индексация и размер массива. Массивы. Многомерные массивы. Алгоритмы сортировки. Алгоритмы поиска.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.11	Потоки данных и файлы.	2	Потоки данных и файлы. Чтение и запись в файл. Буферизованный ввод-вывод.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.12	Исключительные ситуации.	2	Исключительные ситуации. Обработка исключительных ситуаций.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.13	Объекты и классы: основные понятия	4	Объекты и классы – общие сведения. Классы как инструмент организации совместной работы. Инкапсуляция. Наследование. Управление наследованием. Интерфейсы как средство реализации множественного наследования. Полиморфизм. Средства реализации полиморфизма.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2	Объектно-ориентированное программирование на Java	28		
2.1	Объекты и классы: реализация в языке Java	2	Объекты и классы. Абстракция данных. Сценарий построения объектно-ориентированной программы. Члены классов. Методы и поля. Специальные методы классов (конструкторы). Конструктор по умолчанию. Модификаторы уровня доступа (default, public, protected, private). Статические методы классов. Методы функционального и процедурного типа. Процедурное программирование. Объектно ориентированное программирование. Сравнение парадигм. Сигнатура метода. Перегрузка методов. Инкапсуляция. Наследование. Управление наследованием. Интерфейсы как средство реализации множественного наследования. Полиморфизм. Средства реализации полиморфизма.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.2	Средства управления жизнью объекта.	2	Представление объектов и классов. Структура объявления класса. Доступ к членам класса. Спецификаторы доступа для обеспечения инкапсуляции. Знакомство с final. Средства управления жизнью объекта. Конструкторы и метод finalize(). Принципы работы сборщика мусора.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.3	Работа с массивами объектов.	2	Работа с массивами объектов. Статические поля и методы классов. Принцип работы ClassLoader. Реализация отношений между объектами и классами. Использование и зависимость от интерфейсов. Объекты при передаче параметров и возврате из методов. Внутренние классы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.4	Наследование в Java.	2	Средства объектно-ориентированного программирования Java. Наследование в Java. Производные классы. Класс Object. Метод toString(). Управление доступом в производных классах.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

2.5	Абстрактные классы и интерфейсы.	2	Абстрактные классы и интерфейсы. Реализация интерфейсов как альтернатива множественному наследованию. Информация о типе на этапе исполнения. Оператор instanceof. Приведение типов.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.6	Пакеты.	2	Пакеты. Определение пакета. Ограничение доступа. Импорт пакетов. Разрешение конфликтов имен.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.7	Подсистема ввода вывода java.io.	2	Подсистема ввода вывода java.io. Общие концепции организации ввода –вывода в библиотеке Java. Проблема платформенной независимости и локализации. Основные классы потоков ввода-вывода в Java и методы работы с ними. Использование потоков ввода вывода при работе с файлами. Эффективность.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.8	Обработка исключений.	2	Обработка исключений. Основные принципы и типы исключительных ситуаций. Перехват исключительных ситуаций. Операторы try, throw, throws, catch, finally. Использование нескольких блоков catch и вложенный оператор try. Не перехваченные исключительные ситуации.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.9	Классы Throwable и Exception: создание и использование.	2	Встроенные исключительные исключения Java. Классы Throwable и Exception. Принципы создания и использования исключительных ситуаций.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.10	Обобщенное программирование (Java Generics).	2	Родовые компоненты и обобщенное программирование. Java Generics. Ограничения на параметры. Совместимость на уровне байт-кода.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.11	Графическая подсистема Java. Классы AWT.	2	Графическая подсистема Java. Классы AWT. Основы работы с окнами . Component, Container, Panel, Window, Frame, Canvas.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.12	Обработка событий в графической подсистеме (Listeners).	2	Доставка и обработка событий в графической подсистеме. Механизм Listeners. Создание программы с оконным интерфейсом. Рисование графических примитивов. Использование управляющих элементов, диспетчеров компоновки и меню. Связь классов AWT с оконным интерфейсом операционной системы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.13	Основные возможности графической системы Swing.	4	Классы Swing. Основные принципы графической системы Swing. Платформенная независимость, понятие Look&Feel. Написание графического интерфейса с использованием Swing-компонентов. Введение в технологии разработки программного обеспечения.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3	Программы на Java в архитектуре клиент-сервер	40		
3.1	Многопоточное программирование на Java.	4	Многопоточное программирование на Java. Параллелизм. Модель потока в Java. Зависимость от реализации потока в операционной системе.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.2	Класс Thread и интерфейс Runnable.	2	Свойства потока. Синхронизация. Передача сообщений. Класс Thread и интерфейс Runnable. Главный поток и способы создания потоков. Управление потоками и приоритеты потоков. Группы потоков.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

