|  |  |
| --- | --- |
|  |  |
| МИНОБРНАУКИ РОССИИ |  |
| Федеральное государственное бюджетное образовательное учреждение высшего образования **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |  |

**Институт информационных технологий**

КАФЕДРА ИНСТРУМЕТНАЛЬНОГО И ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИиППО)

Практические РАБОТы

по дисциплине «Программирование на языка Джава»

Выполнил студент группы ИКБО-21-23 *Валеев А. Р.*

*(учебная группа)*

Принял старший преподаватель *Рачков А.В.*

Практические работы работа выполнены «\_\_\_»\_\_\_\_\_\_\_2024г.

«Зачтено» «\_\_\_»\_\_\_\_\_\_\_2024г.

Москва 2024

**СОДЕРЖАНИЕ**

[Практическая работа № 1 3](#_Toc178390397)

[Практическая работа № 2 7](#_Toc178390398)

# **Практическая работа № 1**

**Цель работы**

Разработать и протестировать программы на языке Java, которые осуществляют конвертацию суммы денег из китайских юаней в российские рубли по заданному курсу. В рамках выполнения практической работы будет реализована базовая программа конвертации, а также улучшенная версия с использованием структур выбора для обработки различных значений входной валюты.

**Теоретическое введение**

Java предоставляет восемь примитивных типов данных, среди которых числовые типы — float и double. Тип данных double имеет больший диапазон значений, чем float, и соответствует стандарту IEEE 754, который использует 32 бита для float и 64 бита для double.

Основные арифметические операторы: сложение (+), вычитание (-), умножение (\*), деление (/) и остаток от деления (%). Составные операторы присваивания, например, count += 1, упрощают операции изменения значения переменной.

В Java возможно выполнять бинарные операции с операндами разных типов: целое число автоматически преобразуется в число с плавающей точкой. Присвоение значения меньшего диапазона большему — допустимо, но сужение типа требует явного приведения. Приведение типов включает в себя расширение (от меньшего к большему) и сужение (от большего к меньшему). В Java при приведении double к int дробная часть отбрасывается. Условия if-else могут содержать вложенные if, и вложенность не ограничена.

**Выполнение практической работы**

*Задание 1:*

Напишите программу, которая конвертирует сумму денег из китайских юаней в российские рубли по курсу покупки 11.91.

*Решение:*

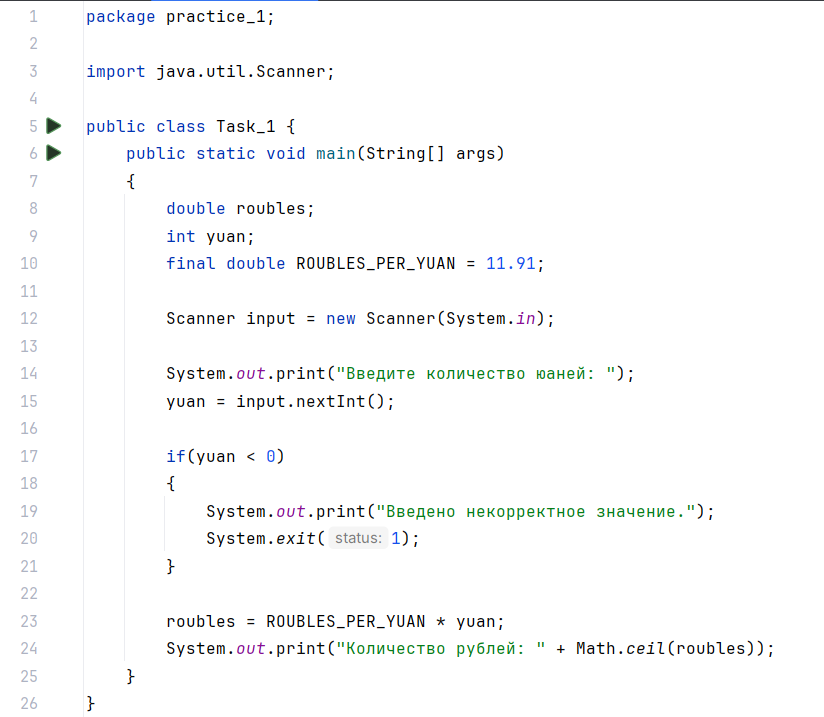
Решение данной задачи представлено на рисунке 1.

