

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления» Кафедра
ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков
программирования» Отчёт по домашнему
заданию

Выполнил:
студент группы ИУ5-35Б
Александров Георгий

Проверил: преподаватель каф. ИУ5
Нардид Анатолий Николаевич

Москва, 2025 г.

О работе

Эта программа — умный калькулятор валют. Она берёт свежие курсы с бирж в реальном времени. Ты вводишь сумму, выбираешь валюты и сразу видишь результат. Всё считается автоматически при каждом изменении.

Программа показывает точный курс на текущий момент. Она работает даже без интернета, используя запасные данные. Интерфейс простой и понятный — только то, что нужно. Ты просто вводишь цифры и получаешь ответ. Ничего лишнего, только быстрая конвертация. Это удобный инструмент для мгновенных расчётов.

Текст программы

app.py

```
from flask import Flask, render_template, request, jsonify
import requests
from datetime import datetime

app = Flask(__name__)

API_KEY = "03f26f207494211a9d3cdxfc"
BASE_URL = f"https://v6.exchangerate-api.com/v6/{API_KEY}"

def get_exchange_rate(base_currency, target_currency):
    try:
        url = f'{BASE_URL}/pair/{base_currency}/{target_currency}'
        response = requests.get(url, timeout=5)
        data = response.json()

        if data['result'] == 'success':
            return {
                'rate': data['conversion_rate'],
                'success': True,
                'timestamp': datetime.now().isoformat()
            }
    except:
        pass

fallback = {
    'USD': {'EUR': 0.92, 'RUB': 92.5, 'GBP': 0.79, 'JPY': 148.3},
    'EUR': {'USD': 1.09, 'RUB': 100.5, 'GBP': 0.86, 'JPY': 161.0},
    'RUB': {'USD': 0.0108, 'EUR': 0.00995, 'GBP': 0.0085, 'JPY': 1.60},
    'GBP': {'USD': 1.27, 'EUR': 1.16, 'RUB': 117.5, 'JPY': 187.5},
}
```

```

if base_currency in fallback and target_currency in fallback[base_currency]:
    return {
        'rate': fallback[base_currency][target_currency],
        'success': True,
        'timestamp': datetime.now().isoformat(),
        'note': 'кэш'
    }

return {'success': False, 'rate': 1.0}

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/api/rates')
def get_all_rates():
    base = request.args.get('base', 'USD')

    try:
        url = f'{BASE_URL}/latest/{base}'
        response = requests.get(url, timeout=5)
        data = response.json()

        if data['result'] == 'success':
            popular = ['USD', 'EUR', 'GBP', 'JPY', 'RUB', 'CAD', 'AUD', 'CHF']
            rates = {}
            for currency in popular:
                if currency in data['conversion_rates'] and currency != base:
                    rates[currency] = data['conversion_rates'][currency]

            return jsonify({
                'success': True,
                'base': base,
                'rates': rates,
                'date': data.get('time_last_update_utc', "")
            })
    except:
        pass

    return jsonify({'success': False, 'rates': {}})

@app.route('/api/convert')
def convert():
    from_curr = request.args.get('from', 'USD')

```

```

to_curr = request.args.get('to', 'EUR')
amount = float(request.args.get('amount', 1))

try:
    url = f'{BASE_URL}/pair/{from_curr}/{to_curr}/{amount}'
    response = requests.get(url, timeout=5)
    data = response.json()

    if data['result'] == 'success':
        return jsonify({
            'success': True,
            'from': from_curr,
            'to': to_curr,
            'amount': amount,
            'rate': data['conversion_rate'],
            'result': data['conversion_result']
        })
    except:
        pass

rate_data = get_exchange_rate(from_curr, to_curr)
if rate_data['success']:
    result = amount * rate_data['rate']
    return jsonify({
        'success': True,
        'from': from_curr,
        'to': to_curr,
        'amount': amount,
        'rate': rate_data['rate'],
        'result': result,
        'note': 'кэш'
    })

return jsonify({'success': False, 'error': 'Ошибка'})

if __name__ == '__main__':
    app.run(debug=True, port=5000)

