

**Программа учебной дисциплины «Объектно-ориентированное программирование»**

*Утверждена  
Академическим руководителем*

\_\_\_\_\_ *Н.В. Асеева*

\_\_\_\_\_ 20\_\_\_\_\_

Автор	Демкин В.М.
Число кредитов	4
Контактная работа (час.)	36
Самостоятельная работа (час.)	116
Курс	2
Формат изучения дисциплины	без использования онлайн курса

**I. ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Целями освоения дисциплины «Объектно-ориентированное программирование» является формирование у студентов системного представления об особенностях создания программ с помощью объектно-ориентированного подхода и формирования практических навыков по созданию, отладке и модификации прикладных программ, написанных в объектно-ориентированном стиле.

В результате освоения дисциплины студент должен:

**знать:**

- теорию объектно-ориентированного программирования;

**уметь:**

- применять полученные знания на практике;

**владеть:**

- представлением о проблемах создания и сопровождения современных программных систем, созданных с использованием объектно-ориентированного программирования;  
- навыками создания программ с использованием объектно-ориентированного программирования.

Изучение дисциплины «Объектно-ориентированное программирование» базируется на следующих дисциплинах:

- Программирование;
- Дискретная математика.

При изучении данной дисциплины студенты должны владеть основами программирования и дискретной математики.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении дисциплин, связанных с информационными технологиями, а также при подготовке курсовых и выпускных квалификационных работ.

## **II. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

### **Тема 1. Введение.**

История развития программирования. Структурное программирование. Процедурно-ориентированное программирование и объектно-ориентированное программирование. Объектно-ориентированные языки программирования. Язык программирования и технология Java. IDE для разработки программ на языке Java.

### **Тема 2. Рассмотрение принципов объектно-ориентированного программирования на примере языка Java.**

Объектно-ориентированное проектирование и программирование. Классы и объекты. Абстракция, инкапсуляция, наследование и полиморфизм. Поля, методы и переменные. Преобразование типов. Оператор instanceof. Перегрузка и переопределение методов. Конструкторы. Управление доступом. Использование строк, массивов и коллекций. Обработка исключительных ситуаций.

### **Тема 3. Использование стандартных и внешних библиотек и фреймворков.**

Обзор возможностей стандартных фреймворков и библиотек Java. Обзор внешних библиотек и фреймворков. Стандартный фреймворк коллекций и его применение. Стандартная графическая библиотека Swing и ее применение. Внешний фреймворк JUnit и его применение.

### **Тема 4. Использование шаблонов проектирования.**

Понятие шаблона проектирования. Классификация шаблонов. Примеры использования шаблонов в Java API.

### **Тема 5. Проект по дисциплине.**

Понятие системы контроля версий. Виды систем контроля версий. Основы работы с системой контроля версий: работа с репозиторием (создание, запись изменений, просмотр истории, откат изменений, раск-файлы), использование клиента для работы с версиями проекта (управление, ветвление и слияние веток проекта). Система контроля версий Git. Методологии совместной разработки программного обеспечения. Реализация совместного проекта.

В ходе работы над совместным проектом предполагается организация совместного доступа и редактирование исходного кода программного продукта посредством системы контроля версий (Git). В ходе реализации проекта студенты получают навыки работы с системой контроля версий, управления различными ветками и изменениями конфигурации репозитория проекта, научатся работать с единым исходным кодом в команде. Также предполагается применение в работе нескольких методологий разработки программного обеспечения, что позволит на практике убедиться в эффективности и обоснованности их применения в разработке реальных IT-проектов.

## **III. ОЦЕНИВАНИЕ**

Накопленная оценка за работу студента в течение курса **Онакопленная** складывается из оценок за сданные лабораторные работы и домашнее задание. Для получения накопленной оценки используются следующие весовые множители:

0,7 – для оценки **Олр** за работу студентов во время лабораторных работ,  
0,3 – для оценки **Одз**, за домашнее задание.

