

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО

Декан ФИТ НГУ

М.М. Лаврентьев

«03» июля 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Объектно-ориентированное программирование на Java

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Форма обучения: очная

Год обучения: 2, семестр: 4

	Вид деятельности	Семестр
		4
1	Лекции, час.	32
2	Практические занятия, час.	
3	Лабораторные занятия, час.	32
4	Занятий в контактной форме без учета промежуточной аттестации, час, из них	66
5	в электронной форме, час.	
6	из них аудиторных занятий, час.	64
7	из них в активной и интерактивной форме, час.	32
8	консультаций, час.	2
9	Самостоятельная работа, час.	76
10	в том числе на выполнение письменных работ, час	
11	Форма аттестации (экзамен, зачет, дифференцированный зачет), час	Э 2
12	Всего зачетных единиц ¹	4

Новосибирск 2019

¹ С учетом выделенных часов на промежуточную аттестацию

Рабочая программа дисциплины составлена на основании федерального государственного образовательного стандарта (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.

Федеральный государственный образовательный стандарт (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА введен в действие приказом Минобрнауки от 19.09.2017 № 929.

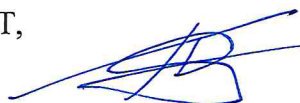
Место дисциплины в структуре учебного плана: Блок 1 Дисциплины (модули), часть формируемая участниками образовательных отношений, обязательная дисциплина.

Рабочая программа дисциплины утверждена решением Ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Программу разработали:

старший преподаватель кафедры общей информатики ФИТ  В.Ю. РЫЛОВ

Заведующий кафедрой общей информатики ФИТ,
доктор физико-математических наук



Д.Е. Пальчунов

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,
кандидат технических наук



А.А. Романенко

Аннотация к рабочей программе дисциплины «Объектно-ориентированное программирование на Java»

Дисциплина «Объектно-ориентированное программирование на Java» реализуется в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, направленность (профиль): ПРОГРАММНАЯ ИНЖЕНЕРИЯ И КОМПЬЮТЕРНЫЕ НАУКИ по очной форме обучения на русском языке.

Цели освоения дисциплины:

Дисциплина «Объектно-ориентированное программирование на Java» имеет своей целью:

- Изучение средств объектно-ориентированного программирования языка Java
- Платформы Java, стандартной библиотеки классов
- Основ многопоточного и распределенного программирования
- Безопасности программных систем использующих технологию Java
- Выработку практических навыков проектирования и реализации объектно-ориентированных программ на языке программирования Java.

Место в образовательной программе:

Дисциплина «Объектно-ориентированное программирование на Java» развивает знания, умения и навыки, сформированные у обучающихся по результатам изучения следующих дисциплин:

- «Информатика»
- «Программирование»
- «Математическая логика и теория алгоритмов»
- «Основы объектно-ориентированного программирования»

Предварительными требованиями к студентам являются:

- Знание одного из классических процедурно-ориентированных языков, предпочтительно языка C
- Знания в области алгоритмической декомпозиции, основных структур данных и технологий работы с ним
- Знание основ теории множеств
- Знание основ объектного подхода

Дисциплина «Объектно-ориентированное программирование на Java» является базовой для освоения следующих дисциплин:

- «Объектно-ориентированный анализ и дизайн»
- «Базы данных»
- «Инженерная и компьютерная графика»
- «Оптимизация производительности Java программ»

Дисциплина «Объектно-ориентированное программирование на Java» реализуется в 4 семестре в рамках базовой части дисциплин (модулей) Блока 1 части формируемой участниками образовательных отношений, и является обязательной дисциплиной.

Содержание дисциплины охватывает круг вопросов, связанных с теорией объектно-ориентированного программирования и особенностями ее поддержки и реализации основных принципов в языке программирования Java и его стандартной библиотеке

По окончании курса студенты получают следующие знания и навыки:

- Знание основ технологии объектно-ориентированной декомпозиции программных систем
- Знание особенностей построения объектно-ориентированных программных систем на Java.
- Знание базовых шаблонов проектирования и построения
- Основные инструментальные средства языка Java и стандартной библиотеки
- Знания и навыки использования системы и библиотеки автоматизированного тестирования JUnit для Java
- Знания и навыки использования системы автоматической сборки Maven
- Навыки использования среды Jet Brains IntelliJ IDEA

Дисциплина «Объектно-ориентированное программирование на Java» направлена на формирование компетенций:

Способен разрабатывать компоненты системных программных продуктов (ПКС-2), в части следующих индикаторов достижения компетенции:

ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня.

ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение.

ПКС-2.9 Знать: шаблоны проектирования и архитектурные шаблоны построения информационных систем.

Перечень основных разделов дисциплины:

1. Виртуальная машина Java
2. Алгоритмические средства языка Java
3. Средства объектного программирования языка Java
4. Средства объектно-ориентированного программирования языка Java
5. Модульность и обобщенное программирование на Java
6. Стандартная библиотека Java
7. Многопоточное программирование на Java, Параллелизм
8. Модель безопасности Java, Программирование распределенных приложений

В процессе изучения дисциплины студенты прослушивают лекции, презентации которых доступны в электронном виде, дополнительно самостоятельно изучают темы с использованием дополнительной литературы и рекомендуемых ресурсов сети Интернет.

В курсе изучается современный стандарт языка программирования Java

Лекционные занятия на курсе проводятся с использованием мультимедийного проектора и в сопровождении с презентациями в формате Power Point.

Дополнительно на лекциях проводятся демонстрации работы основных средств языков/платформ с использованием среды разработки и отладчика.

Лабораторные занятия проходят в терминальных классах, оснащенных персональными компьютерами с установленными средами разработки для Java. Допускается использование студентами собственных персональных компьютеров (ноутбуков).

Во время лабораторных занятий студенты совместно с преподавателем разбирают вопросы по теме курса и занятия, прорабатывают методику решения практических заданий (сформулированных в форме задач на разработку программ на языке Java), решают

лабораторные задания путем разработки программ в процессе самоподготовки. По решению заданий студенты сдают и защищают разработанные программы преподавателю.