3.3	Особенности написания многопоточных программ.	4	Особенности написания многопоточных программ. Использование синхронизирующих блоков и мониторов объектов. Синхронизированные методы объектов. Взаимная блокировка. Использование пула потоков.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.4	Организация пакетов стандартной библиотеки Java.	2	Стандартная библиотека Java. Организация пакетов стандартной библиотеки Java. Пакеты java и javax.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.5	Класс String и производительность в Java приложениях.	4	Использование и методы класса String. Класс String –краеугольный камень производительности в Java приложениях. Использование StringBuffer.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.6	Перечисления Java(java.lang.Enum).	2	Перечисления Java(java.lang.Enum). Особенности классов перечислений. Члены перечисления, поведение. Использование перечислений.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.7	Контейнеры и коллекции в Java.	4	Общие принципы организации контейнеров и коллекций в Java. Использование множеств и списков.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.8	Итераторы	2	Использование отображений и ассоциативных контейнеров. Итераторы и исключительные ситуации при работе с классами утилит.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.9	Технология коллективной разработки Java приложений.	4	Технология коллективной разработки Java приложений. Использование документирующих комментариев. Соглашения при написании кода и именовании классов и объектов.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.10	Автоматизация сборки и размещения приложений.	2	Автоматизация сборки и размещения Java приложений. Магазины.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.11	Разработка прототипа и приложения (проект).	4	Разработка прототипа и приложения (проект). Выбор инструментальных средств разработки. Использование git (системы версионного контроля) и баг-трекера. Документирование.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.12	Взаимодействие с базой данных.	2	Взаимодействие с базой данных. JSON. Транзакции. Виды запросов. Подключение базы данных в среде NetBeans. Разработка элементов информационной системы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.13	Оптимизация приложений.	4	Оптимизация, общие понятия. Оптимизация по времени и по памяти. Алгоритмы и подходы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4	Разработка прототипов и приложений	40		
4.1	Апплеты.	4	Апплеты. Основы работы с апплетом. Жизненный цикл апплета. Дескриптор APPLET. Передача параметров и загрузка апплета.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.2	Введение программирование распределенных	2	Введение в программирование распределенных приложений. Принципы построения распределенных приложений. Проблемы	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

	приложений.		передачи объектов и синхронизации в распределенных приложениях. Реализация сохраняемости.	javase/11/
4.3	Интеграция с Web: Java сервлеты.	4	Интеграция с Web. Java сервлеты. Java Server Pages. Web контейнеры.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.4	Введение в ОС Android.	2	Введение в ОС Android. Установка Android Sdk, эмуляторы для Android Android Studio. Gradle. Hello World! Структура проекта. Основные компоненты. Activity Service. BroadcastReceiver. ContentProvider. AndroidManifest. Ресурсы. Drawable. Values. Анимации. Темы и стили. Классификаторы. Локализация. Разработка пользовательского интерфейса. Activity и ее жизненный цикл.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.5	Основные View.	4	Основные View: TextView, Button, ImageView, CheckBox, ProgressBar и др.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.6	Основные ViewGroup	2	Основные ViewGroup: FrameLayout, LinearLayout, GridLayout, TableLayout, WebView and ChromeTabs	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.7	Стили и темы (практика).	4	Стили и темы (практика). Работа с меню Activity. Toolbar Диалоги и toast, snackbar	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.8	Списки в UI.	2	Списки в UI: ListView, GridView. Адаптеры, RecyclerView.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.9	Управление Activity.	4	Сохранение и восстановление состояния. Вызов новой Activity. Передача данных в Intent (практика). Хранение данных. SharedPreferences.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.10	Взаимодействие подсистемами хранения данных	2	Файлы. База данных Sqlite. Применение SQL. Фоновая работа. Processes and threads. AsyncTask. Loaders	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.11	Фреймворки Java для работы с многопоточностью в Android	4	Актуальные фреймворки для работы с многопоточностью. Service. Жизненный цикл. Запуск. Взаимодействие из Activity. Работа с сетью.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.12	Отправка сообщений (BroadcastReceiver).	2	BroadcastReceiver. Создание BroadcastReceiver. Подписка на системные события. Другие способы для отправки сообщений (EventBus).	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.13	Разработка прототипа и приложения (проект).	4	Разработка прототипа и приложения (проект): выбор инструментальных средств разработки, использование git (системы версионного контроля) и баг-трекера, документирование	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
Лабораторные работы				
№	Тема лекции	часы	Вопросы (тематика решаемых задач)	Литература
1	Основы языка Java	44		

1.1	Особенности языка и платформы Java.	4	Особенности языка и платформы Java. Классификация программ по типу исполнения (компилируемые, интерпретируемые, исполняемые на виртуальных машинах). Виртуальная машина Java. JIT-компиляция. Создание простейшей программы на Java, её компиляция в байт-код и запуск.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.2	Интегрированные среды разработки.	2	Интегрированные среды разработки.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.3	Встроенные типы данных.	4	Встроенные типы данных. Способы задания литералов различных типов. Хранение данных в памяти ЭВМ. Приведение типов (явное и автоматическое).	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.4	Константы и переменные.	2	Константы и переменные. Оператор присваивания. Порядок действий (приоритет операторов). Арифметические операторы. Операторы инкремента и декремента.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.5	Операторы ветвления.	4	Операторы ветвления. Условный оператор. Минимизация количества проверок. Операторы ветвления. Оператор множественного выбора, сравнение с условным оператором.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.6	Встроенный класс String.	2	Встроенный класс String. Строковые операции. Работа со строками.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.7	Псевдослучайные числа.	4	Встроенный класс Math. Псевдослучайные числа.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.8	Организация ввода и вывода данных.	2	Организация ввода и вывода данных. Класс Scanner.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.9	Конструкции цикла в Java	4	Цикл типа «n раз». Цикл типа «пока» (с пред- и постпроверкой условия).	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.10	Массивы.	6	Массивы. Способы объявления и инициализации массивов. Индексация и размер массива. Массивы. Многомерные массивы. Алгоритмы сортировки. Алгоритмы поиска.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.11	Потоки данных и файлы.	4	Потоки данных и файлы. Чтение и запись в файл. Буферизованный ввод-вывод.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.12	Исключительные ситуации.	2	Исключительные ситуации. Обработка исключительных ситуаций.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
1.13	Объекты и классы: основные	4	Объекты и классы – общие сведения.	https://docs.oracle.com/javase/

