

## Практическая работа №1. Java Core, Наследование.

**Цель работы:** Ознакомиться с механизмом наследования в языке Java. Повторить основные языковые конструкции языка Java.

**Общая постановка задачи:**

Для каждого варианта имеется набор из четырех сущностей. Необходимо выстроить иерархию наследования. В каждом классе (базовом и производных) должно быть минимум одно числовое и одно текстовое поле. При вводе числовых параметров **обязательна** проверка на число и на диапазон (даже если число может быть любое, проверку необходимо реализовать).

Для всех классов должны быть реализованы конструкторы (по умолчанию, с параметрами), методы equals(), hashCode(), toString();

Реализовать консольное Java-приложение, которое имеет простейшее пользовательское меню, состоящее как минимум из следующих пунктов:

- Добавить новый элемент. (Элементы должны добавляться в коллекцию элементов типа базового класса. Необходимо предусмотреть возможность добавления любого объекта производного класса в данную коллекцию.)
- Удалить элемент по индексу.
- Сравнение двух элементов на равенство (по индексам).
- Завершение работы приложения.

**Дополнительные комментарии:** Результат работы должен представлять собой отчет, составленный по СФУ СТО. В работе должен быть указан ход работы, либо исходный код (помимо остальных необходимых пунктов).

Работа должна быть защищена преподавателю. Защита состоит в пояснении хода действия для решения поставленной задачи и ответов на теоретические вопросы (находятся в конце документа) или вопросы, возникающие у преподавателя в ходе проверки работы.

**Варианты классов для построения иерархии:**

- 1) Персона, студент, преподаватель, сотрудник.
- 2) Персона, рабочий, инженер, служащий.
- 3) Организация, завод, страховая компания, фабрика.
- 4) Печатное издание, журнал, книга, учебник.
- 5) Испытание, тест, экзамен, выпускной экзамен.
- 6) Товар, продукт, молочный продукт, игрушка.
- 7) Транспортное средство, автомобиль, поезд, экспресс.
- 8) Двигатель, Двигатель внутреннего сгорания, дизельный двигатель, реактивный двигатель.
- 9) Животное, млекопитающее, парнокопытное, птица.
- 10) Корабль, пароход, парусник, корвет.
- 11) ЭВМ, ПК, ноутбук, планшет.
- 12) Многоугольник, прямоугольник, квадрат, треугольник.
- 13) Помещение, учебная аудитория, кабинет, спортзал.

- 14) Автомобиль, грузовик, автобус, легковой автомобиль.
- 15) Транспортное средство, самолет, автомобиль, корабль.
- 16) Животные, хищники, рыбы, рептилии
- 17) Сооружение, дом, гараж, квартира
- 18) Продукты, фрукты, молочная продукция, мясо
- 19) Мебель, стул, стол, шкаф
- 20) Техника, бытовая техника, садовая техника, автомобильная техника
- 21) Драгоценности, кольца, ожерелья, серьги
- 22) Растения, деревья, кустарники, травы
- 23) Одежда, верхняя одежда, обувь, головные уборы
- 24) Еда, первые блюда, вторые блюда, десерты
- 25) Игрушки, мягкие игрушки, конструкторы, радиоуправляемые

**Контрольные вопросы:**

- 1) Можно ли целочисленный тип данных привести к логическому?
- 2) Для чего служит ключевое слово `final`?
- 3) Что означает принцип постоянства символьных строк?
- 4) Как осуществить проверку символьных строк на равенство?
- 5) Что такое статическое поле/метод?
- 6) Как передаются параметры в методы в языке Java?
- 7) Для чего служит ключевое слово `super`?
- 8) Что такое статическое/позднее связывание?
- 9) Какой класс называется абстрактным? В чем его принципиальное отличие от интерфейса?
- 10) Поясните назначение и реализацию методов `equals` и `hashCode`.
- 11) Для чего необходимы объектные оболочки?
- 12) Какое исключение является непроверяемым? Проверяемым?
- 13) Для чего необходимы методы по умолчанию в интерфейсах?
- 14) Для чего служит блок `finally`?