Дополнительно преподаватели по желанию могут осуществлять прием и проверку заданий по электронной почте.

В процессе самостоятельной подготовки студенты готовятся к экзамену и имеют возможность задавать вопросы во время предэкзаменационной консультации.

Общий объем дисциплины – 4 зачетных единицы (144 часа).

Правила аттестации по дисциплине

Текущий контроль по дисциплине «Объектно-ориентированное программирование на Java» осуществляется во время проведения лабораторных занятий в следующей форме:

- За решение и сдачу практических задач студенту начисляются баллы, определяющие успеваемость на лабораторных занятиях в течение семестра.
- В течение семестра проводится тестирование в форме коллоквиума (теста с вопросами, подразумевающими открытую форму ответа)

Сдача лабораторной работы (задачи) подразумевает демонстрацию сборки разработанной программы из исходных кодов на языке программирования Java и демонстрации ее работы в соответствии с требованиями лабораторного задания, прохождение автоматических тестов, ответы на вопросы по коду с целью подтверждения авторства.

Промежуточная аттестация по дисциплине «Объектно-ориентированное программирование на Java» проводится по завершению каждого периода ее освоения (семестра) в форме экзамена.

Количество набранных баллов за сдачу лабораторных работ и коллоквиума является важным критерием при выставлении оценки на экзамене и является одним из условий прохождения промежуточной аттестации.

Экзамен проходит в устной форме по вопросам экзаменационного билета. В процессе сдачи экзамена студенту могут задаваться дополнительные задания по теме вопросов билета, в форме написания фрагмента кода демонстрирующего определенный механизм языка программирования или технику объектно-ориентированного программирования на Java.

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Учебно-методическое обеспечение дисциплины

Учебно-методический комплекс по дисциплине «Объектно-ориентированное программирование на Java» размещен на сайте <http://sites.google.com/site/nguoop>

Мухортов В.В., Рылов В.Ю. Объектно-ориентированное программирование, анализ и дизайн. Методическое пособие. ИМ СО РАН, 2002 г. (электронная версия в формате PDF свободно доступна по адресу <https://sites.google.com/site/nguoop/materialy-lekcij---s/OOP%26OOD.PDF?attredirects=0&d=1>)

1. Внешние требования к дисциплине

Таблица 1.1

Компетенция ПКС-2: Способен разрабатывать компоненты системных программных продуктов, в части следующих индикаторов достижения компетенции:
ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня
ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение
ПКС-2.9 Знать: шаблоны проектирования и архитектурные шаблоны построения информационных систем

2. Требования к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий		
	Лекции	Лабораторные	Самостоятельная работа
ПКС-2: Способен разрабатывать компоненты системных программных продуктов			
ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня			
1. Иметь представление об основных современных средах разработки для Java (Oracle NetBeans, Eclipse, JetBrains IntelliJ IDEA) и др. Владеть одной из сред разработки по выбору (Oracle NetBeans, Eclipse, JetBrains IntelliJ IDEA)		+	+
2. Знать основные функции системы автоматической сборки и управления зависимостями Maven		+	+
3. Знать и владеть основными средствами поддержки процедурного стиля программирования (типы данных, переменные, структура программы, функции, структурные типы) языка программирования Java	+	+	+
4. Знать и владеть средствами языка Java поддерживающими основные принципы объектно-ориентированного подхода (поддержка абстракции, инкапсуляции, иерархии, модульности, типизации, параллелизма и сохраняемости в Java)	+	+	+
5. Знать и владеть основными средствами обобщенного программирования на Java (родовые компоненты)	+	+	+
6. Знать основные средства и принципы организации стандартной библиотеки Java	+	+	+
7. Уметь разрабатывать и реализовывать автоматизированные тесты с целью верификации корректности реализованных программ с использованием библиотеки junit		+	+
ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение:			

8. Знать нотацию языка моделирования UML в части диаграмм классов, диаграмм последовательности		+	+
9. Уметь разрабатывать и визуализировать модель классов системы на языке UML		+	+
10. Уметь осуществлять объектно-ориентированную декомпозицию программ на языке Java с разделением на модули (пакеты) с последующей реализацией		+	+
ПКС-2.9 Знать: шаблоны проектирования и архитектурные шаблоны построения информационных систем			
11. Знать основные принципы и шаблоны GRASP (Информационный эксперт, Создатель, Контроллер, Слабое сцепление, Высокая сплоченность), базовые шаблоны GOF (Прототип, фабрика, строитель, одиночка, наблюдатель, приспособленец, стратегия, команда и др.), знать шаблон Модель-Представление-Контроллер (MVC)	+	+	+

3. Содержание и структура учебной дисциплины

Таблица 3.1

№ лекции	Темы лекций	Активные формы, час	Часы	Ссылки на результаты обучения
1	1. Виртуальная машина Java 1.1. История и предпосылки появления Java. 1.2. Понятие виртуальной машины. Среда исполнения и байт-код. Взаимодействие виртуальной машины с операционной системой. 1.3. Пространства классов. Структура приложений на Java. Загрузка классов и инициализация объектов. 1.4. Сферы применения Java в современном информационном мире. 1.5. Версии Java машины и их эволюция. 1.6. Средства ООП, непосредственно поддерживаемые в Java. 1.7. Простейшие приложения на Java.	1	2	3
2	2. Алгоритмические средства языка Java. 2.1. Строгая типизация Java. 2.2. Базовые типы языка. Строки и литералы. 2.3. Преобразования типов в выражениях. 2.4. Особенности инициализации массивов. Операторы управления памятью. 2.5. Краткий обзор операторов. Использование break.	1	2	3
3-5	3. Средства объектного программирования языка Java. 3.1. Представление объектов и классов. 3.1.1. Структура объявления класса.	3	6	4