	понятия		Классы как инструмент организации совместной работы. Инкапсуляция. Наследование. Управление наследованием. Интерфейсы как средство реализации множественного наследования. Полиморфизм. Средства реализации полиморфизма.	tutorial/ https://docs.oracle.com/en/java/javase/11/
2	Объектно-ориентированное программирование на Java	44		
2.1	Объекты и классы: реализация в языке Java	4	Объекты и классы. Абстракция данных. Сценарий построения объектно-ориентированной программы. Члены классов. Методы и поля. Специальные методы классов (конструкторы). Конструктор по умолчанию. Модификаторы уровня доступа (default, public, protected, private). Статические методы классов. Методы функционального и процедурного типа. Процедурное программирование. Объектно ориентированное программирование. Сравнение парадигм. Сигнатура метода. Перегрузка методов. Инкапсуляция. Наследование. Управление наследованием. Интерфейсы как средство реализации множественного наследования. Полиморфизм. Средства реализации полиморфизма.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.2	Средства управления жизнью объекта.	2	Представление объектов и классов. Структура объявления класса. Доступ к членам класса. Спецификаторы доступа для обеспечения инкапсуляции. Знакомство с final. Средства управления жизнью объекта. Конструкторы и метод finalize(). Принципы работы сборщика мусора.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.3	Работа с массивами объектов.	4	Работа с массивами объектов. Статические поля и методы классов. Принцип работы ClassLoader. Реализация отношений между объектами и классами. Использование и зависимость от интерфейсов. Объекты при передаче параметров и возврате из методов. Внутренние классы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.4	Наследование в Java.	2	Средства объектно-ориентированного программирования Java. Наследование в Java. Производные классы. Класс Object. Метод toString(). Управление доступом в производных классах.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.5	Абстрактные классы и интерфейсы.	4	Абстрактные классы и интерфейсы. Реализация интерфейсов как альтернатива множественному наследованию. Информация о типе на этапе исполнения. Оператор instanceof. Приведение типов.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

2.6	Пакеты.	2	Пакеты. Определение пакета. Ограничение доступа. Импортирование пакетов. Разрешение конфликтов имен.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.7	Подсистема ввода вывода java.io.	4	Подсистема ввода вывода java.io. Общие концепции организации ввода –вывода в библиотеке Java. Проблема платформенной независимости и локализации. Основные классы потоков ввода-вывода в Java и методы работы с ними. Использование потоков ввода вывода при работе с файлами. Эффективность.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.8	Обработка исключений.	2	Обработка исключений. Основные принципы и типы исключительных ситуаций. Перехват исключительных ситуаций. Операторы try, throw, throws, catch, finally. Использование нескольких блоков catch и вложенный оператор try. Не перехваченные исключительные ситуации.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.9	Классы Throwable и Exception: создание и использование.	4	Встроенные исключительные исключения Java. Классы Throwable и Exception. Принципы создания и использования исключительных ситуаций.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.10	Обобщенное программирование (Java Generics).	2	Родовые компоненты и обобщенное программирование. Java Generics. Ограничения на параметры. Совместимость на уровне байт-кода.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.11	Графическая подсистема Java. Классы AWT.	4	Графическая подсистема Java. Классы AWT. Основы работы с окнами . Component, Container, Panel, Window, Frame, Canvas.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.12	Обработка событий в графической подсистеме (Listeners).	6	Доставка и обработка событий в графической подсистеме. Механизм Listeners. Создание программы с оконным интерфейсом. Рисование графических примитивов. Использование управляющих элементов, диспетчеров компоновки и меню. Связь классов AWT с оконным интерфейсом операционной системы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
2.13	Основные возможности графической системы Swing.	4	Классы Swing. Основные принципы графической системы Swing. Платформенная независимость, понятие Look&Feel. Написание графического интерфейса с использованием Swing компонентов. Введение в технологии разработки программного обеспечения.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3	Программы на Java в архитектуре клиент-сервер	60		
3.1	Многопоточное программирование на Java.	4	Многопоточное программирование на Java. Параллелизм. Модель потока в Java. Зависимость от реализации потока в операционной системе.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.2	Класс Thread и интерфейс Runnable.	4	Свойства потока. Синхронизация. Передача сообщений..Класс Thread и интерфейс Runnable. Главный поток и способы создания потоков. Управление потоками и приоритеты потоков.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

			Группы потоков.	
3.3	Особенности написания многопоточных программ.	4	Особенности написания многопоточных программ. Использование синхронизирующих блоков и мониторов объектов. Синхронизированные методы объектов. Взаимная блокировка. Использование пула потоков.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.4	Организация пакетов стандартной библиотеки Java.	4	Стандартная библиотека Java. Организация пакетов стандартной библиотеки Java. Пакеты java и javax.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.5	Класс String и производительность в Java приложениях.	4	Использование и методы класса String. Класс String –краеугольный камень производительности в Java приложениях. Использование StringBuffer.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.6	Перечисления Java(java.lang.Enum).	4	Перечисления Java(java.lang.Enum). Особенности классов перечислений. Члены перечисления, поведение. Использование перечислений.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.7	Контейнеры и коллекции в Java.	4	Общие принципы организации контейнеров и коллекций в Java. Использование множеств и списков.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.8	Итераторы	4	Использование отображений и ассоциативных контейнеров. Итераторы и исключительные ситуации при работе с классами утилит.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.9	Технология коллективной разработки приложений.	4	Технология коллективной разработки Java приложений. Использование документирующих комментариев. Соглашения при написании кода и именовании классов и объектов.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.10	Автоматизация сборки и размещения приложений.	4	Автоматизация сборки и размещения Java приложений. Магазины.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.11	Разработка прототипа и приложения (проект).	4	Разработка прототипа и приложения (проект). Выбор инструментальных средств разработки. Использование git (системы версионного контроля) и баг-трекера. Документирование.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.12	Взаимодействие с базой данных.	12	Взаимодействие с базой данных. JSON. Транзакции. Виды запросов. Подключение базы данных в среде NetBeans. Разработка элементов информационной системы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
3.13	Оптимизация приложений.	4	Оптимизация, общие понятия. Оптимизация по времени и по памяти. Алгоритмы и подходы.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4	Разработка прототипов и приложений	60		
4.1	Апплеты.	4	Апплеты. Основы работы с апплетом. Жизненный цикл апплета. Дескриптор APPLETON. Передача параметров и загрузка апплета.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

