Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации»

Нижегородский институт управления

Кафедра Информатики и информационных технологий

ОТЧЕТ

Задания лекции №9

Выполнил: студент группы:

ИБ-321

Савицкий Руслан Станиславович

Преподаватель:

Окулич Евгения Викторовна

Нижний Новгород

2022 г.

## **Задание 9.1**

* 1. Реализовать проекты с программами рисунков 9.1 и 9.2.
  2. Определить (идентифицировать) причины ошибок проекта рисунка 9.1 и исправить их.
  3. В проекте рисунка 9.2 объяснить работу кодов строчек 6-10 и добиться вывода другого значения возраста по методу **.setAge**.
  4. Разобраться в коде, приведённом в <https://3ty.ru/languages/java/gettery-i-settery-v-java> и реализовать его в соответствующем проекте
  5. Сформулировать преимущества и недостатки инкапсуляции
  6. Определить (идентифицировать) причины ошибок проекта рисунка 9.1 и исправить их.

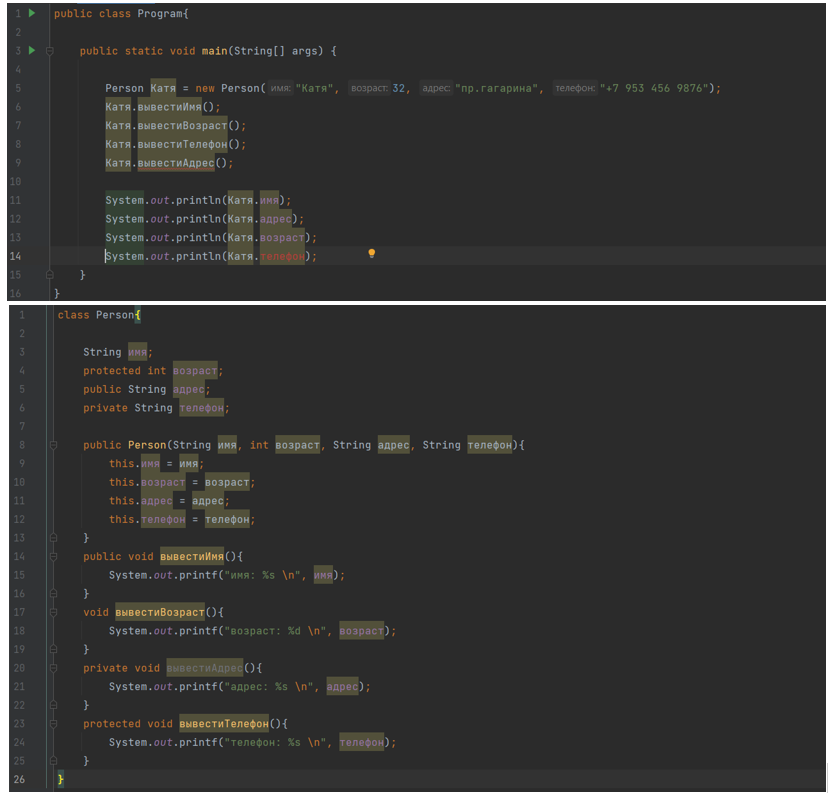
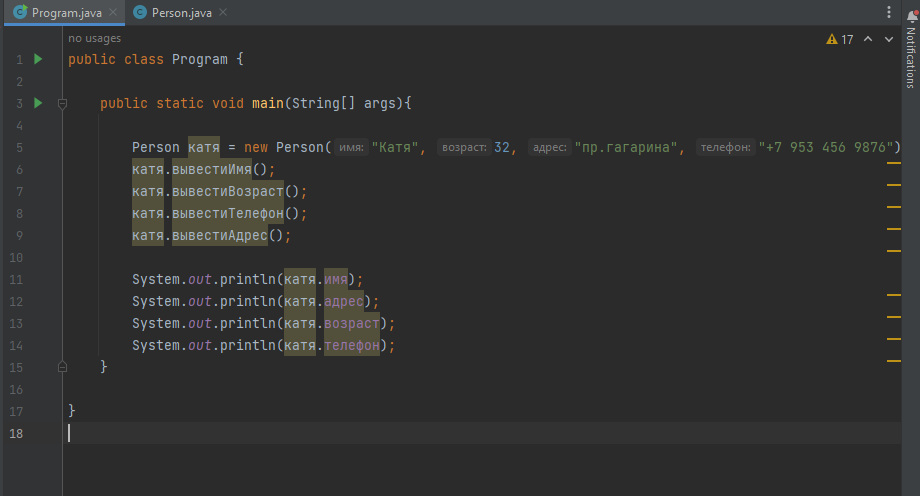
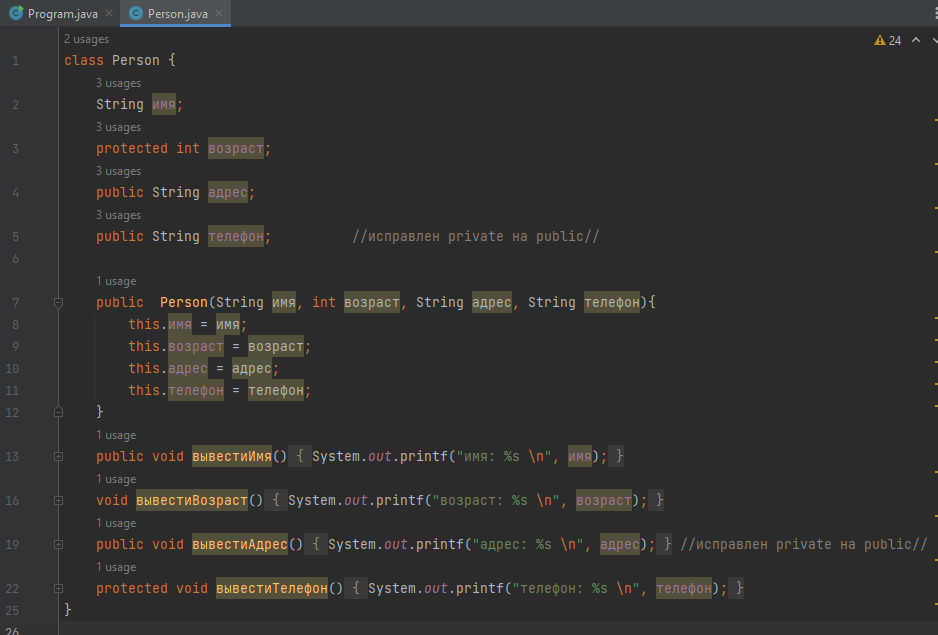


