МИНИСТЕРСТВО ОБЩЕГО И ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

РОСТОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ

«РОСТОВСКИЙ-НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»

Вариант № 7

Дисциплина: Разработка программных модулей

**ОТЧЕТ**

по контрольной работе №2.

Группа: ИС-33

Студент: Клименко О.Ю.

Преподаватель: Кошкина А.А.

Ростов-на-Дону

2023

**Описание.**

**Класс Main:**  
 Основной класс программы, содержащий метод main() для запуска программы.   
В классе Main используются все методы из других классов, показывающие определенную информацию

**Класс Stable:**  
 Предназначен для предоставлении информации о конюшне. Является **абстрактным** классом, хранящий в себе единственное во всей программе свойство, имеющее технологию **static**.  
Имеет атрибуты title - название, address - адрес, quantity - количество лошадей, cost - цена посещения конюшни(всегда равна 1499.99 из-за технологии static).

**Класс Stall:**  
 Показывает информацию о стойлах внутри конюшни. Является единственным классом **final**.  
Имеет следующие свойства: stalls - количество стойл для лошадей, watering\_hol - количество водопоев для лошадей, heating - есть ли отопление в стойлах.

**Класс Horses:**  
 Класс программы, показывающий какая информация о лошадях будет храниться, а именно: id - ID лошади, name - имя лошади, gender - пол лошади, age - возраст лошади.  
В данном классе также представлен метод **findAverageAge()**, который находит средний возраст всех лошадей, находящихся в конюшне.

**Класс HorseList:**  
 Класс, благодаря которому мы можем записывать данные о лошадях в лист.  
В этом классе есть метод **getHorsesID()**, с помощью которого мы можем находить определенную лошадь в списке при указанном пользователе ID лошади и возвращать null, если лошадь с данным нам ID не найдена.

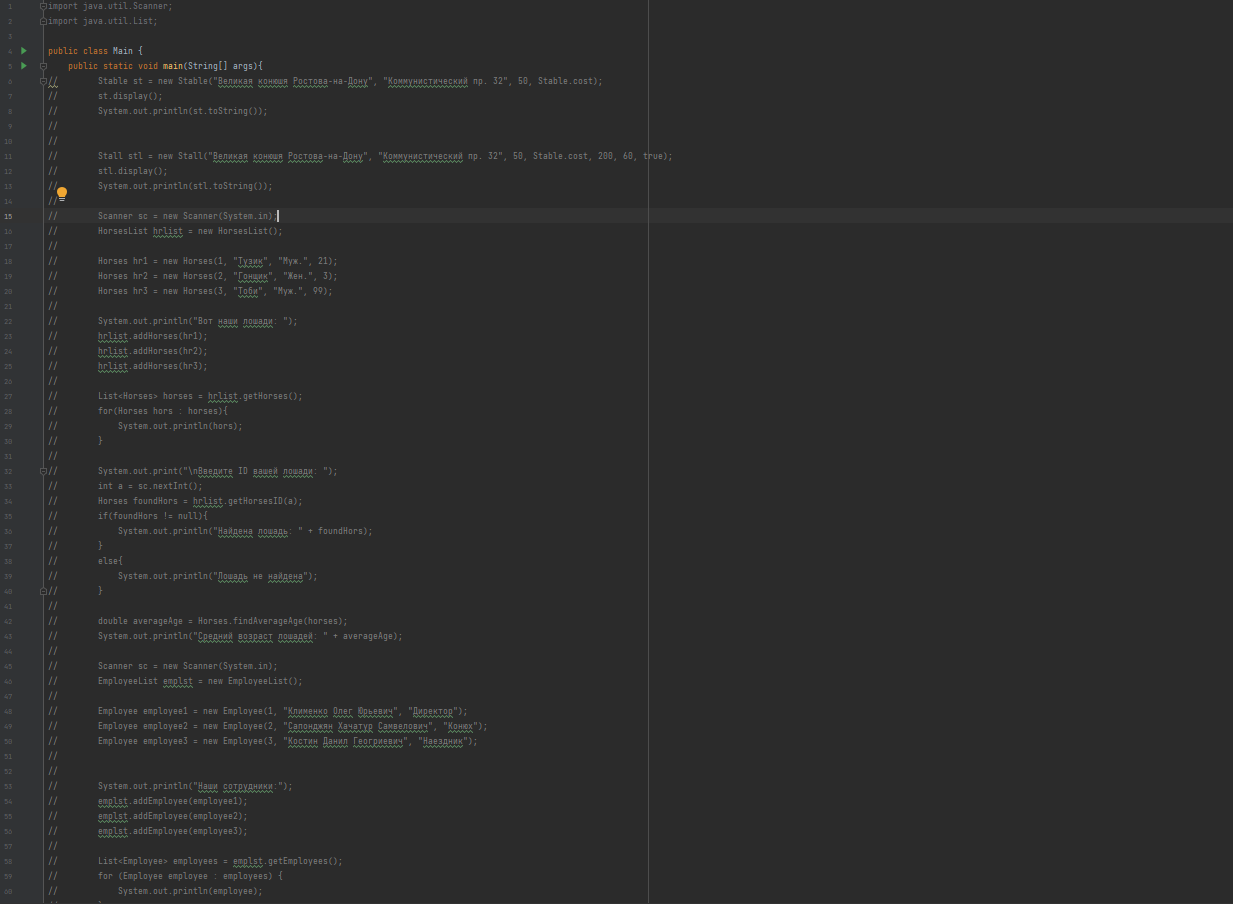
**Класс Employees:**  
 Класс программы, показывающий какая информация о сотрудниках будет храниться, а именно: id - ID сотрудника, FIO - фамилия, имя и отчестиво сотрудника,, should - должность сотрудника.