4.2	Введение программирование распределенных приложений.	в	4	Введение в программирование распределенных приложений. Принципы построения распределенных приложений. Проблемы передачи объектов и синхронизации в распределенных приложениях. Реализация сохраняемости.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.3	Интеграция с Web: Java сервлеты.		4	Интеграция с Web. Java сервлеты. Java Server Pages. Web контейнеры.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.4	Введение в ОС Android.		4	Введение в ОС Android. Установка Android Sdk, эмуляторы для Android Android Studio. Gradle. Hello World! Структура проекта. Основные компоненты. Activity Service. BroadcastReceiver. ContentProvider. AndroidManifest. Ресурсы. Drawable. Values. Анимации. Темы и стили. Классификаторы. Локализация. Разработка пользовательского интерфейса. Activity и ее жизненный цикл.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.5	Основные View.		4	Основные View: TextView, Button, ImageView, CheckBox, ProgressBar и др.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.6	Основные ViewGroup		4	Основные ViewGroup: FrameLayout, LinearLayout, GridLayout, TableLayout, WebView and ChromeTabs	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.7	Стили и темы (практика).		4	Стили и темы (практика). Работа с меню Activity. ToolBar Диалоги и toast, snackbar	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.8	Списки в UI.		4	Списки в UI: ListView, GridView. Адаптеры, RecyclerView.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.9	Управление Activity.		4	Сохранение и восстановление состояния. Вызов новой Activity. Передача данных в Intent (практика). Хранение данных. SharedPreferences.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.10	Взаимодействие подсистемами хранения данных	с	4	Файлы. База данных SQLite. Применение SQL. Фоновая работа. Processes and threads. AsyncTask. Loaders	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.11	Фреймворки Java для работы с многопоточностью в Android	в	4	Актуальные фреймворки для работы с многопоточностью. Service. Жизненный цикл. Запуск.	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

			Взаимодействие из Activity. Работа с сетью.	
4.12	Отправка сообщений (BroadcastReceiver).	4	BroadcastReceiver. Создание BroadcastReceiver. Подписка на системные события. Другие способы для отправки сообщений (EventBus).	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/
4.13	Разработка прототипа и приложения (проект).	12	Разработка прототипа и приложения (проект); выбор инструментальных средств разработки, использование git (системы версионного контроля) и баг-трекера, документирование	https://docs.oracle.com/javase/tutorial/ https://docs.oracle.com/en/java/javase/11/

5.2. Содержание самостоятельной работы по дисциплине (модулю)

№ п/п	Темы дисциплины	Содержание самостоятельной работы студентов	Продукты деятельности
1	Основы языка Java	Индивидуальное домашнее задание	Отчет (работающее приложение, UML, документация)
2	Объектно-ориентированное программирование на Java	Индивидуальное домашнее задание	Отчет (работающее приложение, UML, документация)
3	Программы на Java в архитектуре клиент-сервер	Индивидуальное домашнее задание	Отчет (работающее приложение, UML, документация)
4	Разработка прототипов и приложений	Индивидуальное домашнее задание	Отчет (работающее приложение, UML, документация)

Содержание самостоятельной работы по дисциплине на выбор студента

№ п/п	Темы дисциплины	Содержание самостоятельной работы студентов	Продукты деятельности
1	Основы языка Java	Решение задач повышенной сложности	Программа
2	Объектно-ориентированное программирование на Java	Решение задач повышенной сложности	Программа
3	Программы на Java в архитектуре клиент-сервер	Решение задач повышенной сложности	Программа
4	Разработка прототипов и приложений	Решение задач повышенной сложности	Программа

5.3. Образовательные технологии

При организации изучения дисциплины будут использованы следующие образовательные технологии: информационно-коммуникационные технологии, технология организации самостоятельной работы, технология рефлексивного обучения, технология модульного обучения, технология игрового обучения, технологии групповой дискуссии, интерактивные технологии, технология проблемного обучения, технология организации учебно-исследовательской деятельности, технология проектного обучения, технология развития критического мышления.

