

## Лабораторная работа 1. Развитие представлений о разработке программ. Объектно-ориентированное программирование в *Java*

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 использовать наследование и композицию.
4. На основе *UML*-диаграммы разработать иерархию классов на языке *Java*.
5. Весь код должен быть снабжен элементами документирования (<https://www.jetbrains.com/help/idea/working-with-code-documentation.html>).
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.

Таблица 1

Вариант	Условие задачи
1	1.1 Создать иерархию классов для учёта покупок хот-догов ( <i>HunterDog</i> , <i>MasterDog</i> и <i>Berlinka</i> ); 1.2 При этом все компоненты могут продаваться отдельно. 1.3 Продать каждый вид хот-дога. 1.4 Подсчитать общую сумму всех заказов. 1.5 Подсчитать количество полных заказов. 1.6 Подсчитать среднюю стоимость заказов.
2	1.1 Разработать иерархию классов для хранения информации о деталях: форма, материал, вес, размер. 1.2 Создать не менее 10 объектов, содержащих информацию о различных деталях. 1.3 Подсчитать общий вес деталей, содержащие одинаковую форму. 1.4 Вывести количество деталей. 1.5 Вывести деталь, которая отличается своей формой.
3	1.1 Создать класс <i>Sneakers</i> (Кроссовки) с полями <i>name</i> , <i>cost</i> , <i>type</i> , <i>producer</i> . Кроссовки могут быть футбольные, баскетбольные и т.п. (реализовать через наследование) 1.2 Создать приватный внутренний класс <i>Producer</i> с полями <i>name</i> и <i>country</i> .

	<p>1.3 Создать не менее 10 объектов, содержащих информацию о различных кроссовках</p> <p>1.4 Подсчитать количество производителей.</p> <p>1.5 Подсчитать среднюю стоимость обуви по каждому производителю.</p> <p>1.6 Подсчитать среднюю стоимость обуви по каждому типу.</p> <p>1.7 Дополнить иерархию классов новым типом кроссовок, при этом нельзя изменять методы, реализующие пункты 1.4 – 1.6</p>
4	<p>1.1 Создать иерархию классов для учёта выплаты стипендии студентам по итогам экзаменационной сессии.</p> <p>1.2 Только студенты, обучающиеся на бюджетной форме и сдавшие сессию в срок, получают стипендию.</p> <p>1.3 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%.</p> <p>1.4 Ввести данные с информацией о 10 студентах, которые получили стипендию, увеличенную на 25%.</p> <p>1.5 Вывести студентов, обучающихся на платной основе.</p> <p>1.6 Вывести средний балл по итогам сессии по всем студентам.</p>
5	<p>1.1 Создать класс <i>Car</i>, <i>Engine</i> и <i>Driver</i>.</p> <p>1.2 Класс <i>Driver</i> содержит поля - ФИО, стаж вождения.</p> <p>1.3 Класс <i>Engine</i> содержит поля - мощность, производитель.</p> <p>1.4 Класс <i>Car</i> содержит поля - марка автомобиля, класс автомобиля, вес, водитель типа <i>Driver</i>, мотор типа <i>Engine</i>.</p> <p>1.5 Вывести водителей со стажем более 5 лет.</p> <p>1.6 Вывести автомобили советского производства.</p>
6	<p>1.1 Создать объект класса Компьютер, используя классы Винчестер, Дисковод, ОЗУ.</p> <p>Методы: включить, выключить, проверить на вирусы, вывести на консоль размер винчестера.</p> <p>1.2 Создать не менее 10 объектов.</p> <p>1.3 Добавить возможность докупать компоненты.</p> <p>1.4 Вывести всю информацию о компьютере.</p> <p>1.5 Вывести компьютеры, которые собирались вручную.</p> <p>1.6 Подсчитать самый выгодный компьютер.</p>
7	<p>1.1 Создать иерархию классов для учёта самолетов в аэропорту.</p> <p>1.2 Создать 3 класса самолетов и минимум 3 экземпляра.</p> <p>1.3 Классы должны содержать поля: количество пассажиров, количество топлива, название рейса.</p> <p>1.4 Рассчитать расход топлива.</p>

