



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий

КАФЕДРА ИНСТРУМЕНТАЛЬНОГО И ПРИКЛАДНОГО
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ (ИиППО)

ПРАКТИЧЕСКИЕ РАБОТЫ
ПО ДИСЦИПЛИНЕ «Программирование на языке Джава»

Выполнил студент группы ИКБО-21-23
(учебная группа)

Валеев А. Р.

Принял старший преподаватель

Рачков А.В.

Практические работы работа выполнены «__»_____2024г.

«Зачтено» «__»_____2024г.

Москва 2024

СОДЕРЖАНИЕ

Практическая работа № 1	3
Практическая работа № 2	7

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.

```
1 package practice_1;
2
3 import java.util.Scanner;
4
5 public class Task_1 {
6     public static void main(String[] args)
7     {
8         double roubles;
9         int yuan;
10        final double ROUBLES_PER_YUAN = 11.91;
11
12        Scanner input = new Scanner(System.in);
13
14        System.out.print("Введите количество юаней: ");
15        yuan = input.nextInt();
16
17        if(yuan < 0)
18        {
19            System.out.print("Введено некорректное значение.");
20            System.exit(status: 1);
21        }
22
23        roubles = ROUBLES_PER_YUAN * yuan;
24        System.out.print("Количество рублей: " + Math.ceil(roubles));
25    }
26 }
```

Рисунок 1 – Решение задачи 1

На рисунке 2 представлен результат тестирования программы.

```
Введите количество юаней: 100
Количество рублей: 1191.0
```

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.

```
1  package practice_1;
2
3  import java.util.Scanner;
4
5  public class Task_2 {
6      public static void main(String[] args)
7      {
8          double roubles;
9          int yuan;
10         final double ROUBLES_PER_YUAN = 11.91;
11
12         Scanner input = new Scanner(System.in);
13
14         System.out.print("Введите количество юаней: ");
15         yuan = input.nextInt();
16
17         if(yuan < 0)
18         {
19             System.out.print("Введено некорректное значение.");
20             System.exit( status: 1);
21         }
22
23         if(yuan % 100 >= 11 && yuan % 100 <= 14)
24             System.out.println("Вы ввели " + yuan + " китайских юаней");
25         else if(yuan % 10 == 1)
26             System.out.println("Вы ввели " + yuan + " китайский юань");
27         else if(yuan % 10 > 1 && yuan % 10 < 5)
28             System.out.println("Вы ввели " + yuan + " китайских юаня");
29         else
30             System.out.println("Вы ввели " + yuan + " китайских юаней");
31
32         roubles = ROUBLES_PER_YUAN * yuan;
33         System.out.print("Количество рублей: " + Math.ceil(roubles));
34     }
35 }
```

Рисунок 3 – Решение задачи 2

На рисунке 4 представлен результат тестирования программы.

Введите количество юаней: 100

Вы ввели 100 китайских юаней

Количество рублей: 1191.0

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.

```
1 package practice_2.Task1;
2
3 public class Car { no usages
4     String model; no usages
5     String license; no usages
6     String color; no usages
7     int year; no usages
8     public Car(String model, String license, String color, int year){} no usages
9     public Car(){ no usages
10    public Car(String model, String license){ no usages
11 }
```

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.

```
1 package practice_2.Task2;
2
3 public class Car { 6 usages
4     String model; 6 usages
5     String license; 6 usages
6     String color; 6 usages
7     int year; 7 usages
8     public static int currentYear = 2024; 3 usages
9     public Car(String inputModel, String inputLicense, String inputColor, int inputYear){ 1 usage
10         model = inputModel;
11         license = inputLicense;
12         color = inputColor;
13         year = inputYear;
14     }
15     public Car(){ 1 usage
16         model = license = color = "Unknown";
17         year = currentYear;
18     }
19     public Car(String inputModel, String inputLicense){ 1 usage
20         model = inputModel;
21         license = inputLicense;
22         color = "Unknown";
23         year = currentYear;
24     }
25     public void To_String(){ 3 usages
26         System.out.print("model: " + model + "\nlicense: " + license + "\ncolor: " + color + "\nyear: " + year + '\n');
27     }
28 > public void SetModel(String input) { model = input; }
31 > public String GetModel() { return model; }
34 > public void SetLicense(String input) { license = input; }
37 > public String GetLicense() { return license; }
40 > public void SetColor(String input) { color = input; }
43 > public String GetColor() { return color; }
46 > public void SetYear(int input) { year = input; }
49 > public int GetYear() { return year; }
52 > public int AgeOfCar() { return currentYear - year; }
55 }
```

Рисунок 6 – Решение задачи 2 часть 1

```

1 package practice_2.Task2;
2
3 public class Main {
4     public static void main(String[] args) {
5         Car car1 = new Car( inputModel: "model1", inputLicense: "license1", inputColor: "color1", inputYear: 2023);
6         Car car2 = new Car( inputModel: "model1", inputLicense: "license1");
7         Car car3 = new Car();
8
9         car2.SetYear( input: 2015);
10        car3.SetYear( input: 2020);
11
12        System.out.println("car1:");
13        car1.To_String();
14        System.out.println("\ncar2:");
15        car2.To_String();
16        System.out.println("\ncar3:");
17        car3.To_String();
18        System.out.print('\n');
19
20        System.out.println("Age of car1: " + car1.AgeOfCar());
21        System.out.println("Age of car2: " + car2.AgeOfCar());
22        System.out.println("Age of car3: " + car3.AgeOfCar());
23    }
24 }

```

Рисунок 7 – Решение задачи 2 часть 2

На рисунке 8 представлен результат тестирования программы.

```

car1:
model: model1
license: license1
color: color1
year: 2023

```

```

car2:
model: model1
license: license1
color: Unknown
year: 2015

```

```

car3:
model: Unknown
license: Unknown
color: Unknown
year: 2020

```

```

Age of car1: 1
Age of car2: 9
Age of car3: 4

```

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.