5.4. Текущий контроль, промежуточный контроль и промежуточная аттестация

Балльно-рейтинговая карта дисциплины оформлена как приложение к рабочей программе дисциплины. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине оформлен отдельным документом.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год
Л1.1	Николаев, Е.И.	Объектно-ориентированное программирование: учебное пособие http://biblioclub.ru/index.php?page=book&id=458133	Ставрополь : СКФУ, 2015,
Л1.2	Сухомлин, В.А.	Введение в программирование : учебное пособие http://biblioclub.ru/index.php?page=book&id=232982	Москва : Интернет-Университет Информационных Технологий, 2007,

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год
Л2.1	Комлева, Н.В.	Методы программирования: Учебно-методический комплекс http://biblioclub.ru/index.php?page=book&id=90390	М.: Евразийский открытый институт, 2011,

Л2.2	Москвитин, А.А.	Решение задач на компьютерах: учебное пособие http://biblioclub.ru/index.php?page=book&id=273666	М.; Берлин : Директ-Медиа, 2015,
Л2.3	Москвитин, А.А.	Решение задач на компьютерах: учебное пособие http://biblioclub.ru/index.php?page=book&id=273667	М.; Берлин: Директ-Медиа, 2015,
Л2.4	Е.И. Николаев	Объектно-ориентированное программирование : лабораторный практикум http://biblioclub.ru/index.php?page=book&id=458134 .	Ставрополь : СКФУ, 2015,
Л2.5	Е.И. Николаев	Объектно-ориентированное программирование : лабораторный практикум : в 2 ч. / http://biblioclub.ru/index.php?page=book&id=458135	«Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2015. - Ч. 2

6.2 Перечень программного обеспечения

- Microsoft Office 2016 Professional Plus (Пакет программ Word, Excel, Access, PowerPoint, Outlook, OneNote, Publisher)

6.3 Перечень информационных справочных систем

- ЭБС «E-LIBRARY.RU»

- ЭБС «Университетская библиотека онлайн»

- СПС «ГАРАНТ-Аналитик»

- СПС «Консультант-Плюс»

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1	Наименование специального помещения: учебная аудитория для проведения занятий лекционного типа, лабораторных занятий, групповых консультаций, индивидуальных консультаций, текущего контроля, промежуточной аттестации, помещение для самостоятельной работы. Оснащенность: Комплект учебной мебели, меловая доска, ноутбук, стационарное проекционное оборудование (мультимедийный проектор с потолочным креплением и настенный экран), портативное звукоусиливающее оборудование, ПК - 16 шт.
-----	---

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Методические рекомендации по организации изучения дисциплины для студентов и преподавателей

С целью формирования и развития профессиональных навыков обучающихся, в соответствии с требованиями ФГОС ВО по направлению подготовки «Прикладная информатика», для реализации компетентностного подхода программа предусматривает широкое использование в учебном процессе следующих форм учебной работы: лекции, лабораторные работы и самостоятельная работа.

Лекция – одна из основных форм организации учебного процесса, представляющая собой устное, монологическое, систематическое, последовательное изложение преподавателем учебного материала. Она предшествует всем другим формам организации учебного процесса, позволяет оперативно актуализировать учебный материал дисциплины. На лекциях раскрываются основные понятия курса, приводятся примеры решения задач, отмечаются современные подходы к решаемым проблемам. Продуктом деятельности студента на лекции является опорный конспект.

Во время лабораторных занятий необходимо овладеть методами и приемами решения практических задач. Для выполнения лабораторных работ используются персональные компьютеры с установленным на них необходимым программным обеспечением, имеющие выход в Интернет.

В процессе выполнения лабораторных работ студенты должны научиться:

- работать в информационной образовательной среде СГСПУ;
- искать необходимую информацию на сайте СГСПУ;
- создавать презентации в соответствии с заданными требованиями в MS PowerPoint;
- изучить возможность облачных технологий для совместной работы;
- организовывать поиск информации в различных информационно-поисковых и справочно-правовых системах;
- работать с электронными библиотечными системами.

Каждая лабораторная работа снабжена подробными инструкциями по выполнению и содержит задания для обязательного выполнения. За выполненные задания учащиеся получают баллы в соответствии с балльно-рейтинговой картой.

Некоторые лабораторные работы содержат индивидуальные задания для самостоятельного выполнения.

Самостоятельная работа является одним из основных видов учебной работы, оказывающих значительное влияние на глубину и прочность знаний, на развитие познавательных способностей, на темп усвоения нового материала и формирование навыков самообразования. В основе самостоятельной работы лежит выполнение индивидуальных заданий из лабораторных работ. В качестве самостоятельной работы студентам предлагаются следующие задания:

Составление реферативного обзора, выражающегося в подборе и систематизации статей, монографий и диссертаций по теме магистерского исследования. Развивает у студентов способность выделять ключевые понятия темы и использовать их в Интернет-поиске. Студент должен прочитать материал источника, сформулировать идею, выделить необходимые цитаты; оформить работу и представить в установленный срок.

Написать статью, тематика которой должна быть актуальной, затрагивающей современные проблемы области изучения дисциплины. Студент должен раскрыть не только суть проблемы, привести различные точки зрения, но и выразить собственные взгляды на неё. Этот вид работы требует от студента умения чётко выражать мысли как в письменной форме, так и посредством логических рассуждений, ясно излагать свою точку зрения. Затраты времени на подготовку материала зависят от трудности сбора информации, сложности материала по теме, индивидуальных особенностей студента и определяются преподавателем.

Формирование отчета по лабораторной работе. Отчет представляет собой оформленные в MS Word задания (решения задач) и сформулированные выводы. Этот вид работы требует от студента внимательности, умения чётко выражать свои мысли.

Среди различных источников новых знаний основное занимает книга. Для изучения дисциплины предлагается список основной и дополнительной литературы. При подготовке к занятиям возможно широкое использование образовательных ресурсов сети Интернет.

В конце семестра предусмотрен экзамен. Все баллы, набранные студентом, суммируются (БРК в приложении), и на их основании выставляется экзамен.