**Онакопленная = 0.7\* Олр + 0.3\*Одз**

Для получения результирующей оценки **Орезультат** используются следующие весовые множители:

0,8 – для накопленной оценки

0,2 – для оценки **Оэкзамен**, за экзамен, который является итоговым контролем.

**Орезультат = 0.8\*Онакопленная + 0.2\*Оэкзамен**

В диплом ставится результирующая оценка по учебной дисциплине. Округление оценки до целого значения производится по арифметическим правилам.

#### **IV. ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

##### **Оценочные средства для текущего контроля студента**

*Примерное задание для контрольной работы.*

Проанализируйте предоставленный листинг объектно-ориентированной программы. Выделите основные классы, поля, методы. Опишите программу, выделите ее ограничения, недостатки и достоинства, предложите варианты ее совершенствования.

*Примерное задание для домашней работы.*

Создайте программу на языке Java для объединения нескольких документов в формате pdf в один документ в том же формате. Используйте стороннюю библиотеку для реализации задания. Примените в вашей программе другие возможности найденной библиотеки по работе с документами в формате pdf.

*Основные требования к реферату.*

Реферат должен быть посвящен рассмотрению стандартной или сторонней библиотеки или фреймворка. Тема реферата для каждого студента утверждается преподавателем в индивидуальном порядке. Темы рефератов меняются каждый год.

##### **Оценочные средства для промежуточной аттестации**

*Примерный перечень вопросов к экзамену (итоговый контроль).*

1. Технология Java.
2. IDE для разработки программ на языке Java, Java SDK.
3. Переменные PATH, CLASSPATH, JAVA\_HOME.
4. Основные операторы языка Java.
5. Оператор instanceof и его использование.
6. Соглашения об именах в Java.
7. Элементарные типы данных. Классы-оболочки элементарных типов данных.
8. Преобразование типов данных в выражениях.
9. Ссылочные типы данных. Массивы.

10. Передача параметров по ссылке и по значению в Java.
11. Использование ключевого слова `this` при создании конструкторов.
12. Использование ключевого слова `this` для указания текущего объекта.
13. Конструкторы, перегруженные конструкторы.
14. Использование объектно-ориентированного программирования для создания стека
15. Использование объектно-ориентированного программирования для создания очереди.
16. Статические поля.
17. Статические методы.
18. Пакеты и их использование.
19. Правила именования пакетов.
20. Спецификаторы доступа. Инкапсуляция в Java.
21. Наследование и суперпозиция. Переопределение методов при наследовании.
22. Использование ключевого слова `super` в конструкторах.
23. Использование ключевого слова `super` для указания объекта суперкласса.
24. Преимущества использования общего суперкласса.
25. Использование Java Generics.
26. Полиморфные методы в Java. Вызов полиморфных методов.
27. Основные переопределяемые методы класса `Object`. Метод `equals`.
28. Основные переопределяемые методы класса `Object`. Метод `hashCode`.
29. Основные переопределяемые методы класса `Object`. Метод `toString`.
30. Условия, когда нужно переопределение, соглашения о том, как надо переопределять методы.
31. Коллекции. Интерфейс `Map`.
32. Коллекции. Интерфейс `List`.
33. Коллекции. Интерфейс `Set`.
34. Интерфейсы. Использование интерфейсов `Comparable` и `Comparator`.
35. Неизменяемые (`immutable`) классы, их преимущества и недостатки, правила создания класса, объекты которого будут неизменяемыми.
36. Обработка исключительных ситуаций в Java.
37. Основные методы класса `String`.
38. Сравнение случаев использования `String` и `StringBuilder`.
39. Класс `BigInteger`. Точность вычислений с плавающей точкой.
40. Класс `BigDecimal`. Точность вычислений с плавающей точкой.
41. Принципы построения графического интерфейса пользователя в Java.
42. Обработка событий графического интерфейса пользователя в Java.
43. Что входит в понятие «программная архитектура»?
44. Назовите основные типы программной архитектуры.
45. В чем преимущества деления архитектуры программной системы на горизонтальные слои?
46. Как можно проиллюстрировать высказывание «иерархия уровней воплощает иерархию абстракций»?
47. Что такое точка доступа к сервису?
48. Что реализуется обычно в каркасе?
49. Как в терминах среды разработки Eclipse называется каталог, где хранятся рабочие файлы проектов?
50. Что определяет каждая конкретная перспектива (`Perspective`) среды разработки Eclipse?
51. Какие представления (`Views`) обычно видимы в перспективе `Resources`?
52. Что находится в представлении `Packages`, когда активна перспектива `Java`?
53. Какие возможности по автоматической генерации кода предоставляет `Java`-редактор среды разработки Eclipse?
54. В чем отличия библиотеки `Java Swing` от `Java AWT`?
55. Основное назначение слушателя событий.

