**Лабораторная работа №1**

**ИНКАНСУЛЯЦИЯ**

**Цель работы:** Научиться работать с механизмом сокрытия компонентов программы.

Задание:

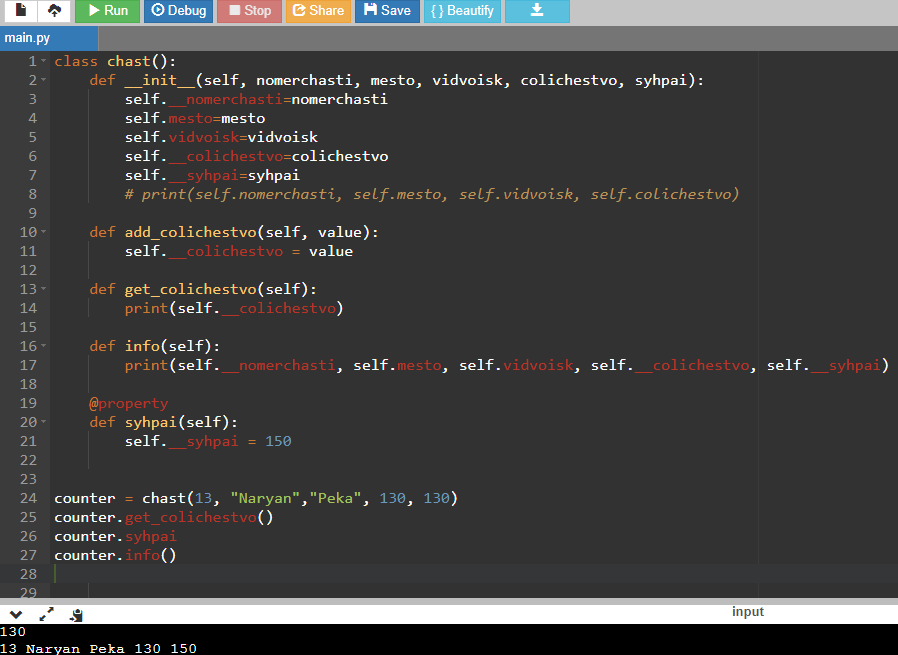


Рис. 1 - Преобразовал атрибуты и методы своих классов, согласно варианта индивидуального задания, в приватные для предотвращения переопределения и использовал в разработанных классах свойства.

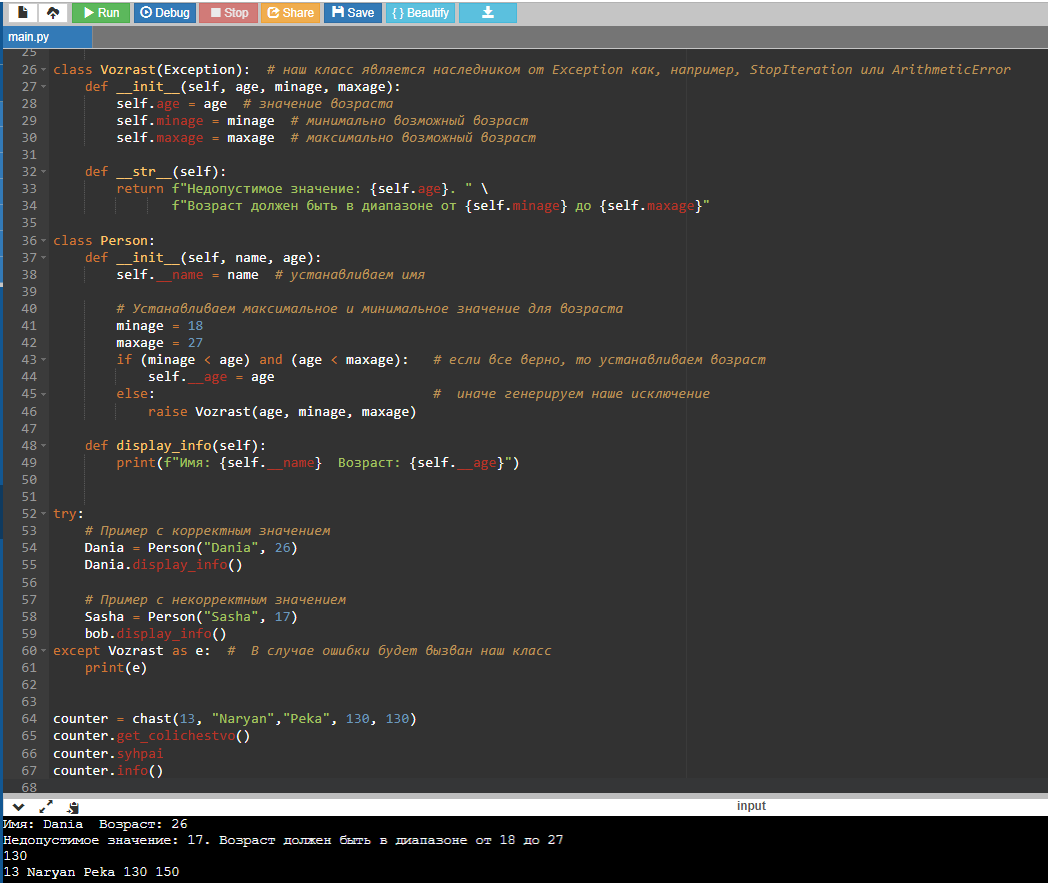


Рис. 2 - Создайте собственные классы исключений с атрибутами и используйте их.

**Контрольные вопросы**

**1. Что представляет собой инкапсуляция?**

Инкапсуляция – это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя.

**2. Какие уровни сокрытия можно выделить в языках программирования; в чем их различия?**

Private - Раздел private устанавливает самые жесткие ограничения на видимость. Поля класса, как правило, помещают именно в этот раздел. Также сюда помещают элементы класса, доступ к которым из других модулей должен быть запрещен.

Protected - Раздел protected содержит поля и методы, доступные для потомков класса. Этот уровень инкапсуляции крайне важен при создании расширений для существующих классов.

Public - Раздел public содержит поля и методы, доступ к которым может быть осуществлен из любого места программы, в том числе и из других модулей.

Internal - Раздел internal позволяет обращаться к объектам, их полям и методам во всех файлах (например, библиотеках), с использованием которых компилируется программа. Если при описании класса не указан никакой из атрибутов, то для всех полей и методов класса подразумевается internal.

**3. Каким образом инкапсуляция реализована в языке Python?**

Инкапсуляция в Python дает нам возможность получить доступ к необходимым переменным, не предоставляя программе полноценный доступ к любой из этих переменных.

**4. Каким образом инкапсуляция реализована в языке C#?**

Инкапсуляция реализуется с использованием спецификаторов доступа. Спецификатор доступа определяет область видимости члена класса. C # поддерживает следующие спецификации доступа:

* общественного
* Частный
* защищенный
* внутренний
* Защищенный внутренний

**Вывод:** Научились работать с механизмом сокрытия компонентов программы.