

Лабораторная работа №3

Объектно-ориентированное программирование в Java. Системы сборки проекта

Теоретические сведения

С ростом сложности программ, увеличением числа исходных файлов, библиотек и зависимостей в проектах стало очевидно, что ручное управление процессом сборки становится слишком трудоемким и подверженным ошибкам. В начале разработки программ использовались простые команды для компиляции отдельных файлов, но по мере того, как проекты разрастались, управление зависимостями между модулями и библиотеками требовало автоматизации.

Одним из ключевых факторов стало распространение различных языков программирования, которые предлагали свои методы компиляции и линковки. Для обеспечения совместимости и повторяемости процесса сборки на разных платформах потребовалось создавать универсальные решения, которые могли бы работать одинаково эффективно независимо от операционной системы или среды разработки.

Необходимость поддержания стабильности программного кода при внесении изменений также подталкивала разработчиков к использованию систем сборки. Каждый раз, когда происходило обновление или изменение в одном из модулей проекта, требовалось пересобирать не только измененный код, но и весь проект с учетом его зависимостей. Этот процесс становился всё более сложным, что создавало потребность в инструментах, способных автоматически определять, какие части проекта нужно пересобирать.

Система сборки проекта - это программный инструмент, который автоматизирует процесс сборки ПО. Сборка включает в себя компиляцию исходного кода, объединение различных модулей и библиотек, создание исполняемых файлов или архивов, а также выполнение других задач, таких как тестирование и упаковка.

Наиболее популярным представителем подобных систем для сборки проектов на *Java* является [Maven](#).

Ссылки:

- [Подробнее про Maven](#)
- [Использование диаграммы классов UML при проектировании и документировании программного обеспечения.](#)

Задание

1. Разработать UML-диаграмму иерархии классов, согласно варианта (таблица 1).
2. При наименовании компонентов руководствоваться соглашением о наименовании (<https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>).
3. При описании иерархии классов использовать наследование и композицию.
4. На основе UML-диаграммы разработать иерархию классов на языке Java.
5. Весь код должен быть снабжен элементами документирования (JavaDoc).
6. Разработанную иерархию поместить в *.jar* файл для дальнейшего использования в качестве библиотечных классов.
7. Создать консольное приложение для демонстрации работы созданных классов.
8. Составить отчет о проделанной работе.

Требования:

1. Использовать *IntelliJ IDEA* запрещено (рекомендуется использовать *VS Code*).
2. Инициализацию проекта выполнить с помощью *Maven*.
3. Сборку проекта выполнить с помощью *Maven*.

Таблица 1

Вариант	Условие задачи
1	<ol style="list-style-type: none">1. Создать иерархию классов для учёта покупок хот-догов (HunterDog, MasterDog и Berlinka).2. При этом все компоненты могут продаваться отдельно.3. Продать каждый вид хот-дого.4. Подсчитать общую сумму всех заказов.5. Подсчитать количество полных заказов.6. Подсчитать среднюю стоимость заказов.
2	<ol style="list-style-type: none">1. Разработать иерархию классов для хранения информации о деталях: форма, материал, вес, размер.2. Создать не менее 10 объектов, содержащих информацию о различных деталях.3. Подсчитать общий вес деталей, содержащие одинаковую форму.4. Вывести количество деталей.5. Вывести деталь, которая отличается своей формой.
3	<ol style="list-style-type: none">1. Создать класс Sneakers (Кроссовки) с полями name, cost, type, producer. Кроссовки могут быть футбольные, баскетбольные и т.п. (реализовать через наследование).2. Создать приватный внутренний класс Producer с полями name и country.1.3 Создать не менее 10 объектов, содержащих информацию о различных кроссовках.4. Подсчитать количество производителей.5. Подсчитать среднюю стоимость обуви по каждому производителю.6. Подсчитать среднюю стоимость обуви по каждому типу.

Вариант	Условие задачи
	7. Дополнить иерархию классов новым типом кроссовок, при этом нельзя изменять методы, реализующие пункты 4-6
4	<ol style="list-style-type: none"> 1. Создать иерархию классов для учёта выплаты стипендии студентам по итогам экзаменационной сессии. 2. Только студенты, обучающиеся на бюджетной форме и сдавшие сессию в срок, получают стипендию. 3. Студент, имеющий средний бал меньше 5 стипендию не получает, от 6 до 7 получают минимальную стипендию, от 6 до 8 – увеличенную на 25%, от 8 до 10 – на 50%. 4. Ввести данные с информацией о 10 студентах, которые получили стипендию, увеличенную на 25%. 5. Вывести студентов, обучающихся на платной основе. 6. Вывести средний балл по итогам сессии по всем студентам.
5	<ol style="list-style-type: none"> 1. Создать класс Car, Engine и Driver. 2. Класс Driver содержит поля - ФИО, стаж вождения. 3. Класс Engine содержит поля - мощность, производитель. 4. Класс Car содержит поля - марка автомобиля, класс автомобиля, вес, водитель типа Driver, мотор типа Engine. 5. Вывести водителей со стажем более 5 лет. 6. Вывести автомобили советского производства.
6	<ol style="list-style-type: none"> 1. Создать объект класса Компьютер, используя классы Винчестер, Дисковод, ОЗУ. Методы: включить, выключить, проверить на вирусы, вывести на консоль размер винчестера. 2. Создать не менее 10 объектов. 3. Добавить возможность докупать компоненты. 4. Вывести всю информацию о компьютере. 5. Вывести компьютеры, которые собирались вручную. 6. Подсчитать самый выгодный компьютер.
7	<ol style="list-style-type: none"> 1. Создать иерархию классов для учёта самолетов в аэропорту. 2. Создать 3 класса самолетов и минимум 3 экземпляра. 3. Классы должны содержать поля: количество пассажиров, количество топлива, название рейса. 4. Рассчитать расход топлива. 5. Вывести все самолеты, которые содержат количество пассажиров меньше заданного. 6. Вывести все самолеты, название которых начинается с буквы заданную букву.
8	<ol style="list-style-type: none"> 1. Создать иерархию для учета книг в библиотеках. 2. Библиотека содержит в себе название и коллекцию книг, а также методы добавления книги. 3. Класс книги содержит поля: название, автор, год написания. 4. Создать несколько объектов библиотек и добавить книги в каждую. 5. Вывести все книги одного автора из каждой библиотеки. 6. Вывести всю информация по книге(название, автор, год написания).
9	<ol style="list-style-type: none"> 1. Создать иерархию для учета блюд в меню. 2. Меню содержит блюда: напитки и тосты (реализовать через

