Код:

//Задача №1

class Worker {

  constructor(name, age, status, salary, post){

    name= prompt("Введите своё имя")

    age= prompt("Введите свой возраст")

    post= prompt("Введите вашу должность")

    salary= prompt("Введите заработную плату")

    status = "Принят"

    alert(`Ваша карточка\nИмя: ${name}\nВозраст: ${age}\nДолжность: ${post}\nЗарплата: ${salary}\nСтатус: ${status}`)

  }

  changeSalary (salary, change){

    change = prompt("Хотите изменить заработную плату?")

    if(change == "да"||"ДА"||"Да"){

      salary = prompt("Введите новую заработную плату")

      return salary

    }else{

      return salary

    }

  }

  changePost (post, change){

    change = prompt("Хотите изменить вашу должность?")

    if(change == "да"||"ДА"||"Да"){

      post= prompt("Введите новую должность")

      return post

    }else{

      return post

    }

  }

}

let worker = new Worker

class Company extends Worker {

  constructor (nameCom) {

    nameCom = prompt("Введите наименование предприятия")

    alert("Наименование предприятия: " + (nameCom))

  }

  transferCompany (nameCompany, transfer) {

    transfer = prompt("Хотите изменить вашу должность?")

    if(transfer == "да"||"ДА"||"Да"){

      nameCompany = prompt("Введите новую должность")

      return nameCompany

    }else{

      return nameCompany

    }

  }

  disbandStatus (status, disband) {

    disband = prompt("Хоттите ли вы уволить сотрудника")

    if(disband == "да"||"ДА"||"Да"){

      status = "Уволен"

      return status

    }else{

      return status

    }

  }

}

let company = new Company

// Задача №2

class Car {

  constructor (marka, model, number, color){

    marka = prompt('Введите марку автомобиля')

    model = prompt('Введите модель автомобиля')

    number = prompt('Введите номер автомобиля')

    color = prompt('Введите цвет автомобиля')

    alert(`Регистрация:\nМарка: ${marka}\nМодель: ${model}\nНомер: ${number}\nЦвет: ${color}`)

  }

}

class Parking extends Car{

  inforCar (info){

    info = super.constructor

    alert(info)

  }

  appendCar (append, car){

    append = []

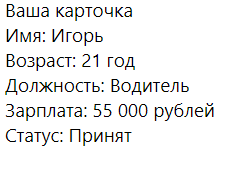
    append.push(car)

  }

}

let parking = new Parking

Результат:

1) 

2) 