Балльно-рейтинговая карта дисциплины Программирование

Курс 1 Семестр 1

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
1 семестр			
Наименование модуля «Основы языка Java»			
Текущий контроль по модулю:			
1	Аудиторная работа	13	26
2	Самостоятельная работа (специальные обязательные формы)	5	10
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по модулю		–	–
Промежуточный контроль		20	40
Промежуточная аттестация		36	60
Итого		56	100

...

Вид контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
1 семестр		
Текущий контроль по модулю «Основы языка Java»		
Аудиторная работа	Лабораторная работа №5 «Операторы ветвления» Написать программу, определяющую, есть ли в записи введенного 3-х значного числе хотя бы один 0 Решение задач	Темы: Особенности языка и платформы Java. Интегрированные среды разработки. Встроенные типы данных.

	<p>Критерии оценивания:</p> <ul style="list-style-type: none"> • не решал задачи или решил неправильно – 0 баллов; • задачи решены с несущественными ошибками – 1 балл; • задачи решены без ошибок – 2 балла. <p>Итого – 13x2=26 баллов</p>	<p>Константы и переменные.</p> <p>Операторы ветвления.</p> <p>Встроенный класс String.</p> <p>Псевдослучайные числа.</p> <p>Организация ввода и вывода данных.</p> <p>Конструкции цикла в Java</p> <p>Массивы.</p> <p>Потоки данных и файлы.</p> <p>Исключительные ситуации.</p> <p>Объекты и классы: основные понятия Образовательные результаты:</p> <p>Умеет:</p> <p>составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ</p>
Самостоятельная работа (обяз.)	Индивидуальное домашнее задание (ИДЗ) (x2)	Темы:

	<ul style="list-style-type: none"> • решены все задачи ИДЗ – 3 балла; • решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл; • отчет представлен преподавателю (загружен на проверку в систему управления обучением) в установленные сроки – 1 балл. <p>Итого – 5x2=10 баллов</p>	<p>Особенности языка и платформы Java.</p> <p>Интегрированные среды разработки.</p> <p>Встроенные типы данных.</p> <p>Константы и переменные.</p> <p>Операторы ветвления.</p> <p>Встроенный класс String.</p> <p>Псевдослучайные числа.</p> <p>Организация ввода и вывода данных.</p> <p>Конструкции цикла в Java</p> <p>Массивы.</p> <p>Потоки данных и файлы.</p> <p>Исключительные ситуации.</p> <p>Объекты и классы: основные понятия</p> <p>Образовательные результаты:</p> <p>Умеет:</p> <p>составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в</p>
--	---	---

		компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ
Самостоятельная работа (на выбор)	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> • решены все задачи – 3 балла; • решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл. <p>Итого – 4 балла</p>	<p>Темы:</p> <p>Особенности языка и платформы Java.</p> <p>Интегрированные среды разработки.</p> <p>Встроенные типы данных.</p> <p>Константы и переменные.</p> <p>Операторы ветвления.</p> <p>Встроенный класс String.</p> <p>Псевдослучайные числа.</p> <p>Организация ввода и вывода данных.</p> <p>Конструкции цикла в Java</p> <p>Массивы.</p> <p>Потоки данных и файлы.</p> <p>Исключительные ситуации.</p> <p>Объекты и классы: основные понятия</p> <p>Образовательные результаты:</p> <p>Умеет:</p> <p>составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для</p>

		решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ
Контрольное мероприятие по модулю	-	
Промежуточный контроль (кол-во баллов)	Минимальное количество баллов – 20, максимальное – 40	
Промежуточная аттестация	Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	

Курс _1__ Семестр 2

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
2 семестр			
Наименование модуля «Объектно-ориентированное программирование на Java»			
Текущий контроль по модулю:			
1	Аудиторная работа	13	26
2	Самостоятельная работа (специальные обязательные формы)	5	10
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по модулю		–	–
Промежуточный контроль		20	40
Промежуточная аттестация		36	60
Итого		56	100

...

Вид контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
2 семестр		
Текущий контроль по модулю «Объектно-ориентированное программирование на Java»		
Аудиторная работа	<p>Лабораторная работа №1 «Объекты и классы»</p> <p>Записать объект, моделирующий свойства и методы сотрудника организации. Составить UML-диаграмму.</p> <p>Решение задач</p> <p>Критерии оценивания:</p> <ul style="list-style-type: none"> • не решал задачи или решил неправильно – 0 баллов; 	<p>Темы:</p> <p>Объекты и классы: реализация в языке Java</p> <p>Средства управления жизнью объекта.</p> <p>Работа с массивами объектов.</p> <p>Наследование в Java.</p>

	<ul style="list-style-type: none">• задачи решены с несущественными ошибками – 1 балл;• задачи решены без ошибок – 2 балла. <p>Итого – 13x2=26 баллов</p>	<p>Абстрактные классы и интерфейсы.</p> <p>Пакеты.</p> <p>Подсистема ввода вывода java.io.</p> <p>Обработка исключений.</p> <p>Классы Throwable и Exception: создание и использование.</p> <p>Обобщенное программирование (Java Generics).</p> <p>Графическая подсистема Java. Классы AWT.</p> <p>Обработка событий в графической подсистеме (Listeners).</p> <p>Основные возможности графической системы Swing.</p> <p>Образовательные результаты:</p> <p>Умеет:</p> <p>составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных</p>
--	--	--