Рисунок 1 – Решение задачи 1

На рисунке 2 представлен результат тестирования программы.

Рисунок 2 – Результат тестирования программы

*Задание 2:*

Перепишите программу, которая конвертирует сумму денег из китайских юаней в российские рубли по курсу покупки 11.91, добавив структуру выбора для принятия решений об окончаниях входной валюты в зависимости от ее значения.

*Решение:*

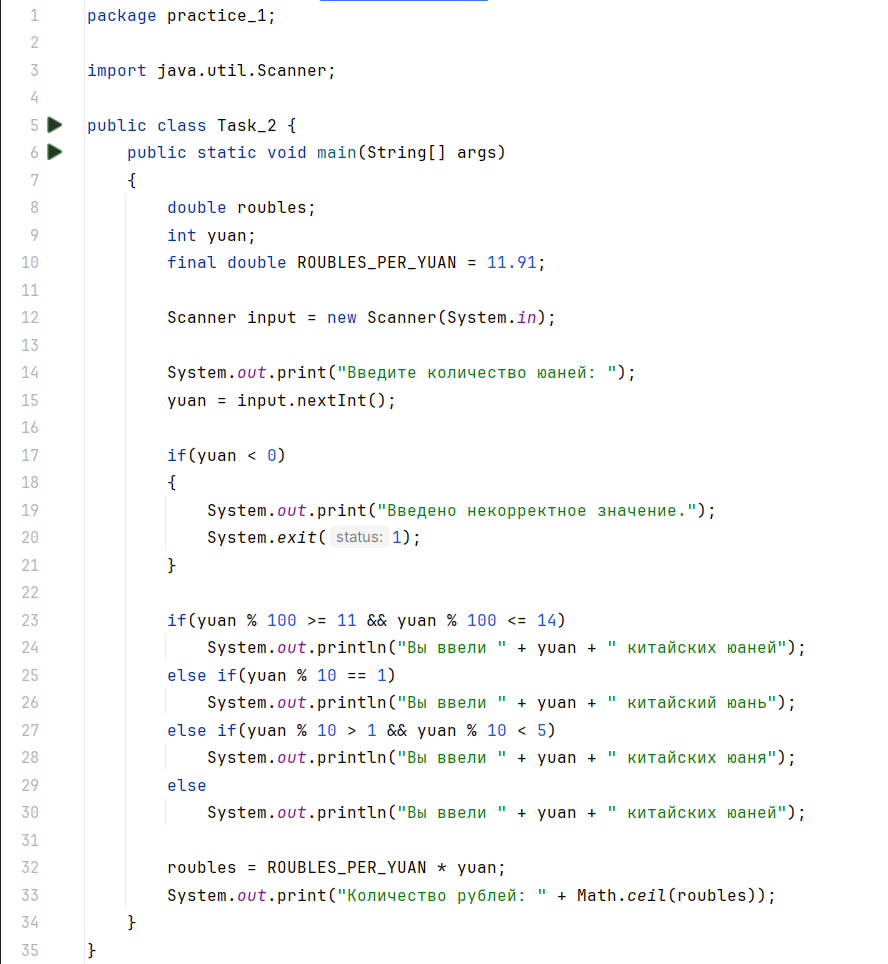
Решение данной задачи представлено на рисунке 3.

Рисунок 3 – Решение задачи 2

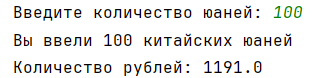
 На рисунке 4 представлен результат тестирования программы.

Рисунок 4 – Результат тестирования программы

**Вывод**

В ходе выполнения практической работы были получены навыки работы с языком Java. Были написаны программы конвертации валют из китайских юаней в рубли в соответствии с введенной суммой в юанях. Также были получены навыки работы с виртуальной средой разработки IntelliJ IDEA.