	<p>3.1.2. Доступ к членам класса.</p> <p>3.1.3. Спецификаторы доступа для обеспечения инкапсуляции.</p> <p>3.1.4. Знакомство с final.</p> <p>3.1.5. Средства управления жизнью объекта. Конструкторы и метод finalize().</p> <p>3.1.6. Принципы работы сборщика мусора.</p> <p>3.1.7. Работа с массивами объектов.</p> <p>3.1.8. Статические поля и методы классов. Классы – утилиты.</p> <p>3.1.9. Блок статической инициализации.</p> <p>3.1.10. Принцип работы ClassLoader.</p> <p>3.2. Реализация отношений между объектами и классами.</p> <p>3.2.1. Ассоциация и агрегация объектов и классов.</p> <p>3.2.2. Использование и зависимость от интерфейсов.</p> <p>3.2.3. Объекты при передаче параметров и возврате из методов.</p> <p>3.2.4. Реализация отношения Клиент – Сервер.</p> <p>3.2.5. Внутренние классы.</p>			
6-7	<p>4. Средства объектно-ориентированного программирования Java.</p> <p>4.1. Наследование в Java.</p> <p>4.1.1. Производные классы.</p> <p>4.1.2. Класс Object. Метод toString().</p> <p>4.1.3. Управление доступом в производных классах.</p> <p>4.1.4. Абстрактные классы и интерфейсы.</p> <p>4.1.5. Реализация интерфейсов как альтернатива множественному наследованию.</p> <p>4.1.6. Информация о типе на этапе исполнения. Оператор instanceof. Приведение типов.</p> <p>4.1.7. Использование класса Class.</p> <p>4.2. Перечисления Java (java.lang.Enum)</p> <p>4.2.1. Особенности классов перечислений</p> <p>4.2.2. Члены перечисления, поведение</p> <p>4.2.3. Использование перечислений</p> <p>4.4. Обработка исключений.</p> <p>4.4.1. Основные принципы и типы исключительных ситуаций.</p> <p>4.4.2. Перехват исключительных ситуаций. Операторы try, throw, throws, catch, finally.</p> <p>4.4.3. Использование нескольких блоков catch и вложенный оператор try.</p> <p>4.4.4. Не перехваченные исключительные ситуации.</p> <p>4.4.5. Встроенные исключительные исключения Java. Классы Throwable и Exception.</p> <p>4.4.6. Принципы создания и использования исключительных ситуаций.</p>	2	4	4

8-10	<p>5. Модульность и обобщенное программирование на Java</p> <p>5.1. Пакеты.</p> <p>5.1.1. Определение пакета.</p> <p>5.1.2. Ограничение доступа.</p> <p>5.1.3. Импортирование пакетов. Разрешение конфликтов имен.</p> <p>5.2. Родовые компоненты и обобщенное программирование</p> <p>5.2.1. Java Generics</p> <p>5.2.2. Отличие от шаблонов C++</p> <p>5.2.3. Ограничения на параметры</p> <p>5.2.4. Совместимость на уровне байт-кода</p>	3	6	4,5
11-13	<p>6.1. Организация пакетов стандартной библиотеки Java. Пакеты java и javax.</p> <p>6.2. Пакет java.lang.</p> <p>6.2.1. Обработка строк.</p> <p>6.2.2. Использование класса System. Управление средой исполнения.</p> <p>6.2.3. Использование классов Number, Double, Integer, Character и др.</p> <p>6.2.4. Класс Math. Класс Compiler и класс ClassLoader.</p> <p>6.3. Пакет java.util</p> <p>6.3.1. Общие принципы организации контейнеров и коллекций в Java.</p> <p>6.3.2. Использование множеств и списков.</p> <p>6.3.3. Использование отображений и ассоциативных контейнеров.</p> <p>6.3.4. Итераторы и исключительные ситуации при работе с классами утилит.</p> <p>6.3.5. Класс Properties.</p> <p>6.3.6. Классы Date и Calendar.</p> <p>6.4. Подсистема ввода вывода java.io</p> <p>6.4.1. Общие концепции организации ввода – вывода в библиотеке Java.</p> <p>6.4.2. Проблема платформенной независимости и локализации.</p> <p>6.4.3. Основные классы потоков ввода-вывода в Java и методы работы с ними.</p> <p>6.4.4. Использование потоков ввода вывода при работе с файлами. Эффективность.</p> <p>6.4.5. Концепция Reader и Writer. Управление локализацией.</p> <p>6.4.6. Использование Tokenizer.</p> <p>6.5. Пакет java.net.</p> <p>6.5.1. Основы работы с сетью в Internet. Адресация.</p> <p>6.5.2. Сокеты. Жизненный цикл сокета.</p> <p>6.5.3. Работа с протоколом HTTP и класс URL.</p> <p>6.6. Графическая подсистема Java. JFC</p> <p>6.6.1. Классы AWT.</p>	2	4	6

	<p>6.6.2. Основы работы с окнами. Component, Container, Panel, Window, Frame, Canvas.</p> <p>6.6.3. Доставка и обработка событий в графической подсистеме. Механизм Listeners.</p> <p>6.6.4. Создание программы с оконным интерфейсом. Рисование графических примитивов.</p> <p>6.6.5. Использование управляющих элементов, диспетчеров компоновки и меню.</p> <p>6.6.6. Связь классов AWT с оконным интерфейсом операционной системы.</p> <p>6.6.7. Классы Swing.</p> <p>6.6.8. Основные принципы графической системы Swing. Платформенная независимость, понятие Look&Feel</p> <p>6.6.9. Написание графического интерфейса с использованием Swing компонентов.</p> <p>6.6.10. Работа с таблицами, текстом, диалогами и HTML.</p>			
14-15	<p>7. Многопоточное программирование на Java. Параллелизм.</p> <p>7.1. Модель потока в Java. Зависимость от реализации потока в операционной системе.</p> <p>7.2. Свойства потока. Синхронизация. Передача сообщений.</p> <p>7.3. Класс Thread и интерфейс Runnable.</p> <p>7.4. Главный поток и способы создания потоков.</p> <p>7.5. Управление потоками и приоритеты потоков. Группы потоков.</p> <p>7.6. Особенности написания многопоточных программ.</p> <p>7.7. Использование синхронизирующих блоков и мониторов объектов. Синхронизированные методы объектов.</p> <p>7.8. Взаимная блокировка.</p> <p>7.9. Использование пула потоков</p>	2	4	4,6
16	<p>8. Модель безопасности Java, Программирование распределенных приложений.</p> <p>8.1. Система безопасности</p> <p>8.1.1. Принципы организации и эволюция модели безопасности в Java.</p> <p>8.1.2. SecurityManager. Инициализация и функции.</p> <p>8.1.3. Права доступа. Управление и проверка прав доступа.</p> <p>8.1.4. Исключительные ситуации.</p> <p>8.1.5. Java Cryptography Extension</p> <p>8.1.6. Алгоритмы шифрования. Ключи и цифровые подписи.</p> <p>8.2. Программирование распределенных</p>	1	2	6