	<p>1.5 Вывести все самолеты, которые содержать количество пассажиров меньше заданного.</p> <p>1.6 Вывести все самолеты, название которых начинается с букву заданную букву.</p>
8	<p>1.1 Создать иерархию для учета книг в библиотеках.</p> <p>1.2 Библиотека содержит в себе название и коллекцию книг, а также методы добавления книги.</p> <p>1.3 Класс книги содержит поля: название, автор, год написания.</p> <p>1.4 Создать несколько объектов библиотек и добавить книги в каждую.</p> <p>1.5 Вывести все книги одного автора из каждой библиотеки.</p> <p>1.6 Вывести всю информация по книге(название, автор, год написания).</p>
9	<p>1.1 Создать иерархию для учета блюд в меню.</p> <p>1.2 Меню содержит блюда: напитки и тосты (реализовать через наследование). Напитки и тосты имеют название, стоимость, калорийность и метку <i>vegan friendly</i>.</p> <p>1.3 Создать объект меню и не менее 10 различных блюд.</p> <p>1.4 Вывести все меню.</p> <p>1.5 Вывести все <i>vegan friendly</i> блюда.</p> <p>1.6 Подсчитать среднюю стоимость блюд с калорийностью более 300.</p>
10	<p>1.1 Создать иерархию комнатных растений с полями название, высота(размер), продолжительность жизни.</p> <p>1.2 Растения могут быть нескольких видов: Красивоцветущие, Суккуленты, Папоротники. Реализовать через наследование. У красивоцветущих добавить поле продолжительность цветения.</p> <p>1.3 Создать не менее 10 объектов различных растений.</p> <p>1.4 Вывести всю информацию по красивоцветущим.</p> <p>1.5 Вывести все растения, продолжительность жизни которых менее 5 лет.</p> <p>1.6 Вывести все папоротники, высота которых больше заданной.</p>
11	<p>1.1 Создать абстрактный класс пицца и наследовать от него виды Пепперони, Сырная, Мясная.</p> <p>1.2 Каждая пицца содержит поля цена, вес, диаметр и калорийность.</p> <p>1.3 Создать не менее 3 экземпляров каждого класса пицц.</p> <p>1.4 Вывести количество пицц, вес которых превышает заданный.</p> <p>1.5 Вывести цену каждой пиццы, диаметр которой превышает заданный.</p>

12	<p>1.1 Создать абстрактный класс Булка и наследовать от него минимум 3 вида булок.</p> <p>1.2 Абстрактный классный должен содержать поля: цена, ингредиенты, дата момента создания.</p> <p>1.3 Создать не менее 10 объектов различных булок с разным промежутком выпечки.</p> <p>1.4 При этом ингредиенты можно добавить отдельно.</p> <p>1.4 Вывести все булки, которые были созданы 5 мин. назад.</p> <p>1.5 Вывести булки в которых есть перец.</p>
13	<p>1.1 Создать иерархию Аниме, три типа: сериал <i>ONA</i>, <i>OVA</i>.</p> <p>1.2 Класс Аниме должен содержать поля: жанры(коллекция), рейтинг, количество серий, дата выхода.</p> <p>1.3 Создать не менее 10 экземпляров.</p> <p>1.4 Вывести все Аниме, рейтинг которых больше 7.</p> <p>1.5 Вывести Аниме, которые вышли недавно.</p> <p>1.6 Перечислить жанры Аниме, которые встречаются чаще всего.</p>
14	<p>1.1 Создать иерархию классов для учета рыб в Аквариуме.</p> <p>1.2 В классе рыба реализовать поля: вид (хищная, мирная и т.д.), вес, размер (отдельный класс с полями ширина и длина).</p> <p>1.3 Создать не менее 5 видов рыб, каждой по 2 экземпляра.</p> <p>1.4 Вывести информацию о видах рыб, которые являются хищниками и весят больше 100 гр.</p> <p>1.5 Вычислить самую длинную рыбу.</p>
15	<p>1.1 Создать абстрактный класс <i>Food</i> и наследовать от него фрукты, овощи, мясо.</p> <p>1.2 Создать в классе <i>Food</i> метод расчета энергетической ценности</p> <p>1.3 В классе <i>Food</i> создать поля: съедобное / несъедобное, коллекция пищевой ценности (белки, жиры, углеводы), название.</p> <p>1.4 Создать не менее 10 объектов.</p> <p>1.5 Вывести все съедобные фрукты, пищевая ценность которых меньше заданного числа.</p> <p>1.6 Вывести еду, название которых заканчивается на введенную букву.</p>

## Лабораторная работа 2. Технологии проектирования проектных решений: общие принципы, методы, стандарты

1. Написать реферат на заданную тему согласно варианта (таблица 2). При наличии собственных идей, тему можно согласовать с преподавателем.
2. Создать презентацию на основе написанного реферата.
3. Презентация должна содержать не менее 10 слайдов.

