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]](#footnote-1) | 4 |

Новосибирск 2019

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 доступные в электронном виде, дополнительно самостоятельно изучают темы с использванием 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 использованием студентами собственных персональных компьютеров (ноутбуков).

Во время лабораторных занятий студенты совместно с преподавателем разбирают вопросы по теме курса и зантия, 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 |

1. 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) |  | + | + |
| 1. Знать основные функции системы автоматической сборки и управления зависимостями Maven |  | + | + |
| 1. Знать и владеть основными средства поддержки процедурного стиля программирования (типы данных, переменные, структура программы, функции, структурные типы) языка программирования Java | + | + | + |
| 1. Знать и владеть средствами языка Java поддерживающими основные принципы объектно-ориентированного подхода (поддержка абстракции, инкапсуляции, иерархии, модульности, типизации, параллелизма и сохраняемости в Java) | + | + | + |
| 1. Знать и владеть основными средствами обобщенного программирования на Java (родовые компоненты) | + | + | + |
| 1. Знать основные средства и принципы организации стандартной библиотеки Java | + | + | + |
| 1. 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: | | | |
| 1. Знать нотацию языка моделирования UML в части диаграмм классов, диаграмм последовательности |  | + | + |
| 1. Уметь разрабатывать и визуализировать модель классов системы на языке UML |  | + | + |
| 1. 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 | | | |
| 1. Знать основные принципы и шаблоны GRASP (Информационный эксперт, Создатель, Контроллер, Слабое зацепление, Высокая сплоченность), базовые шаблоны GOF (Прототип, фабрика, строитель, одиночка, наблюдатель, приспособленец, стратегия, команда и др.), знать шаблон Модель-Представление-Контроллер (MVC) | + | + | + |