	приложений. 8.2.1. Принципы построения распределенных приложений. 8.2.2. Проблемы передачи объектов и синхронизации в распределенных приложениях. 8.2.3. Реализация сохраняемости. 8.2.4. Three-tier технология. 8.3. Remote Method Invocation 8.3.1. Основные принципы и протокол взаимодействия. 8.3.2. Интерфейс Remote и класс UnicastRemoteObject. 8.3.3. Класс Naming и rmiregistry сервис. 8.3.4. RMI сервер. 8.3.5. RMI клиент. 8.3.6. Модель безопасности, синхронизация и сборка мусора в распределенных RMI приложениях. 8.3.7. Механизм Activation.			
	9. Базовые принципы и шаблоны проектирования 9.1 Принципы GRASP 9.1.1. Информационный эксперт 9.1.2. Создатель 9.1.3. Контроллер 9.1.4. Слабое зацепление 9.1.5. Высокая сплоченность 9.2 Шаблоны проектирования 9.2.1. Прототип 9.2.2. Одиночка 9.2.3. Фабрика 9.2.4. Строитель 9.2.5. Наблюдатель 9.2.6. Приспособленец 9.2.7. Стратегия 9.2.8. Команда	1	2	11
	Итого	16	32	

Таблица 3.2

Темы лабораторных занятий	Активные формы, час	Часы	Ссылки на результаты обучения	Учебная деятельность
Тема 1. Контейнеры, потоки. Object, String. Реализация программы подсчета частоты встречаемости слов в файле (Задание №1)	3	6	1,2,3,4,6	Изучение методических указаний, обсуждение с преподавателем, реализация и сдача программы задания №1

Тема 2. Шаблон проектирования «фабричный метод», журналирование, модульное тестирование. Реализация стекового калькулятора (Задание №2)	4	8	1,2,3,4,7,6	Изучение методических указаний, обсуждение с преподавателем . Реализация и сдача программы задания № 2
Тема 3. Шаблон проектирования MVC. Графический интерфейс пользователя. Реализация игры сапер или тетрис (по выбору, Задание №3)	3	6	1,2,3,4,5,6,7,8,9, 10,11	Изучение методических указаний, обсуждение с преподавателем . Реализация и сдача программы задания № 3
Тема 4. Многопоточность и параллелизм. Реализация фабрики-конвейера по сбору «изделий» из «деталей» (Задание №4)	3	6	1,2,3,4,5,6,7,8,9, 10,11	Изучение методических указаний, обсуждение с преподавателем . Реализация и сдача программы задания № 4
Тема 5. Распределенное программирование. Сетевое взаимодействие. Реализация многопользовательского чата (Задание №5)	3	6	1,2,3,4,5,6,7,8,9, 10,11	Изучение методических указаний, обсуждение с преподавателем . Реализация и сдача программы задания № 5
Итого	16	32		

4. Самостоятельная работа студентов

Таблица 4.1

№	Виды самостоятельно работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
1	Подготовка к практическим занятиям по теме 1, реализация заданий лабораторных работ	1,2,3,4,6	10	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами			

	сети Интернет			
2	Подготовка к практическим занятиям по теме 2, реализация заданий лабораторных работ	1,2,3,4,7,6	10	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
3	Подготовка к практическим занятиям по теме 3, реализация заданий лабораторных работ	1,2,3,4,5,6,7,8,9,10,11	10	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
4	Подготовка к практическим занятиям по теме 4, реализация заданий лабораторных работ	1,2,3,4,5,6,7,8,9,10,11	10	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
5	Подготовка к практическим занятиям по теме 5, реализация заданий лабораторных работ	1,2,3,4,5,6,7,8,9,10,11	12	
	Обучающиеся изучают материалы лекций, опубликованные в электронном виде, читают дополнительную литературу, изучают методические указания к заданию, опубликованные на сайте курса, пользуются справочными ресурсами сети Интернет			
6	Подготовка к экзамену, консультации	3,4,5,6,8,11	24	2
	Итого		76	2

5. Образовательные технологии

Лекционные занятия на курсе проводятся с использованием мультимедийного проектора и в сопровождении с презентациями в формате Power Point, а также с демонстрациями исходного кода и работы программ в среде разработки Jet Brains IntelliJ IDEA, в том числе с использованием отладчика.

В процессе лекции студентам предлагаются вопросы для коллективного обсуждения и анализа, студенты имеют возможность активно задавать вопросы. Продуктивной является форма анализа в посылке «Как бы сделал я если бы разрабатывал язык и компилятор».

Для особо интересных и сложных вопросов используется коллективное голосование в режиме «кто за вариант А», а теперь «кто за вариант Б» с последующим анализом и объяснением «как оно сделано на самом деле и почему».

Данный подход позволяет дать студентам почувствовать саму философию языка Java.

Подобная интерактивная форма концентрирует внимание слушателей и позволяет лектору лучше чувствовать степень понимания материала студентами с возможностью корректировки стиля и глубины изложения.