Вариант	Условие задачи
	<p>наследование). Напитки и тосты имеют название, стоимость, калорийность и метку <code>vegan friendly</code>.</p> <p>3. Создать объект меню и не менее 10 различных блюд.</p> <p>4. Вывести все меню.</p> <p>5. Вывести все <code>vegan friendly</code> блюда.</p> <p>6. Подсчитать среднюю стоимость блюд с калорийностью более 300.</p>
10	<p>1. Создать иерархию комнатных растений с полями название, высота(размер), продолжительность жизни.</p> <p>2. Растения могут быть нескольких видов: Красивоцветущие, Суккуленты, Папоротники. Реализовать через наследование. У красивоцветущих добавить поле "продолжительность цветения".</p> <p>3. Создать не менее 10 объектов различных растений.</p> <p>4. Вывести всю информацию по красивоцветущим.</p> <p>5. Вывести все растения, продолжительность жизни которых менее 5 лет.</p> <p>6. Вывести все папоротники, высота которых больше заданной.</p>
11	<p>1. Создать абстрактный класс <code>пицца</code> и наследовать от него виды <code>Пепперони</code>, <code>Сырная</code>, <code>Мясная</code>.</p> <p>2. Каждая пицца содержит поля <code>цена</code>, <code>вес</code>, <code>диаметр</code> и <code>калорийность</code>.</p> <p>3. Создать не менее 3 экземпляров каждого класса пицц.</p> <p>4. Вывести количество пицц, вес которых превышает заданный.</p> <p>5. Вывести цену каждой пиццы, диаметр которой превышает заданный.</p>
12	<p>1. Создать абстрактный класс <code>Булка</code> и наследовать от него минимум 3 вида булок.</p> <p>2. Абстрактный классный должен содержать поля: <code>цена</code>, <code>ингредиенты</code>, <code>дата момента создания</code>.</p> <p>3. Создать не менее 10 объектов различных булок с разным промежутком выпечки.</p> <p>4. При этом ингредиенты можно добавит отдельно.</p> <p>5. Вывести все булки, которые были созданы ровно 5 мин. назад.</p> <p>6. Вывести булки в которых есть перец.</p>
13	<p>1. Создать иерархию <code>Аниме</code>, три типа: <code>сериал</code> <code>ONA</code>, <code>OVA</code>.</p> <p>2. Класс <code>Аниме</code> должен содержать поля: <code>жанры(коллекция)</code>, <code>рейтинг</code>, <code>количество серий</code>, <code>дата выхода</code>.</p> <p>3. Создать не менее 10 экземпляров.</p> <p>4. Вывести все <code>Аниме</code>, рейтинг которых больше 7.</p> <p>5. Вывести <code>Аниме</code>, которые вышли недавно.</p> <p>6. Перечислить жанры <code>Аниме</code>, которые встречаются чаще всего.</p>
14	<p>1. Создать иерархию классов для учета рыб в Аквариуме.</p> <p>2. В классе <code>рыба</code> реализовать поля: <code>вид</code> (<code>хищная</code>, <code>мирная</code> и т.д.), <code>вес</code>, <code>размер</code> (отдельный класс с полями <code>ширина</code> и <code>длина</code>).</p> <p>3. Создать не менее 5 видов рыб, каждой по 2 экземпляра.</p> <p>4. Вывести информацию о видах рыб, которые являются хищниками и весят больше 100 гр.</p> <p>5. Вычислить самую длинную рыбу.</p>

Вариант	Условие задачи
15	<ol style="list-style-type: none">1. Создать абстрактный класс Food и наследовать от него фрукты, овощи, мясо.2. Создать в классе Food метод расчета энергетической ценности3. В классе Food создать поля: съедобное / несъедобное, коллекция пищевой ценности (белки, жиры, углеводы), название.4. Создать не менее 10 объектов.5. Вывести все съедобные фрукты, пищевая ценность которых меньше заданного числа.6. Вывести еду, название которых заканчивается на введенную букву.