# 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 приложение 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.  3.1.2. Доступ к членам класса.  3.1.3. Спецификаторы доступа для обеспечения инкапсуляции.  3.1.4. Знакомство с final.  3.1.5. Средства управления жизнью объекта. Конструкторы и метод finalize().  3.1.6. Принципы работы сборщика мусора.  3.1.7. Работа с массивами объектов.  3.1.8. Статические поля и методы классов. Классы – утилиты.  3.1.9. Блок статической инициализации.  3.1.10. Принцип работы ClassLoader.  3.2. Реализация отношений между объектами и классами.  3.2.1. Ассоциация и агрегация объектов и классов.  3.2.2. Использование и зависимость от интерфейсов.  3.2.3. Объекты при передаче параметров и возврате из методов.  3.2.4. Реализация отношения Клиент – Сервер.  3.2.5. Внутренние классы. | 3 | 6 | 4 |
| 6-7 | 4. Средства объектно-ориентированного программирования Java.  4.1. Наследование в Java.  4.1.1. Производные классы.  4.1.2. Класс Object. Метод toString().  4.1.3. Управление доступом в производных классах.  4.1.4. Абстрактные классы и интерфейсы.  4.1.5. Реализация интерфейсов как альтернатива множественному наследованию.  4.1.6. Информация о типе на этапе исполнения. Оператор instanceof. Приведение типов.  4.1.7. Использование класса Class.  4.2. Перечисления Java (java.lang.Enum)  4.2.1. Особенности классов перечислений  4.2.2. Члены перечисления, поведение  4.2.3. Использование перечислений  4.4. Обработка исключений.  4.4.1. Основные принципы и типы исключительных ситуаций.  4.4.2. Перехват исключительных ситуаций. Операторы try, throw, throws, catch, finally.  4.4.3. Использование нескольких блоков catch и вложенный оператор try.  4.4.4. Не перехваченные исключительные ситуации.  4.4.5. Встроенные исключительные исключения Java. Классы Throwable и Exception.  4.4.6. Принципы создания и использования исключительных ситуаций. | 2 | 4 | 4 |
| 8-10 | 5. Модульность и обобщенное программирование на Java  5.1. Пакеты.  5.1.1. Определение пакета.  5.1.2. Ограничение доступа.  5.1.3. Импортирование пакетов. Разрешение конфликтов имен.  5.2. Родовые компоненты и обобщенное программирование  5.2.1. Java Generics  5.2.2. Отличие от шаблонов C++  5.2.3. Ограничения на параметры  5.2.4. Совместимость на уровне байт-кода | 3 | 6 | 4,5 |
| 11- 13 | 6.1. Организация пакетов стандартной библиотеки Java. Пакеты java и javax.  6.2. Пакет java.lang.  6.2.1. Обработка строк.  6.2.2. Использование класса System. Управление средой исполнения.  6.2.3. Использование классов Number, Double, Integer, Character и др.  6.2.4. Класс Math. Класс Compiler и класс ClassLoader.  6.3. Пакет java.util  6.3.1. Общие принципы организации контейнеров и коллекций в Java.  6.3.2. Использование множеств и списков.  6.3.3. Использование отображений и ассоциативных контейнеров.  6.3.4. Итераторы и исключительные ситуации при работе с классами утилит.  6.3.5. Класс Properties.  6.3.6. Классы Date и Calendar.  6.4. Подсистема ввода вывода java.io  6.4.1. Общие концепции организации ввода – вывода в библиотеке Java.  6.4.2. Проблема платформенной независимости и локализации.  6.4.3. Основные классы потоков ввода-вывода в Java и методы работы с ними.  6.4.4. Использование потоков ввода вывода при работе с файлами. Эффективность.  6.4.5. Концепция Reader и Writer. Управление локализацией.  6.4.6. Использование Tokenizer.  6.5. Пакет java.net.  6.5.1. Основы работы с сетью в Internet. Адресация.  6.5.2. Сокеты. Жизненный цикл сокета.  6.5.3. Работа с протоколом HTTP и класс URL.  6.6. Графическая подсистема Java. JFC  6.6.1. Классы AWT.  6.6.2. Основы работы с окнами. Component, Container, Panel, Window, Frame, Canvas.  6.6.3. Доставка и обработка событий в графической подсистеме. Механизм Listeners.  6.6.4. Создание программы с оконным интерфейсом. Рисование графических примитивов.  6.6.5. Использование управляющих элементов, диспетчеров компоновки и меню.  6.6.6. Связь классов AWT с оконным интерфейсом операционной системы.  6.6.7. Классы Swing.  6.6.8. Основные принципы графической системы Swing. Платформенная независимость, понятие Look&Feel  6.6.9. Написание графического интерфейса с использованием Swing компонентов.  6.6.10. Работа с таблицами, текстом, диалогами и HTML. | 2 | 4 | 6 |
| 14-15 | 7. Многопоточное программирование на Java. Параллелизм.  7.1. Модель потока в Java. Зависимость от реализации потока в операционной системе.  7.2. Свойства потока. Синхронизация. Передача сообщений.  7.3. Класс Thread и интерфейс Runnable.  7.4. Главный поток и способы создания потоков.  7.5. Управление потоками и приоритеты потоков. Группы потоков.  7.6. Особенности написания многопоточных программ.  7.7. Использование синхронизирующих блоков и мониторов объектов. Синхронизированные методы объектов.  7.8. Взаимная блокировка.  7.9. Использование пула потоков | 2 | 4 | 4,6 |
| 16 | 8. Модель безопасности Java, Программирование распределенных приложений.  8.1. Система безопасности  8.1.1. Принципы организации и эволюция модели безопасности в Java.  8.1.2. SecurityManager. Инициализация и функции.  8.1.3. Права доступа. Управление и проверка прав доступа.  8.1.4. Исключительные ситуации.  8.1.5. Java Cryptography Extension  8.1.6. Алгоритмы шифрования. Ключи и цифровые подписи.  8.2. Программирование распределенных 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. | 1 | 2 | 6 |
|  | 9. Базовые принципы и шаблоны проектирования  9.1 Принипы 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 |  |  |

# 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 |

# 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 почуствовать 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/nguoop |
| Консультирование | Электронная почта лектора ([v.rylov@g.nsu.ru](mailto: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/nguoop |

# 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 осущестлвяется 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](callto:978-5-9775-3785-8). - Текст : электронный. - URL: <https://znanium.com/catalog/product/978545>

2. Хеффельфингер, Д. Разработка приложений Java ЕЕ 7 в NetBeans 8 / Дэвид Хеффельфингер ; пер. с англ. А.Н. Киселева. - Москва : ДМК Пресс, 2016. - 348 с. - ISBN [978-5-97060-329-1](callto: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](callto: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. Харитонова, Е. А. Цопа. — Санкт-Петербург : Университет ИТМО, 2015. — 123 c. — 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 c. — 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 c. — ISBN [978-5-4488-0127-3](callto: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/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)

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 |

# Профессиональные базы данных и информационные справочные системы

При изучении дисциплины не используются.

# 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 обучающихся |

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».

# Лист актуализации рабочей программы дисциплины «Объектно-ориентированное программирование на Java»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Характеристика внесенных изменений (с указанием пунктов документа) | Дата и №  протокола Ученого совета ФИТ | Подпись  ответственного |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. С учетом выделенных часов на промежуточную аттестацию [↑](#footnote-ref-1)