Таблица 2

Вариант	Условие задачи
1	<i>Pipe-filter pattern</i>
2	<i>Broker pattern</i>
3	<i>Microservices Architecture</i>
4	<i>Event-bus pattern</i>
5	<i>Blackboard pattern</i>
6	<i>Peer-to-peer pattern</i>
7	<i>Space-Based Architecture</i>
8	<i>Master-slave pattern</i>
9	<i>Event Sourcing</i>
10	<i>Client-server pattern</i>
11	<i>Interpreter pattern</i>
12	<i>Microkernel Architecture</i>
13	<i>Layered pattern</i>
14	<i>Swing Architecture</i>
15	<i>Model-view-controller pattern</i>

### Лабораторная работа 3. Моделирование и алгоритмизация как средства проектирования программного обеспечения

1. Описать блок-схему алгоритма согласно варианта (таблица 3).
2. На основе описанной блок-схемы реализовать алгоритм средствами языка Java.
3. Протестировать разработанный алгоритм при помощи модульных тестов.
4. Создать консольное приложение для демонстрации работы алгоритма.
5. Составить отчет о проделанной работе.

Таблица 3

Вариант	Условие задачи
1	Быстрая сортировка
2	Блочная сортировка
3	Поразрядная сортировка
4	Троичный поиск
5	Двоичный поиск
6	Числа Фибоначчи
7	Алгоритм сортировки пузырьком
8	Сортировка выбором
9	Сортировка слиянием
10	Сортировка подсчетом
11	Экспоненциальный поиск
12	Двоичный поиск
13	Сортировка перемешиванием
14	Пирамидальная сортировка
15	Алгоритм КМП

## Лабораторная работа 4. Инструменты программирования и отладки. Системы контроля версий

1. Создать локальный git репозиторий.
2. Создать три ветки (*main*, *lr1*, *lr3*). Закоммитить лабораторные работы 1 и 3 в ветки *lr1* и *lr3* соответственно.
3. Создать удаленный репозиторий на *GitHub*. Привязать локальный репозиторий к удаленному. Запустить ветки в удаленный репозиторий.
4. Создать и одобрить пул-реквесты на слияние веток *lr1* и *lr3* с веткой *main*.
5. Составить отчет о проделанной работе. Отчет должен содержать описание и скриншоты всех этапов выполнения лабораторной работы.

## **Лабораторная работа 5. Принципы и технологии создания качественного кода в *Java***

1. Создать новую ветку для рефакторинга в ранее созданном *git*-репозитории.
2. Дополнить разработанную в лабораторной работе 1 иерархию классом-репозиторием. Класс должен содержать коллекцию экземпляров иерархии, а также методы добавления, удаления и изменения элементов коллекции.
3. Отрефакторить код лабораторных работ 1 и 3.
4. Участки измененного кода снабдить комментариями.
5. Создать и одобрить пул-реквест на слияние ветки с изменениями с главной веткой.
6. Составить отчет о проделанной работе. Отчет должен содержать описание и обоснование всех изменений в коде.



## **Лабораторная работа 6. Принципы и технологии создания дружественных пользовательских интерфейсов в *Java***

1. Создать новую ветку для графического приложения в ранее созданном *git*-репозитории.
2. Используя библиотеку *Java Swing / JavaFX*, создать графический интерфейс для разработанных ранее лабораторных работ 1 и 3.
3. Элементы интерфейса должны позволять пользователю создавать, удалять и обновлять элементы коллекции, созданного в лабораторной работе 5 класса-репозитория.
4. Создать и одобрить пул-реквест на слияние ветки с интерфейсом с главной веткой.
5. Составить отчет о проделанной работе.