		программ
Самостоятельная работа (обяз.)	<p>Индивидуальное домашнее задание (ИДЗ) (x2)</p> <ul style="list-style-type: none"> • решены все задачи ИДЗ – 3 балла; • решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл; • отчет представлен преподавателю (загружен на проверку в систему управления обучением) в установленные сроки – 1 балл. <p>Итого – 5x2=10 баллов</p>	<p>Темы:</p> <p>Объекты и классы: реализация в языке Java</p> <p>Средства управления жизнью объекта.</p> <p>Работа с массивами объектов.</p> <p>Наследование в Java.</p> <p>Абстрактные классы и интерфейсы.</p> <p>Пакеты.</p> <p>Подсистема ввода вывода java.io.</p> <p>Обработка исключений.</p> <p>Классы Throwable и Exception: создание и использование.</p> <p>Обобщенное программирование (Java Generics).</p> <p>Графическая подсистема Java. Классы AWT.</p> <p>Обработка событий в графической подсистеме (Listeners).</p> <p>Основные возможности графической системы Swing.</p> <p>Образовательные результаты:</p> <p>Умеет</p> <p>составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных</p>

		<p>средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ</p>
<p>Самостоятельная работа (на выбор)</p>	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> • решены все задачи – 3 балла; • решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл. <p>Итого – 4 балла</p>	<p>Темы:</p> <p>Объекты и классы: реализация в языке Java</p> <p>Средства управления жизнью объекта.</p> <p>Работа с массивами объектов.</p> <p>Наследование в Java.</p> <p>Абстрактные классы и интерфейсы.</p> <p>Пакеты.</p> <p>Подсистема ввода вывода java.io.</p> <p>Обработка исключений.</p> <p>Классы Throwable и Exception: создание и использование.</p> <p>Обобщенное программирование (Java Generics).</p> <p>Графическая подсистема Java. Классы AWT.</p> <p>Обработка событий в графической</p>

		<p>подсистеме (Listeners).</p> <p>Основные возможности графической системы Swing.</p> <p>Образовательные результаты:</p> <p>Умеет</p> <p>составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ</p>
Контрольное мероприятие по модулю	-	
Промежуточный контроль (кол-во баллов)	Минимальное количество баллов – 20, максимальное – 40	
Промежуточная аттестация	Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	

Курс 2 Семестр 3

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
3 семестр			
Наименование модуля «Программы на Java в архитектуре клиент-сервер»			
Текущий контроль по модулю:			
1	Аудиторная работа	13	26
2	Самостоятельная работа (специальные обязательные формы)	5	10
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по модулю		–	–
Промежуточный контроль		20	40
Промежуточная аттестация		36	60
Итого		56	100

...

Вид контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
3 семестр		
Текущий контроль по модулю «Программы на Java в архитектуре клиент-сервер»		
Аудиторная работа	<p>Практическое занятие №6 «Класс String и производительность в Java приложениях.»</p> <p>Вывести гласные слова S, встречающихся более одного раза. Решить задачу с и без использования StringBuilder.</p> <p>Решение задач</p> <p>Критерии оценивания:</p> <ul style="list-style-type: none"> • не решал задачи или решил неправильно – 0 баллов; 	<p>Темы:</p> <p>Многопоточное программирование на Java.</p> <p>Класс Thread и интерфейс Runnable.</p> <p>Особенности написания многопоточных программ.</p> <p>Организация пакетов стандартной</p>

	<ul style="list-style-type: none"> • задачи решены с несущественными ошибками – 1 балл; • задачи решены без ошибок – 2 балла. <p>Итого – 13x2=26 баллов</p>	<p>библиотеки Java.</p> <p>Класс String и производительность в Java приложениях.</p> <p>Перечисления Java(java.lang.Enum).</p> <p>Контейнеры и коллекции в Java.</p> <p>Итераторы</p> <p>Технология коллективной разработки Java приложений.</p> <p>Автоматизация сборки и размещения Java приложений.</p> <p>Разработка прототипа и приложения (проект).</p> <p>Взаимодействие с базой данных.</p> <p>Оптимизация приложений.</p> <p>Образовательные результаты:</p> <p>Умеет:</p> <p>самостоятельно планировать собственную работу и работу малой группы при разработке программного обеспечения; обеспечивать самообразование в области профессиональной деятельности</p>
<p>Самостоятельная работа (обяз.)</p>	<p>Индивидуальное домашнее задание (ИДЗ) (x2)</p> <ul style="list-style-type: none"> • решены все задачи ИДЗ – 3 балла; • решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл; • отчет представлен преподавателю (загружен на проверку в систему управления обучением) в установленные сроки – 1 балл. 	<p>Темы:</p> <p>Многопоточное программирование на Java.</p> <p>Класс Thread и интерфейс Runnable.</p> <p>Особенности написания многопоточных</p>

Итого – 5x2=10 баллов

программ.

Организация пакетов стандартной библиотеки Java.

Класс String и производительность в Java приложениях.

Перечисления Java(java.lang.Enum).

Контейнеры и коллекции в Java.

Итераторы

Технология коллективной разработки Java приложений.

Автоматизация сборки и размещения Java приложений.

Разработка прототипа и приложения (проект).

Взаимодействие с базой данных.

Оптимизация приложений.