56. Основное назначение менеджера компоновки.
57. Основные составляющие элемента JFrame.
58. Каковы функции модели при организации графического интерфейса по принципу «Модель-Вид-Контроллер»?
59. Что определяют шаблоны проектирования?
60. Назначение паттерна «адаптер».
61. Когда целесообразно использовать паттерн «стратегия»?
62. Как связаны между собой паттерны «Фасад» и «Одиночка»?
63. Какой элемент дизайна изменяется при использовании паттерна «Компоновщик»?
64. Что позволяет технология java-апплетов?
65. Какие операции не может обычно выполнять апплет?
66. Преимущества использования технологии java-апплетов.
67. Этапы жизненного цикла апплета.
68. Назначение тега “applet” в языке HTML.
69. Понятие системы контроля версий. Виды систем контроля версий.
70. Работа с репозиторием системы контроля версий: создание, запись изменений, просмотр истории, откат изменений, pack-файлы.
71. Гибкие методологии разработки программного обеспечения. Agile.
72. Гибкие методологии разработки программного обеспечения. Экстремальное программирование.
73. Классические методологии разработки программного обеспечения. Каскадная и итерационная модели.

## V. РЕСУРСЫ

### 5.1 Основная литература

1. Тузовский, А.Ф. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие для прикладного бакалавриата / А.Ф.Тузовский; ЭБС Юрайт. — М.: Юрайт, 2017. — 206 с. — (Университеты России). – Режим доступа: <https://www.biblio-online.ru/viewer/BDEEFB2D-532D-4306-829E-5869F6BDA5F9#page/1>. - Загл. с экрана.

### 5.2 Дополнительная литература

1. Приемы объектно-ориентированного проектирования: паттерны проектирования / Э.Гамма, Р.Хелм, Р.Джонсон, Дж.Влиссидес. - СПб. : Питер, 2004. - 366 с. : ил. - (Библиотека программиста).
2. Дёмкин, В.М. Императивное программирование в примерах на C++ : практикум / В.М.Дёмкин; Гос. ун-т - Высшая школа экономики. - М.: ИД ГУ ВШЭ, 2007
3. Дёмкин, В.М. Основы алгоритмизации и императивного программирования: учебное пособие / В.М.Демкин. - Н.Новгород: НФ ГУ-ВШЭ, 2003

### 5.3 Программное обеспечение

№ п/п	Наименование	Условия доступа
1.	Microsoft Office 2013 Prof +	Государственный контракт

### 5.4 Профессиональные базы данных, информационные справочные системы, интернет-ресурсы (электронные образовательные ресурсы)

№ п/п	Наименование	Условия доступа
	<i>Профессиональные базы данных, информационно-справочные системы</i>	
1.	Сайт документации по языку Java	URL: <a href="https://docs.oracle.com/javase/tutorial">https://docs.oracle.com/javase/tutorial</a>
2.	Сайт компании Oracle	URL: <a href="http://www.oracle.com">www.oracle.com</a>
3.	Сайт проекта Eclipse	URL: <a href="http://www.eclipse.org">www.eclipse.org</a>
	Сайт проекта Git	URL: <a href="http://git-scm.com">http://git-scm.com</a>

### **5.5 Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

- ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением.

Практические занятия проводятся в компьютерных классах с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде НИУ ВШЭ.