Лабораторные занятия проходят в терминальных классах, оснащенных персональными компьютерами с установленными средами разработки для Java. Допускается использованием студентами собственных персональных компьютеров (ноутбуков).

Во время лабораторных занятий студенты совместно с преподавателем разбирают вопросы по теме курса и занятия, прорабатывают методику решения практических заданий (сформулированных в форме задач на разработку программ на языке Java), решают лабораторные задания путем разработки программ в процессе самоподготовки. По решению заданий студенты здают и защищают разработанные программы преподавателю.

Дополнительно преподаватели по желанию могут осуществлять прием и проверку заданий по электронной почте.

В процессе самостоятельной подготовки студенты готовятся к экзамену и имеют возможность задавать вопросы во время предэкзаменационной консультации.

Для организации и контроля самостоятельной работы студентов, а также проведения консультаций применяются информационно-коммуникационные технологии.

Таблица 5.1

Информирование	Группы рассылки по электронной почте формируемые семинаристами для каждой группы. Сайт курса по адресу http://sites.google.com/site/nguoor
Консультирование	Электронная почта лектора (v.rylov@g.nsu.ru), электронная почта семинаристов
Контроль	Электронные ведомости учета успеваемости и посещаемости размещаемые на платформе Google docs (http://docs.google.com), репозитории системы контроля версий на платформе bitbucket.org (http://bitbucket.org)
Размещение учебных материалов	Сайт курса по адресу http://sites.google.com/site/nguoor

6. Правила аттестации студентов по учебной дисциплине

По дисциплине «Объектно-ориентированное программирование на Java» проводится текущая и промежуточная аттестация (итоговая по дисциплине)

Текущий контроль по дисциплине «Объектно-ориентированное программирование на Java» осуществляется во время проведения лабораторных занятий в следующей форме:

- За решение и сдачу практических задач студенту начисляются баллы, определяющие успеваемость на лабораторных занятиях в течение семестра.
- В течение семестра проводится тестирование в форме коллоквиума (теста с вопросами, подразумевающими открытую форму ответа)

Сдача лабораторной работы (задачи) подразумевает демонстрацию сборки разработанной программы из исходных кодов на языке программирования Java и демонстрации ее работы в соответствии с требованиями лабораторного задания, прохождение автоматических тестов, ответы на вопросы по коду с целью подтверждения авторства.

Решенные и успешно сданные лабораторные работы совместно с ответами на вопросы коллоквиума формируют портфолио обучающегося.

С целью контроля прогресса решения лабораторных заданий студенты сохраняют исходный код в процессе работы над практическими заданиями в системе контроля версий на основе технологии git или mercurial на общедоступных репозиториях в сети интернет. Выбор репозитория осуществляется на усмотрение студента по согласованию с преподавателем (типичным выбором является общеизвестный репозиторий bitbucket.org со свободным планом обслуживания для маленьких проектов).

Промежуточная аттестация по дисциплине «Объектно-ориентированное программирование на Java» проводится по завершению периода ее освоения (семестра) в форме экзамена.

Количество набранных баллов за сдачу лабораторных работ и коллоквиума является важным критерием при выставлении оценки на экзамене и является одним из условий прохождения промежуточной аттестации.

Экзамен проходит в устной форме по вопросам экзаменационного билета. В процессе сдачи экзамена студенту могут задаваться дополнительные задания по теме вопросов билета, в форме написания фрагмента кода демонстрирующего определенный механизм языка программирования или технику объектно-ориентированного программирования на Java.

Результаты промежуточной аттестации по дисциплине оцениваются по шкале «неудовлетворительно», «удовлетворительно», «хорошо», «отлично». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

В таблице 6.1 представлено соответствие форм аттестации заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.1

Коды компетенций ФГОС	Результаты обучения	Формы аттестации	
		Портфолио	Экзамен
ПКС-2	ПКС-2.1: Владеть: навыками разработки программ на языках высокого уровня	+	+
	ПКС-2.7: Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение	+	+
	ПКС-2.9 Знать: шаблоны проектирования и архитектурные шаблоны построения информационных систем	+	+

Оценочные средства, а также критерии оценки сформированности компетенций и освоения дисциплины в целом, представлены в Фонде оценочных средств, являющемся приложением 1 к настоящей рабочей программе дисциплины.

7. Литература

1. Прохоренок, Н. А. Основы Java: Самоучитель Учебное пособие / Прохоренок Н.А. - СПб:БХВ-Петербург, 2017. - 704 с. ISBN [978-5-9775-3785-8](#). - Текст : электронный. - URL: <https://znanium.com/catalog/product/978545>
2. Хеффельфингер, Д. Разработка приложений Java EE 7 в NetBeans 8 / Дэвид Хеффельфингер ; пер. с англ. А.Н. Киселева. - Москва : ДМК Пресс, 2016. - 348 с. - ISBN [978-5-97060-329-1](#). - Текст : электронный. - URL: <https://znanium.com/catalog/product/1028113>
3. Монахов, В. В. Язык программирования Java и среда NetBeans: Курс лекций / Монахов В.В. - СПб:БХВ-Петербург, 2011. - 703 с. ISBN [978-5-9775-0671-7](#). - Текст : электронный. - URL: <https://znanium.com/catalog/product/355260>
4. Программирование на языке Java : конспект лекций / А. В. Гаврилов, С. В. Клименков, А. Е. Харитонов, Е. А. Цопа. — Санкт-Петербург : Университет ИТМО, 2015. — 123 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/68692.html>
5. Вязовик, Н. А. Программирование на Java / Н. А. Вязовик. — 2-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 603 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/73710.html>
6. Блох, Дж. Java. Эффективное программирование / Дж. Блох ; перевод В. Стрельцов ; под редакцией Р. Усманов. — 2-е изд. — Саратов : Профобразование, 2019. — 310 с. — ISBN [978-5-4488-0127-3](#). — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/89870.html>

