

Факультет компьютерных технологий и прикладной математики
Кафедра вычислительных технологий
02.03.02

Паттерны программирования
Лабораторная работа № 3. разработка простейшего ООП приложения

Каждое задание должно быть загружено на личный git-репозиторий отдельным коммитом. Лабораторная работа выполняется в одной папке. Защита работы возможна на любой лабораторной работе от 1 до 16. Последний коммит для данной работы должен быть сделан не позднее лабораторной работы № 5. За защиту выставляется оценка. В случае, если последний коммит сделан позже срока, но до 10 ЛР, за работу выставляется оценка минус ОДИН балл. В случае, если последний коммит сделан до 16 ЛР, за работу выставляется оценка минус 2 балла. Если последний коммит сделан позже, работа не проверяется. Наличие выполненных работ учитывается на зачете.

Если часть задач выполнена в один коммит, работа не проверяется. Если все коммиты сделаны в один час, работа не проверяется.

Общее задание состоит из двух частей, первая часть, реализация методов для классов указанной предметной области. В качестве задачи выбрана следующая: реализация учета кадровой информации заданной фирмы.

Задание 1. Информация о возможном сотруднике фирмы будет следующей: Фамилия Имя Отчество, год рождения, телефон, адрес, e-mail, серия номер паспорта, специальность, стаж работы по специальности, название предыдущего места работы, наименование должности, заработная плата на предыдущем месте работы. Реализовать класс Employee, в котором будут указанные поля, реализовать геттеры и сеттеры для этих полей, реализовать конструктор, с учетом того, что поля «название предыдущего места работы, наименование должности, заработная плата на предыдущем месте работы» могут либо присутствовать одновременно, либо отсутствовать, если стаж = 0.

Задание 2. Из отдельного тестового класса TestEmployee создать несколько сотрудников, отобразить информацию о них на экран. Для отображения переопределить соответствующий метод суперкласса Вашего класса о возможном сотруднике.

Задание 3. Реализовать метод класса Employee, который будет проверять, является ли введенная строка Российским номером мобильного телефона, возвращаемое значение true или false. В номере могут быть любые дополнительные символы. Дополнительно реализовать

метод класса, который выбрасывает исключение о некорректном аргументе, если номер некорректный, иначе возвращает номер телефона в формате 7-918-1234567. С использованием построенного метода определить setter.

Задание 4. Реализовать метод класса Employee, который проверяет, является ли введенная строка адресом email, возвращаемое значение true или false. Дополнительно реализовать метод класса, который выбрасывает исключение о некорректном аргументе, иначе возвращает e-mail полностью в нижнем регистре. С использованием построенного метода определить setter.

Задание 5. Реализовать в проектируемом классе Employee проверку, является ли поданная строка Фамилией Именем Отчеством, причем количество слов 3-6 (учитываем двойные фамилии через дефис, двойные имена через Дефис, двойные отчества «Ахмед оглы»), нет знаков препинания и прочих символов, за исключением тире или дефиса (двойные имена и фамилии могут содержать). Пример ФИО, которые должны быть распознаны Салтыков - щЕдрин Иван-Руслан Ахмед заде, Иванов Иван Иванович. Пробелы вначале, в конце, между словами и дефисами могут быть в любом количестве. Проверку выполнить с помощью регулярных выражений, возвращаемое значение true или false. . Дополнительно реализовать метод класса, который выбрасывает исключение о некорректном аргументе, если ФИО некорректный, иначе возвращает без пробелов вначале, конце и вокруг тире/дефиса, без тире, только дефис, с одним пробелом между словами, все слова, кроме второй части отчества с большой буквы, все остальные буквы – маленькие. Пример возвращаемого значения Салтыков-Щедрин Иван-Руслан Ахмед заде. С использованием построенного метода определить setter.

Задание 6. Реализовать метод, который проверяет, является ли введенная строка датой рождения в любом из форматов 10.02.2021, 01.02.2021, 1.02.2021, 10.02.21, 01.02.21, 1.02.21, возвращаемое значение true или false. Дополнительно реализовать метод класса, который выбрасывает исключение о некорректном аргументе, если дата некорректна, иначе вернуть дату рождения в формате 10.02.2021. С использованием построенного метода определить setter.

Задание 7. Реализовать подобные методы для проверки корректности паспортных данных.

Задание 8. Из созданного в задании 2 тестового класса TestEmployee протестировать работу составленных методов. Пользователь указывает, что он будет вводить, ФИО, телефон, дату или e-mail, вводит, программа

выводит информацию в нужном формате или обрабатывает исключение и говорит, что ввод работает неверно.

Задание 9. Внести изменения в конструктор с учетом изменившихся сеттеров.

Задание 10. Реализовать отдельный класс `TerminalViewListEmployee` для отображения списка возможных сотрудников в терминале. Необходимо попросить пользователя возможного сотрудника, проверить корректность введенных данных, в случае корректности сохранить в коллекцию, иначе попросить пользователя внести данные заново.

Задание 11. После завершения ввода всех возможных сотрудников отобразить их список на экране.

Задание 12. Реализовать метод записи всех возможных сотрудников коллекции в файл. (*) Учесть следующее, хранить паспортные данные в незащищенном виде не рекомендуется по закону о персональных данных, в связи с чем хранить паспортные данные в зашифрованном виде. Продумать организацию ключа.

Задание 13. Реализовать метод чтения списка возможных сотрудников из файла. Будем предполагать, что данные пользователей в файле записаны корректно, на основании заданий 10 – 12. (*) учесть, что паспортные данные нужно расшифровать.

Задание 14. Вывести коллекцию возможных сотрудников в отдельный класс `ListEmployee`. Методы чтение всех возможных сотрудников, добавление одного сотрудника, отображение списка сотрудников внести в методы класса.

Задание 15. Реализовать метод поиска сотрудника по ФИО, ФИО должно быть корректным, внести изменения в класс сотрудник.

Задание 16. Реализовать метод поиска сотрудника по e-mail, телефону, паспортным данным, внести соответствующие изменения в класс сотрудник.

Задание 17. В классе `TerminalViewListEmployee` из задания 10, отвечающем за работу с пользователем реализовать программный продукт, который выполняет одно из действий:

1. Добавляет нового пользователя
2. Отображает список пользователей
3. Находит пользователя по введенной строке(задание 15,16)

4. Изменяет конкретного пользователя
5. Удаляет конкретного пользователя
6. Сохраняет изменения в файл.
7. Закрывает программу

Задание 18. Реализовать в классе коллекции сотрудников ListEmployee методы сортировки по каждому из полей.

Задание 19. В классе TerminalViewListEmployee отвечающем за работу с пользователем реализовать программный продукт, который выполняет одно из действий:

1. Добавляет нового пользователя
2. Отображает список пользователей
3. Находит пользователя по введенной строке (задание 15,16)
4. Изменяет конкретного пользователя
5. Удаляет конкретного пользователя
6. Сохраняет изменения в файл.
7. Сортирует по введенному полю и выводит первые n записей на экран.
8. Закрывает программу

Задание 20. Индивидуальное задание. Выбрать тему.