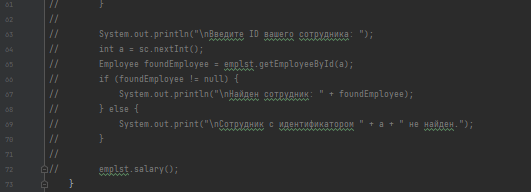
**Класс EmployeesList:**  
 Класс, благодаря которому мы можем записывать данные о сотрудниках в лист.  
В этом классе есть метод **getEmplyeeById()**, с помощью которого мы можем находить определенного сотрудника в списке при указанном пользователе ID сотрудника и возвращать null, если сотрудник с данным нам ID не найден.  
Данный класс использует интерфейс **Salary**, в котором указывается зарплата каждой профессии в конюшне.

**P.S.**Данная программа реализует все ранее изученные мной технологии: инкапсуляцию, наследование, абстрактные классы, интерфейсы, final, static, полиморфизм, конструкторы и инициализаторы.  
Классы **Horses** и **Employees** не наследуются из класса **Stable**, а имеют собственные конструкторы, которые выводятся в **Main**.

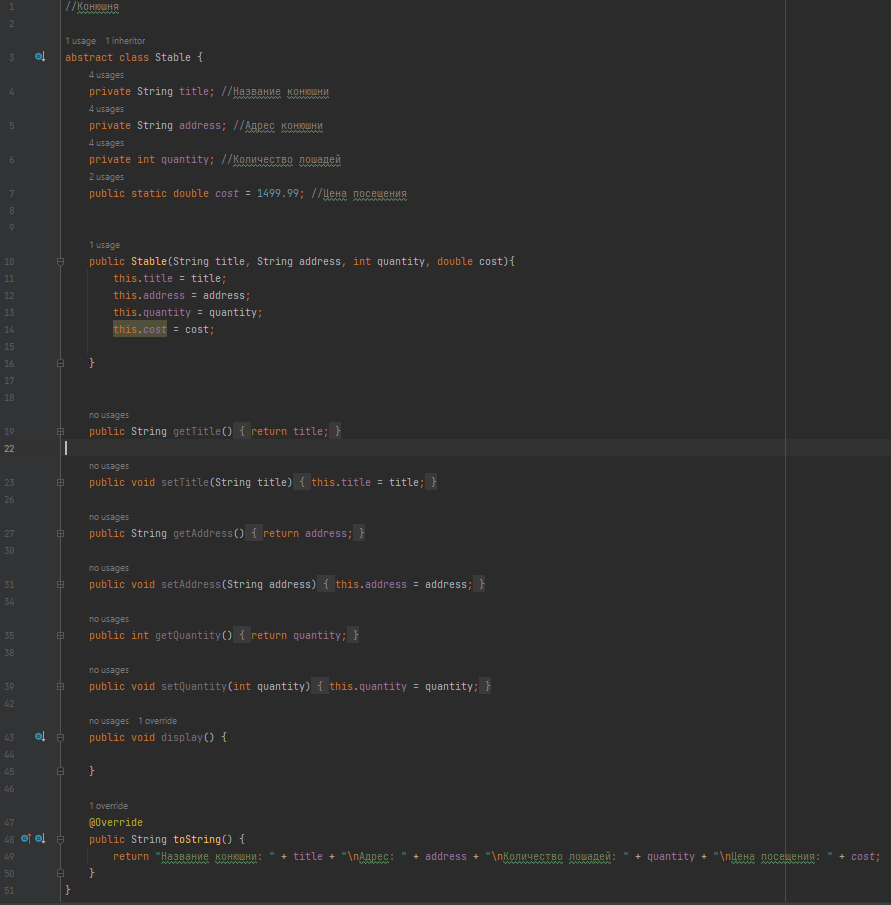
**Скриншоты кода из каждого класса.**

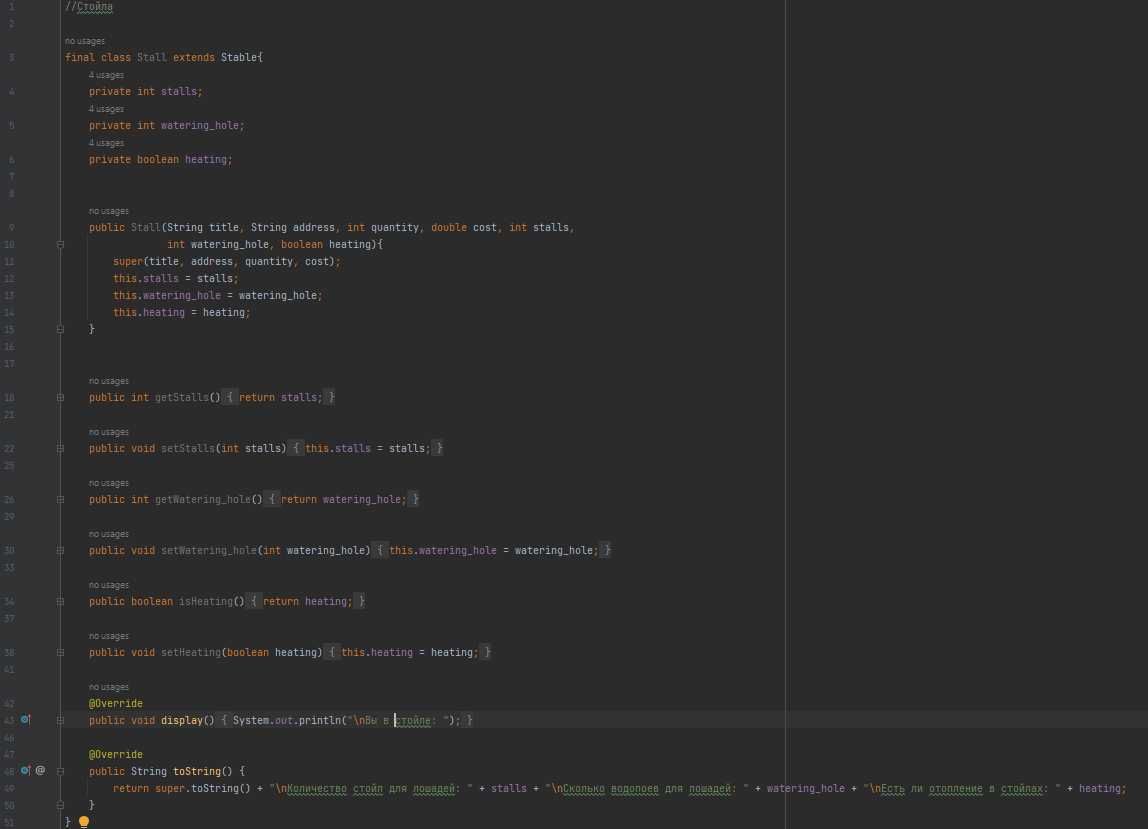
Класс Main:

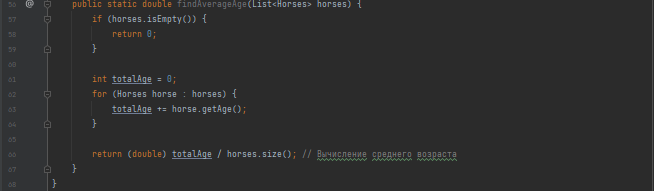
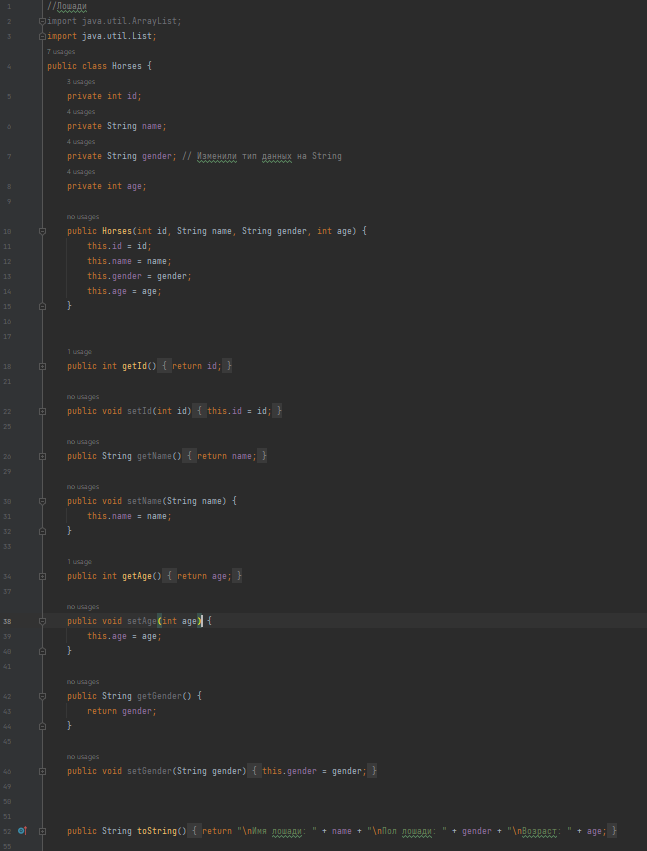
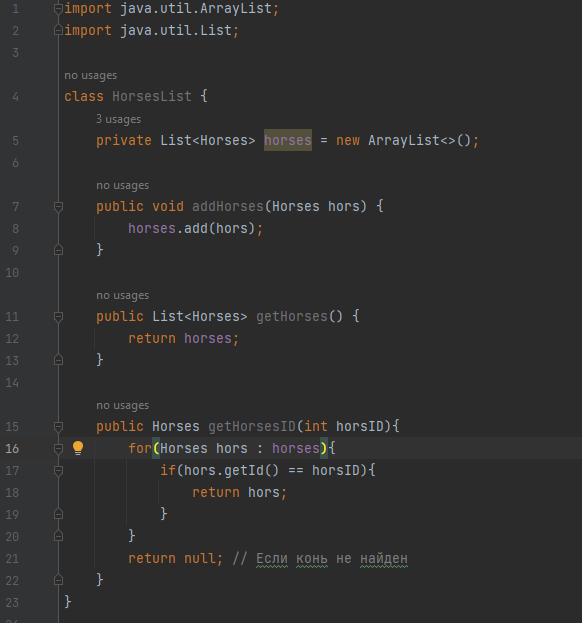