Электронные ресурсы сети Интернет

1. Спецификации языка программирования и виртуальной машины Java. (на английском языке), свободный доступ: <https://docs.oracle.com/javase/specs/>
2. Документация Java Standard Edition (на английском языке), свободный доступ: <https://www.oracle.com/technetwork/java/javase/documentation/index.html>
3. Справочный сайт по системе сборки Apache Maven (на английском языке), свободный доступ: <https://maven.apache.org/guides/>
4. Справочные материалы по системе автоматического модульного тестирования JUnit (на английском языке), свободный доступ: <https://junit.org/junit5/docs/current/user-guide/>

8. Учебно-методическое и программное обеспечение дисциплины

Учебно-методическое обеспечение

Основным учебно-образовательным ресурсом курса является WWW сайт <http://sites.google.com/site/nguooop>

Мухортов В.В., Рылов В.Ю. Объектно-ориентированное программирование, анализ и дизайн. Методическое пособие. ИМ СО РАН, 2002 г. (электронная версия в формате PDF свободно доступна по адресу <https://sites.google.com/site/nguooop/materialy-lekcij---s/OOP%26OOD.PDF?attredirects=0&d=1>)

На данном сайте представлены:

- Правила учета успеваемости
- Посещаемость лекций в текущем учебном году
- Демонстрационные презентации лекций курса в формате Microsoft Power Point
- Демонстрационные примеры программ, представленные на лекциях
- Условия практических заданий и курсовых работ для текущего учебного года
- Список основной и дополнительной литературы
- Список вопросов для самоподготовки к экзамену

Программное обеспечение

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows.

Перечень специализированного программного обеспечения для изучения дисциплины представлен в таблице 8.1.

Специализированное программное обеспечение

Таблица 8.1

№	Наименование ПО	Назначение
1	IntelliJ IDEA Community Edition 2019	Среда разработки приложений
2	NetBeans IDE 8.2	Среда разработки приложений
3	Java SE Development Kit 8 (64-bit)	Среда разработки приложений

9. Профессиональные базы данных и информационные справочные системы

При изучении дисциплины не используются.

10. Материально-техническое обеспечение

Таблица 10.1

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Для проведения лекционных занятий
2	Компьютерный класс (с выходом в Internet)	Для проведения лабораторных занятий и организации самостоятельной работы

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Факультет информационных технологий

СОГЛАСОВАНО

Декан ФИТ НГУ



М.М. Лаврентьев

«03» июля 2019 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
по дисциплине **Объектно-ориентированное программирование на Java**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Квалификация: бакалавр

Форма обучения: очная

Год обучения: 2, семестр 4


Форма аттестации	Семестр
Экзамен	4

Новосибирск 2019

Фонд оценочных средств промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Объектно-ориентированное программирование на Java», реализуемой в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль): Программная инженерия и компьютерные науки

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Разработчики:

Старший преподаватель кафедры общей информатики ФИТ  В.Ю. Рылов

Заведующий кафедрой общей информатики ФИТ,
доктор физико-математических наук



Д.Е. Пальчунов

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,
кандидат технических наук



А.А. Романенко

1. Содержание и порядок проведения промежуточной аттестации по дисциплине

1.1. Общая характеристика содержания промежуточной аттестации

Промежуточная аттестация по дисциплине «Объектно-ориентированное программирование на Java» проводится по завершению периода освоения образовательной программы (семестра) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

Код	Компетенции, формируемые в рамках дисциплины	Семестр 4	
		Портфолио	Экзамен
ПКС-2 - Способность разрабатывать компоненты программных продуктов, в части следующих результатов обучения			
ПКС-2.1	Владеть: навыками разработки программ на языках высокого уровня.	+	+
ПКС-2.7	Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение	+	+
ПКС-2.9	Знать: шаблоны проектирования и архитектурные шаблоны построения информационных систем	+	+

Промежуточная аттестация включает 2 этапа. Все компетенций оценивается портфолио, в которое входят работы, выполненные в рамках дисциплины, и экзаменом.

Тематика экзаменационных вопросов и заданий экзамена включает следующие темы (разделы) в двух категориях:

Категория 1 – Основы объектного подхода

- Эволюция стилей и поколения языков программирования
- Основные принципы объектно-ориентированного подхода (абстракция, инкапсуляция, модульность, иерархия, типизация, параллелизм, сохраняемость)
- Объект с точки зрения ООП, взаимоотношения между объектами
- Класс с точки зрения ООП, взаимоотношения между классами
- Многоуровневая метамодель
- Принципы и шаблоны проектирования

Категория 2 – Средства объектно-ориентированного программирования Java

- Виртуальная машина Java
- Средства поддержки процедурного стиля в языке Java
- Средства поддержки объектно-ориентированного подхода в Java
- Средства поддержки обобщенного программирования в Java
- Стандартная библиотека Java

1.2. Порядок проведения промежуточной аттестации по дисциплине

Промежуточная аттестация проводится в форме экзамена и включает 2 этапа: портфолио и экзамен. Для оценивания портфолио студенту необходимо сдать все работы, входящие в структуру портфолио.

Оценка отдельных элементов портфолио осуществляется по балловой системе.

При защите задания обучающийся должен:

- Изложить необходимый для решения теоретический материал
- Указать методику решения задания, предоставить диаграмму классов в нотации UML

- Предоставить исходный код программы или программ реализующий требования задания
- Продемонстрировать корректную работу программы и прохождения автоматических модульных тестов.

За решенные задания начисляются баллы в соответствии с условиями задач. В случае сдачи задания с недочетами или позже установленного срока начисляются баллы меньше балловой стоимости задачи но не менее 50% при условии успешного решения.

Коллоквиум состоит из 18-20 вопросов проводится в виде письменного теста, за каждый правильный ответ на вопрос коллоквиума начисляется от 1 до 2 баллов в зависимости от сложности вопроса.

Для получения оценки «зачтено» необходимо набрать не менее 65 баллов портфолио при типичной стоимости задачи от 10 до 30 баллов. Должно быть сдано не менее 4 заданий.