# **Практическая работа № 2**

**Цель работы**

Изучение и практическое применение принципов объектно-ориентированного программирования (ООП) на языке Java путем создания классов, методов, конструкторов и работы с объектами.

**Теоретическое введение**

Объектно-ориентированное программирование — это технология, направленная на разработку многократно используемого программного обеспечения. Основным элементом ООП является класс, который определяет свойства (атрибуты) и поведение (методы) объектов. Объекты представляют сущности реального мира и являются экземплярами классов.

Класс — это шаблон, определяющий поля данных и методы для объекта. Экземпляр класса создается с помощью конструктора, который инициализирует объект. Конструкторы могут быть перегружены, что позволяет создавать объекты с различными начальными значениями полей данных.

Основные понятия:

1. **Состояние объекта** — представлено полями данных, которые характеризуют объект. Например, у класса Circle это может быть поле radius.
2. **Поведение объекта** — определяется методами. Например, методы getArea() и getPerimeter() для класса Square.
3. **Конструктор** — специальный метод, вызываемый при создании объекта для его инициализации. Конструктор не имеет возвращаемого типа и должен иметь то же имя, что и класс.

**Выполнение практической работы**

*Задание 1:*

Напишите программу, в которой создается класс Car. В данном классе должны быть обозначены следующие поля: String model, String license, String color, int year – модель автомобиля, номер автомобиля, цвет автомобиля и год выпуска соответственно. Класс должен содержать три конструктора, один конструктор, который включает в себя все поля класса, один конструктор по умолчанию, один включает поля по выбору студента.

*Решение:*

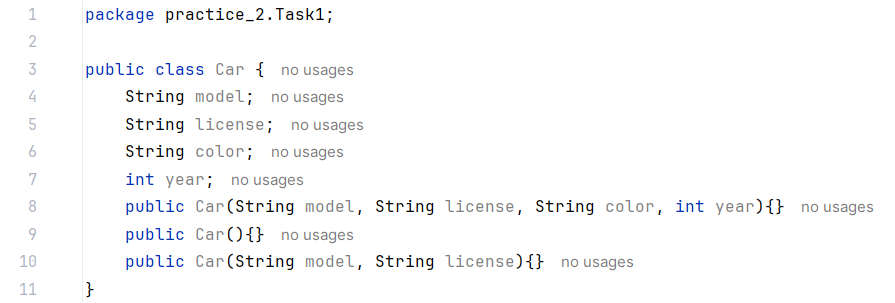
Решение данной задачи представлено на рисунке 5.

Рисунок 5 – Решение задачи 1

*Задание 2:*

В отдельном классе Main создайте экземпляры классов (объекты), используя различные конструкторы, реализованные в задаче #1. Создайте в классе метод To\_String(), который будет выводить значения полей экземпляров класса. Проверьте работу созданного метода, вызвав его у объекта. Дополните класс методами для получения и установки значений для всех полей (геттерами и сеттерами). Создайте метод класса, который будет возвращать возраст автомобиля, вычисляющийся от текущего года, значение текущего года допускается сделаться константным.