```

templates/index.html

```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```
<title>Калькулятор валют</title>
<style>
  * { margin: 0; padding: 0; box-sizing: border-box; }
  body {
    font-family: Arial, sans-serif;
    background: linear-gradient(135deg, #1a2980 0%, #26d0ce 100%);
    min-height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 20px;
  }
  .container {
    background: white;
    border-radius: 20px;
    padding: 40px;
    box-shadow: 0 20px 50px rgba(0,0,0,0.3);
    width: 100%;
    max-width: 500px;
  }
  h1 {
    color: #1a2980;
    text-align: center;
    margin-bottom: 30px;
  }
  .input-group {
    display: flex;
    align-items: center;
    gap: 10px;
    margin: 20px 0;
  }
  input {
    flex: 1;
    padding: 15px;
    border: 2px solid #ddd;
    border-radius: 10px;
    font-size: 18px;
    outline: none;
  }
  input:focus {
    border-color: #1a2980;
  }
  select {
    padding: 15px;
```

```
border: 2px solid #ddd;
border-radius: 10px;
font-size: 16px;
min-width: 120px;
outline: none;
}
.swap-btn {
background: #1a2980;
color: white;
border: none;
width: 50px;
height: 50px;
border-radius: 50%;
font-size: 20px;
cursor: pointer;
margin: 10px auto;
display: block;
}
.calc-btn {
background: linear-gradient(135deg, #1a2980 0%, #26d0ce 100%);
color: white;
border: none;
padding: 18px;
border-radius: 10px;
font-size: 18px;
font-weight: bold;
cursor: pointer;
width: 100%;
margin-top: 20px;
}
.result {
background: #f8f9fa;
padding: 25px;
border-radius: 15px;
margin-top: 25px;
text-align: center;
border: 2px solid #e9ecf;
}
.result-amount {
font-size: 36px;
font-weight: bold;
color: #1a2980;
margin: 10px 0;
}
```

```
.rate-info {
    color: #666;
    font-size: 16px;
    margin-top: 10px;
}

.status {
    text-align: center;
    margin-top: 20px;
    padding: 10px;
    background: #e8f5e9;
    border-radius: 8px;
    color: #2e7d32;
    font-size: 14px;
}

</style>
</head>
<body>
<div class="container">
    <h1> 💰 Калькулятор валют</h1>

    <div class="input-group">
        <input type="number" id="amount" value="100" min="0" step="0.01"
placeholder="Сумма">
        <select id="fromCurrency">
            <option value="USD">USD</option>
            <option value="EUR">EUR</option>
            <option value="RUB">RUB</option>
            <option value="GBP">GBP</option>
            <option value="JPY">JPY</option>
        </select>
    </div>

    <button class="swap-btn" onclick="swapCurrencies()">₦</button>

    <div class="input-group">
        <div style="flex: 1; text-align: center; font-size: 24px;">→</div>
        <select id="toCurrency">
            <option value="RUB">RUB</option>
            <option value="USD">USD</option>
            <option value="EUR">EUR</option>
            <option value="GBP">GBP</option>
            <option value="JPY">JPY</option>
        </select>
    </div>

```

```
<button class="calc-btn" onclick="calculate()">Рассчитать</button>

<div class="result" id="resultBox" style="display: none;">
  <div>Результат:</div>
  <div class="result-amount" id="resultAmount">0</div>
  <div class="rate-info">
    1 <span id="rateFrom">USD</span> = <span id="rateValue">0</span> <span
    id="rateTo">EUR</span>
  </div>
</div>

<div class="status" id="status">
  Загрузка курсов...
</div>
</div>

<script>
  let rates = {};

  async function loadRates() {
    try {
      const base = document.getElementById('fromCurrency').value;
      const response = await fetch(`/api/rates?base=${base}`);
      const data = await response.json();

      if (data.success) {
        rates = data.rates;
        document.getElementById('status').textContent =
          `Курсы обновлены • API: ${data.date} || 'активен'`;
        document.getElementById('status').style.background = '#e8f5e9';
        document.getElementById('status').style.color = '#2e7d32';
      } else {
        throw new Error();
      }
    } catch {
      rates = {};
      document.getElementById('status').textContent =
        'API временно недоступен • Используются кэшированные курсы';
      document.getElementById('status').style.background = '#fff3e0';
      document.getElementById('status').style.color = '#ef6c00';
    }
  }
}
```

```

async function calculate() {
    const amount = document.getElementById('amount').value;
    const from = document.getElementById('fromCurrency').value;
    const to = document.getElementById('toCurrency').value;

    if (!amount || amount <= 0) {
        document.getElementById('resultBox').style.display = 'none';
        return;
    }

    try {
        const response = await fetch(
            `/api/convert?from=${from}&to=${to}&amount=${amount}`
        );
        const data = await response.json();

        if (data.success) {
            document.getElementById('resultAmount').textContent =
                `${data.result.toFixed(2)} ${to}`;
            document.getElementById('rateFrom').textContent = from;
            document.getElementById('rateTo').textContent = to;
            document.getElementById('rateValue').textContent = data.rate.toFixed(4);
            document.getElementById('resultBox').style.display = 'block';
        }
    } catch {
        alert('Ошибка расчета');
    }
}

function swapCurrencies() {
    const from = document.getElementById('fromCurrency');
    const to = document.getElementById('toCurrency');
    const temp = from.value;
    from.value = to.value;
    to.value = temp;
    loadRates();
    calculate();
}

document.getElementById('amount').addEventListener('input', calculate);
document.getElementById('fromCurrency').addEventListener('change', function() {
    loadRates();
    calculate();
});

```

```
document.getElementById('toCurrency').addEventListener('change', calculate);

loadRates();
calculate();

</script>
</body>
</html>
```

Пример работы