Необходимым условием для прохождения промежуточной аттестации является оценка «зачтено» по результатам выполненного портфолио.

Экзамен проводится в устной форме. Во время проведения экзамена студенту не разрешается использовать технические средства и литературу. В процессе ответа на вопросы экзаменационного билета студенту могут быть заданы дополнительные вопросы по темам дисциплины

2. Требования к структуре и содержанию фонда оценочных средств промежуточной аттестации по дисциплине

Перечень оценочных средств, применяемых на каждом этапе проведения промежуточной аттестации по дисциплине, представлен в таблице П1.2.

Таблица П1.2

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
Семестр 4			
	Коллоквиум	Средство контроля усвоения учебного материала дисциплины, организованное как письменный тест	Вопросы по темам/разделам дисциплины
	Портфолио	Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в одной или нескольких учебных дисциплинах.	Структура портфолио
	Экзаменационный билет	Комплекс вопросов и/или разноуровневых заданий (списать подходящее из п. про задачи)	Список теоретических вопросов и задач

2.1. Требования к структуре и содержанию оценочных средств аттестации

2.1.1. Список вопросов коллоквиума

- 1) Какие средства Java поддерживают принцип абстракции?

- 2) Какие средства Java обеспечивают поддержку инкапсуляции, расположите их в порядке открытости?
- 3) Определите понятие объекта. Перечислите варианты отношений между объектами, в чем состоит взаимосвязь поведения и состояния объекта?
- 4) Определите понятие класса. Перечислите варианты отношений между классами. Обозначьте взаимосвязь между этими отношениями, если она существует.
- 5) Сформулируйте принцип подстановки Лисковой.
- 6) В чем состоит поддержка принципа типизации в языке Java? Приведите пример кода безопасного приведения типов
- 7) Перечислите примитивные типы языка Java, которые вы знаете. Каков их диапазон допустимых значений?
- 8) Для чего нужен класс `java.lang.Class`. Как узнать класс объекта, если у вас есть ссылка типа `Object o` ?
- 9) Какие строки содержат ошибки и почему?
- `int[] a = new int[4];`
 - `int[] a = new int[];`
 - `int[4] a = new int[4];`
 - `int a[] = new int[4];`
- 10) Какие строки содержат ошибки и почему?
- `final class A { ...`
 - `abstract class A { ...`
 - `protected class A { ...`
 - `abstract int b;`
 - `protected int b(3);`
- 11) Какой блок `catch` будет ловить все?
- `catch(Error e){ }`
 - `catch(AbstractException e){ }`
 - `catch(Throwable e){ }`
 - `catch(Exception e){ }`
- 12) У вас есть следующая иерархия типов:
- ```
class A {
 public void f() {...}
}
class B extends A {
 public void f() {...}
}
class C extends B {
 public void f() {
```
- }
- }
- Вы имеете экземпляр `C c = new C();` как вызвать метод `B.f()` из метода `C.f()`?
  - Как вызвать метод `A.f()` из метода `C.f()`?
- 13) Как правильно создать экземпляр класса `Inner` в методе `main()` класса `Outer`
- ```
public class Outer {
    public class Inner { }
```



```

    public static void main() {
        ???
    }
}

```

14) Что выведет на экран данная программа:

```

public class A{
    public void method() throws Exception {
        System.out.println("Шаг 2");
        throw new Error("Ошибка");
    }
    public static void main(String[] args) {
        A a = new A();
        System.out.println("Шаг 1");
        try {
            a.method();
        } catch (Exception ex) {
            System.out.println("Исключение!");
        }
        System.out.println("Шаг 3");
    }
}

```

15) Что выведет на экран данная программа:

```

public class Test {
    public static void add3(Integer i) {
        int val = i.intValue();
        val += 3;
        i = new Integer(val);
    }
    public static void main(String[] args) {
        Integer i = new Integer (0);
        add3(i);
        System.out.println (i.intValue() );
    }
}

```

16) Какое из следующих определений запрещает наследование класса FooBar:

- static class FooBar { }
- private class FooBar { }
- abstract public class FooBar { }
- final public class FooBar { }
- final abstract class FooBar { }

17) Какое утверждение является верным:

- Анонимный класс может быть объявлен статическим
- Статический внутренний класс не может быть статическим членом внешнего класса
- Статический внутренний класс не требует экземпляра объемлющего класса

- d. На переменные экземпляра внутреннего статического класса можно ссылаться, используя имя этого класса
- e. Для создания экземпляра статического внутреннего класса требуется экземпляр внешнего класса

18) Напишите цикл, который подсчитает среднее значение объектов-значений хранящихся в Map:

```
public class PrintAverage {  
    public static int average(Map<String, Integer> map) {
```

```
    }  
}
```

19) Напишите метод быстрого копирования из InputStream в OutputStream (не забудьте про исключения):

```
public class Copy {  
    public static void copy(InputStream srcStream, OutputStream destStream) {
```

```
    }  
}
```

2.1.2 Требования к портфолио

Портфолио должно содержать результаты участия в коллоквиуме (письменный тест) и 4-5 выполненных заданий по следующим темам:

- Алгоритмические средства Java, классы Object, String, ввод-вывод
- Шаблон проектирования «фабричный метод», журналирование, модульное тестирование
- Шаблон проектирования MVC. Графический интерфейс пользователя
- Многопоточность и параллелизм
- Распределенное программирование и сетевое взаимодействие

2.1.3 Форма и перечень вопросов экзаменационного билета

Форма экзаменационного билета

Таблица П1.3

Новосибирский государственный университет Экзамен	
Объектно-ориентированное программирование на Java наименование дисциплины	
09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА Программная инженерия и компьютерные науки наименование образовательной программы	
ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №	
<ol style="list-style-type: none"> 1. Вопрос из категории 1 2. Вопрос из категории 2 	
Составитель	В.Ю. РЫЛОВ

(подпись)	
Ответственный за образовательную программу	А.А. Романенко

(подпись)	
« ____ » _____ 20 ____ г.	