Образовательные результаты:

Умеет:

составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей

		(функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных программ
Самостоятельная работа (на выбор)	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> • решены все задачи – 3 балла; • решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл. <p>Итого – 4 балла</p>	<p>Темы:</p> <p>Многопоточное программирование на Java.</p> <p>Класс Thread и интерфейс Runnable.</p> <p>Особенности написания многопоточных программ.</p> <p>Организация пакетов стандартной библиотеки Java.</p> <p>Класс String и производительность в Java приложениях.</p> <p>Перечисления Java(java.lang.Enum).</p> <p>Контейнеры и коллекции в Java.</p> <p>Итераторы</p> <p>Технология коллективной разработки Java приложений.</p> <p>Автоматизация сборки и размещения Java приложений.</p> <p>Разработка прототипа и приложения (проект).</p> <p>Взаимодействие с базой данных.</p> <p>Оптимизация приложений.</p> <p>Образовательные результаты:</p>

		<p>Умеет:</p> <p>самостоятельно планировать собственную работу и работу малой группы при разработке программного обеспечения; обеспечивать самообразование в области профессиональной деятельности</p>
Контрольное мероприятие по модулю	-	
Промежуточный контроль (кол-во баллов)	Минимальное количество баллов – 20, максимальное – 40	
Промежуточная аттестация	Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	

Курс __2_ Семестр 4

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
4 семестр			
Наименование модуля «Разработка прототипов и приложений»			
Текущий контроль по модулю:			
1	Аудиторная работа	13	26
2	Самостоятельная работа (специальные обязательные формы)	5	10
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по модулю		–	–
Промежуточный контроль		20	40
Промежуточная аттестация		36	60
Итого		56	100

...

Вид контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
4 семестр		
Текущий контроль по модулю «Разработка прототипов и приложений»		
Аудиторная работа	<p>Практическое занятие №4 «Введение в ОС Android.»</p> <p>Пример задания</p> <p>Установить Android SDK и необходимые эмуляторы. Составить отчет.</p> <p>Решение задач</p> <p>Критерии оценивания:</p> <ul style="list-style-type: none"> • не решал задачи или решил неправильно – 0 баллов; 	<p>Темы:</p> <p>Апплеты.</p> <p>Введение в программирование распределенных приложений.</p> <p>Интеграция с Web: Java сервлеты.</p> <p>Введение в ОС Android.</p>

	<ul style="list-style-type: none"> • задачи решены с несущественными ошибками – 1 балл; • задачи решены без ошибок – 2 балла. <p>Итого – 13x2=26 баллов</p>	<p>Основные View.</p> <p>Основные ViewGroup</p> <p>Стили и темы (практика).</p> <p>Списки в UI.</p> <p>Управление Activity.</p> <p>Взаимодействие с подсистемами хранения данных</p> <p>Фреймворки Java для работы с многопоточностью в Android</p> <p>Отправка сообщений (BroadcastReceiver).</p> <p>Разработка прототипа и приложения (проект).</p> <p>Образовательные результаты:</p> <p>Умеет:</p> <p>самостоятельно планировать собственную работу и работу малой группы при разработке программного обеспечения; обеспечивать самообразование в области профессиональной деятельности</p>
<p>Самостоятельная работа (обяз.)</p>	<p>Индивидуальное домашнее задание (ИДЗ) (x2)</p> <p><пример задания>.</p> <ul style="list-style-type: none"> • решены все задачи ИДЗ – 3 балла; • решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл; • отчет представлен преподавателю (загружен на проверку в систему управления обучением) в установленные сроки – 1 балл. 	<p>Темы:</p> <p>Апплеты.</p> <p>Введение в программирование распределенных приложений.</p> <p>Интеграция с Web: Java сервлеты.</p> <p>Введение в ОС Android.</p>

Итого – 5x2=10 баллов

Основные View.

Основные ViewGroup

Стили и темы (практика).

Списки в UI.

Управление Activity.

Взаимодействие с подсистемами хранения данных

Фреймворки Java для работы с многопоточностью в Android

Отправка сообщений (BroadcastReceiver).

Разработка прототипа и приложения (проект).

Образовательные результаты:

Умеет:

составлять простые программы с использованием консольного вывода и операторов условия и цикла; организовывать ввод и вывод данных средствами графического интерфейса; создавать собственную иерархию классов и интерфейсов (и представлять ее с помощью UML-диаграмм) для решения конкретных прикладных задач; проектировать и создавать несложные мобильные и веб-приложения; выполнять преобразования простых математических моделей (функциональных, статистических) в компьютерные с применением языка программирования Java; создавать прототипы-интерфейсы для прикладных

		программ
Самостоятельная работа (на выбор)	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> решены все задачи – 3 балла; решения задач с иллюстрациями оформлены развернуто, в соответствии с требованиями преподавателя – 1 балл. <p>Итого – 4 балла</p>	<p>Темы:</p> <p>Апплеты.</p> <p>Введение в программирование распределенных приложений.</p> <p>Интеграция с Web: Java сервлеты.</p> <p>Введение в ОС Android.</p> <p>Основные View.</p> <p>Основные ViewGroup</p> <p>Стили и темы (практика).</p> <p>Списки в UI.</p> <p>Управление Activity.</p> <p>Взаимодействие с подсистемами хранения данных</p> <p>Фреймворки Java для работы с многопоточностью в Android</p> <p>Отправка сообщений (BroadcastReceiver).</p> <p>Разработка прототипа и приложения (проект).</p> <p>Образовательные результаты:</p> <p>Умеет:</p> <p>самостоятельно планировать собственную работу и работу малой группы при разработке программного обеспечения; обеспечивать самообразование в области</p>

		профессиональной деятельности
Контрольное мероприятие по модулю	-	
Промежуточный контроль (кол-во баллов)	Минимальное количество баллов – 20, максимальное – 40	
Промежуточная аттестация	Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	