*Решение:*

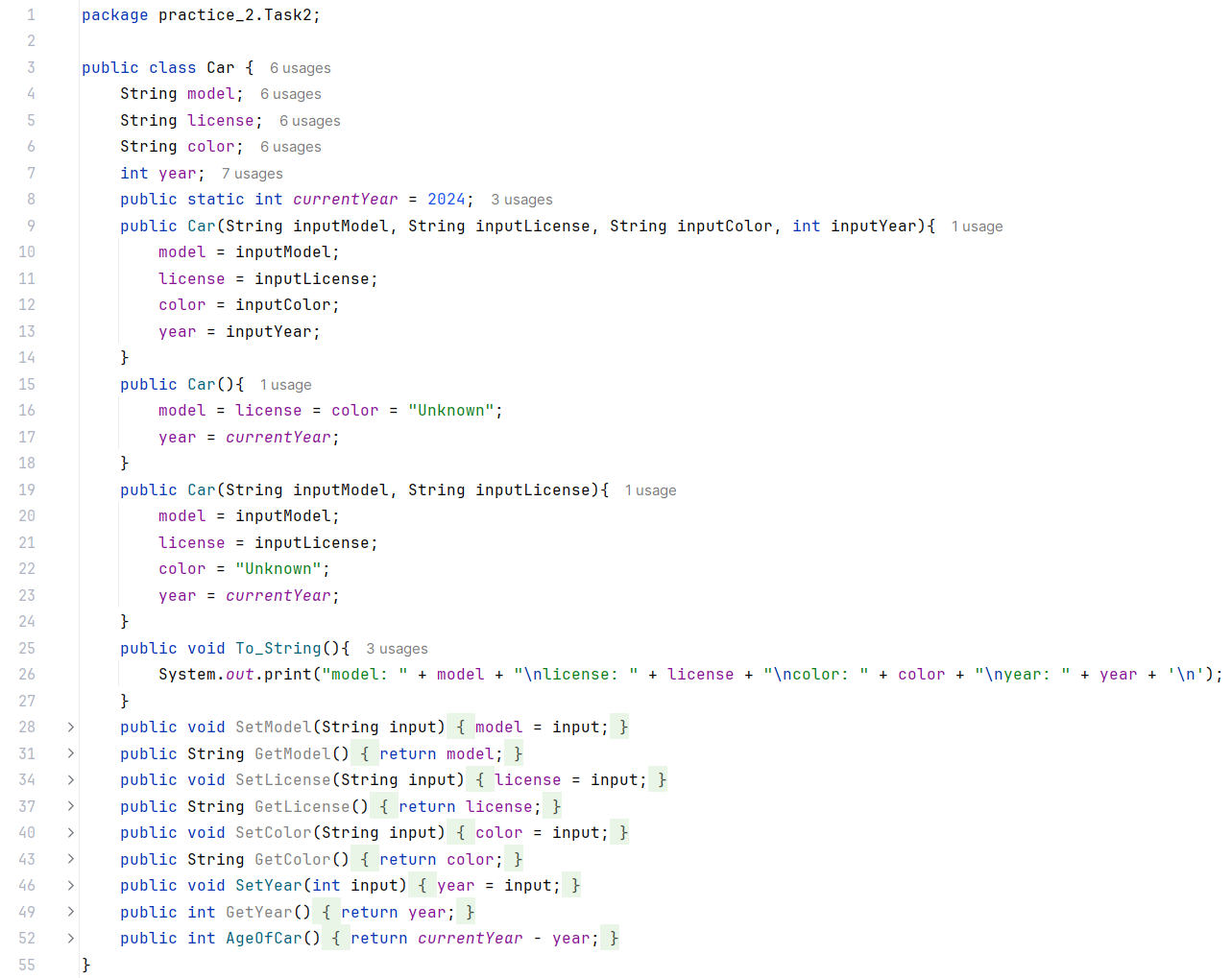
Решение данной задачи представлено на рисунках 6 и 7.

Рисунок 6 – Решение задачи 2 часть 1

Рисунок 7 – Решение задачи 2 часть 2

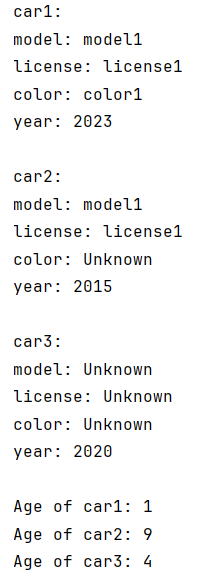
На рисунке 8 представлен результат тестирования программы.

Рисунок 8 – Результат тестирования программы задачи 2

**Вывод**

В результате выполнения работы были изучены основы объектно-ориентированного программирования на языке Java, включая создание классов, объектов, методов и конструкторов. Это позволило закрепить навыки работы с основными концепциями ООП и их применением на практике.