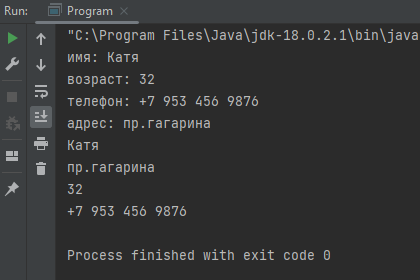
Рисунок 9.1. Иллюстрация применения модификаторов доступа

Исправлено



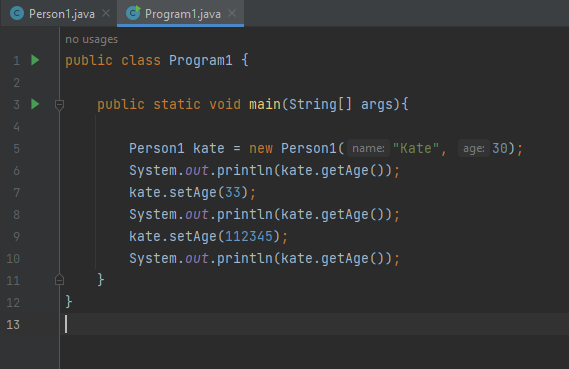


Результат



1.3 В проекте рисунка 9.2 объяснить работу кодов строчек 6-10 и добиться вывода другого значения возраста по методу **.setAge**.





Программа состоит из двух классов: `**Program1** и **Person1**. Класс **Person1** определяет объект типа **Person1**, имеющий два поля: **name** и `age**age**, которые задаются в **Person1**.

Класс **Person1** имеет также четыре метода: **getName()**, **setName()**, **getAge()** и **setAge()**. Методы **getName()** и **getAge()** используются для получения значения соответствующих полей, а методы **setName()** и **setAge()** используются для установки новых значений для этих полей.

Класс **Program1** содержит метод **main()**, который создает новый объект **Person1** с именем "Kate" и возрастом 30. Затем он вызывает метод **getAge()**, чтобы распечатать текущий возраст "Kate" (30). Затем он вызывает метод **setAge()** и передает ему значение 33, чтобы изменить возраст "Kate" на 33. Затем он снова вызывает **getAge()**, чтобы распечатать обновленный возраст (33). В заключение, он вызывает **setAge()** и передает ему значение 112345, которое находится за пределами разумных значений для возраста, поэтому **setAge()** не изменит значение поля **age**. После этого программа распечатает последнее значение возраста "Kate" (33).

1.5 Сформулировать преимущества и недостатки инкапсуляции

Преимущества инкапсуляции:

1. Безопасность: инкапсуляция обеспечивает безопасность объектов, скрывая их внутреннее состояние от несанкционированного доступа извне. Это предотвращает случайное изменение данных, а также обеспечивает контролируемый доступ к ним.
2. Модульность: использование инкапсуляции позволяет создавать отдельные модули, каждый из которых может быть разработан, тестирован и отлажен независимо от других модулей. Это упрощает поддержку и улучшение кода.
3. Гибкость: благодаря инкапсуляции объекты могут быть изменены без изменения интерфейса, что позволяет избежать ошибок в других частях программы, использующих этот объект.
4. Сокрытие информации: скрытие деталей реализации объектов облегчает их использование

Недостатки инкапсуляции:

1. Дополнительный код: использование инкапсуляции требует написания дополнительного кода для создания методов доступа к полям объекта.
2. Усложнение кода: инкапсуляция может увеличить сложность кода и усложнить его понимание, особенно при работе с большим количеством объектов.
3. Ограничение доступа: инкапсуляция может ограничить доступ к определенным полям объекта, что может усложнить его использование.

В целом, преимущества инкапсуляции превышают ее недостатки, поэтому инкапсуляция является важным принципом ООП.

Программно реализовать ошибки и их исправление из раздела 9.3.3;

**Ошибка № 1: Использование геттеров и сеттеров для public-полей**.