Перечень вопросов экзамена, структурированный по категориям, представлен в таблице П1.4

Категория, компетенции	Формулировка вопроса
Категория 1 – Основы объектного подхода	
ПКС -2, ПКС-2.1	Эволюция методологий программирования. Парадигмы программирования
ПКС -2, ПКС-2.1	Основные принципы объектного подхода. Абстрагирование
ПКС -2, ПКС-2.1	Основные принципы объектного подхода. Инкапсуляция
ПКС -2, ПКС-2.1	Основные принципы объектного подхода. Модульность
ПКС -2, ПКС-2.1	Основные принципы объектного подхода. Иерархия
ПКС -2, ПКС-2.1	Основные принципы объектного подхода. Типизация
ПКС -2, ПКС-2.1	Объект с точки зрения ООП. Состояние. Поведение
ПКС -2, ПКС-2.1	Объект с точки зрения ООП. Идентичность и жизненный цикл объектов
ПКС -2, ПКС-2.1	Объект с точки зрения ООП. Взаимоотношения между объекта-

	ми.
ПКС -2, ПКС-2.1	Классы. Природа классов. Мета модель. Инстанцирование.
ПКС -2, ПКС-2.1	Классы. Структура класса. Абстрактные классы и интерфейсы
ПКС -2, ПКС-2.9	Классы. Принцип подстановки Лисковской. Принцип разделения интерфейсов
ПКС -2, ПКС-2.7	Классы. Средства UML для построения диаграмм классов
ПКС -2, ПКС-2.1	Классы. Отношения между классами. Ассоциация и агрегация
ПКС -2, ПКС-2.1	Классы. Иерархии классов. Зависимость
ПКС -2, ПКС-2.9	Принципы и шаблоны GRASP
ПКС -2, ПКС-2.9	Шаблоны GOF
Категория 2 – Средства объектно-ориентированного программирования на Java	
ПКС -2, ПКС-2.1	Виртуальная машина. Структура программ. Типы переменных в Java. Принципы работы ClassLoader
ПКС -2, ПКС-2.1	Средства абстракции Java. Структура класса. Статические члены. Перечисления
ПКС -2, ПКС-2.1	Внутренние и вложенные классы. Статический и динамический контекст. Локальные и анонимные классы
ПКС -2, ПКС-2.1	Средства инкапсуляции Java. Поддержка модульности. Пакеты.
ПКС -2, ПКС-2.1	Представление иерархических отношений. Наследование. Интерфейсы и абстрактные классы
ПКС -2, ПКС-2.1	Агрегация и зависимость от времени жизни. Реализация отношений клиент-сервер. Стандартные контейнеры
ПКС -2, ПКС-2.1	Типизация. Правила преобразования типов. instanceof и ClassCastException. Класс Class
ПКС -2, ПКС-2.1	Родовые компоненты Java, отличия от шаблонов C++, ограничения на параметры-типы
ПКС -2, ПКС-2.1	Исключения. Обработка исключительных ситуаций
ПКС -2, ПКС-2.1	Система ввода-вывода, пакет java.io, пакет java.net
ПКС -2, ПКС-2.1	Средства поддержки параллелизма. Активные и пассивные объекты. Класс Object
ПКС -2, ПКС-2.1	Использование Thread и Runnable. Пул потоков, назначение и принципы реализации
ПКС -2, ПКС-2.1	Сохраняемость. Serializable и Externalizable. Программирование распределенных приложений
ПКС -2, ПКС-2.1	Модель безопасности Java. Policy, Permissions, AccessController
ПКС -2, ПКС-2.1	Графическая подсистема. Основы AWT, Applet, Swing components. Событийная модель

Набор экзаменационных билетов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, завершающих освоение дисциплины «Объектно-ориентированное программирование на Java» в текущем учебном году.

3. Критерии оценки сформированности компетенций в рамках промежуточной аттестации по дисциплине

Таблица П1.5

Шифр компетенций	Структурные элементы оценочных средств	Показатель сформированности	Не сформирован («неудовлетворительно»)	Пороговый уровень («удовлетворительно»)	Базовый уровень («хорошо»)	Продвинутый уровень («отлично»)
ПКС-2	Портфолио, экз. вопрос категории 1, экз. вопрос категории 2	ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня	Знает отдельные объектно-ориентированные средства языка Java, не знает отличий виртуального и параметрического полиморфизма, модели наследования и преобразования типов	Знает основные средства ООП языка Java, знает функции управления жизненным циклом объекта, умеет реализовывать основные отношения между классами	Знает все средства ООП языка Java, виртуальный полиморфизм, виртуальное наследование, управление жизненным циклом объектов, умеет реализовывать все типы отношений между классами	Демонстрирует глубокие знания и особенности реализации на уровне языка ассемблера JVM
ПКС-2	Портфолио, Экз. вопрос категории 1	ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение	Не знает нотации, не умеет строить диаграммы UML для визуализации отношений между классами	Может представлять классы и их структуру, может представить диаграмму классов своей программы	Умеет декорировать (именовать, аннотировать, специфицировать мощность) отношения между классами, знает различия в обозначениях зависимости, ассоциации, агрегации и композиции	Уверенно строит диаграммы классов, решения практических задания портфолио снабжены диаграммами, демонстрирующими структуру программы
ПКС-2	Портфолио,	ПКС-2.9 Знать:	Не знает основных	Знает основные прин-	Знает основные	Демонстрирует

	экз. вопрос категории 1	шаблоны проектирования и архитектурные шаблоны построения информационных систем	принципов и шаблонов проектирования	ципы и шаблоны, не демонстрирует глубокого понимания, затрудняется применять их на практике	принципы и шаблоны, может применять их на практике	глубокие познания, знает особенности применения, активно применяет шаблоны и принципы при реализации заданий
--	-------------------------	---	-------------------------------------	---	--	--

4. Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Результаты промежуточной аттестации в 4 семестре определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Оценка «неудовлетворительно» выставляется, если хотя бы одна компетенция не сформирована.