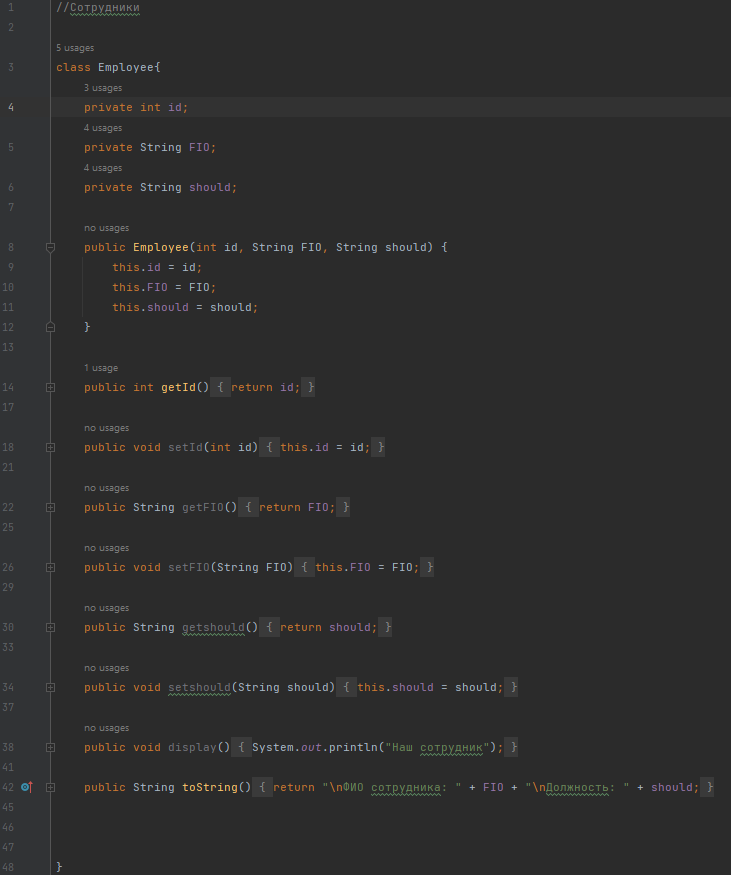


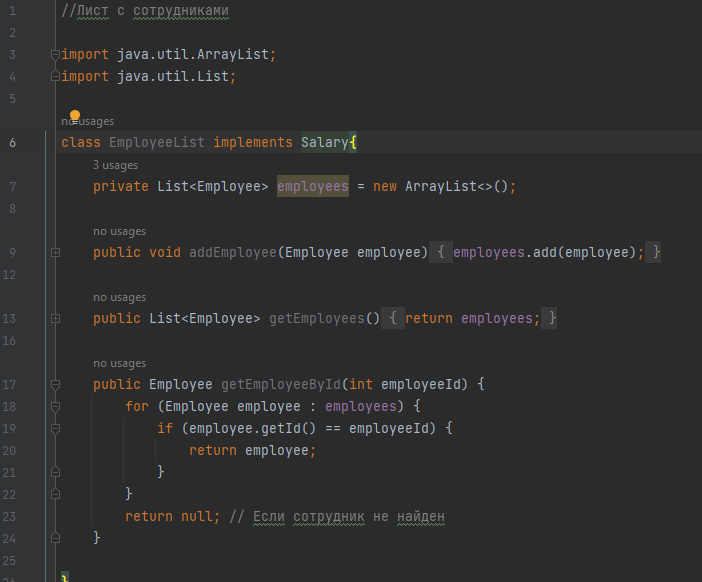
Класс Stable:

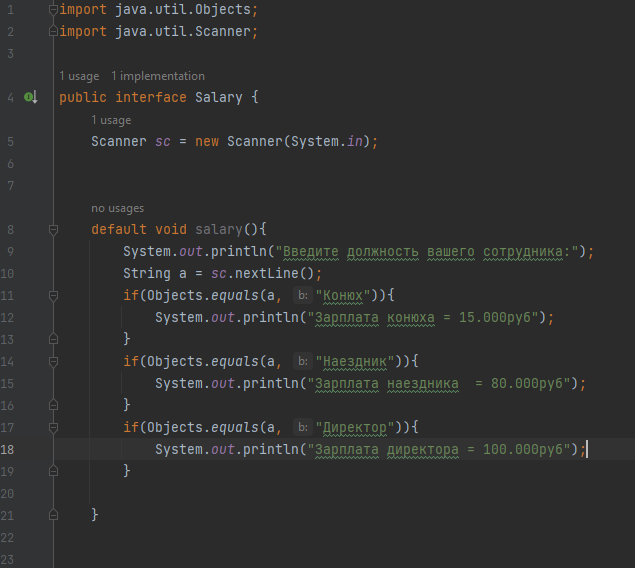


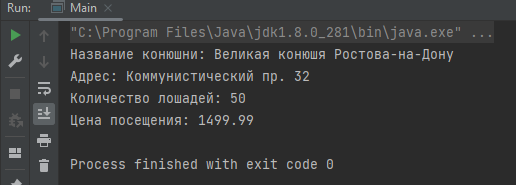
Класс Stall:  


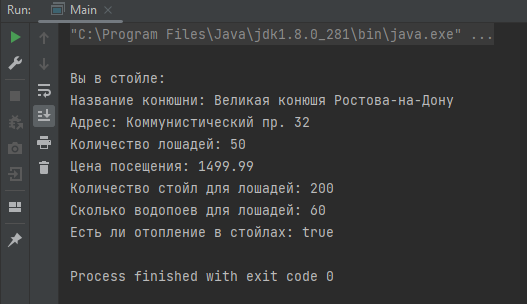
Класс Horses:  
  
Класс HorsesList:  


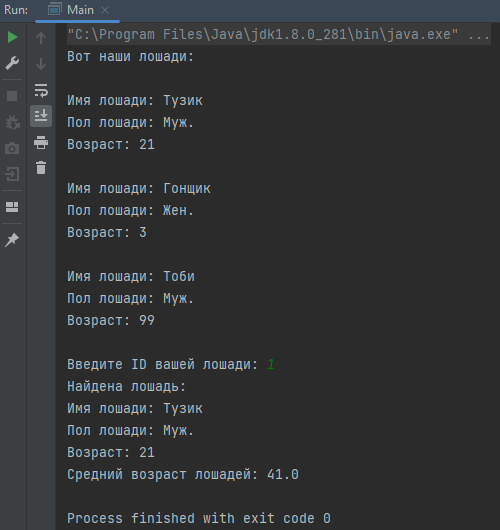
Класс Employees:  


Класс EmployeesList:  


Интерфейс Salary:  


Вывод с класса Stable:  


Вывод с класса Stall:  


Вывод с классов Horses и HorsesList:  


Вывод с классов Employees и EmployeesList:  
