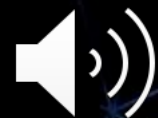


# Программа расчета налогового вычета на лечение для физических лиц

Подготовил Краснов Богдан Андреевич



## Немного о проекте

**С появлением данного веб-приложения вам больше не придется тратить огромное кол-во времени на расчет налогового вычета на лечение.**

Расходы:

Предыдущие вычеты:

Максимальная сумма лечения(в тысячах)

120

[РАССЧИТАТЬ](#)

[ENGLISH](#)

# Постановка задачи ?

1. Создание программы расчета налогового вычета на лечение для физических лиц
2. Полное соответствие ТЗ
3. Удобство использования веб-приложения
4. Интуитивно понятный интерфейс
5. Программа должна быть совместима с операционными системами Windows, MacOS и Linux
6. Программа должна иметь возможность выбора языка интерфейса (Русский, Английский).
7. Возможность изменения показателя «Лимит», при соблюдении условия:
  - если показатель не задан, то использовать значение 120 000;
  - если показатель = 0, то не использовать лимит в формуле расчета, то есть не сравнивать сумму лечения и лимит, всегда брать в расчет стоимость лечения;
  - если показатель > 0, то использовать значение нового лимита.



## Основные требования к программе

1

Учет видов налоговых вычетов, предусмотренных законодательством, таких как налоговый вычет на лечение.

2

Расчет налогового вычета в соответствии с действующим законодательством.

1.Необходимо произвести расчета налогового вычета без учета дохода физического лица по формуле:

**НВ = (Стоимость лечения \* 13%) -**  
сумма уже полученных вычетов на лечение за год.

3

Возможность сохранения результатов расчета в файле для последующего использования.

# 3 основных требования

01

## Требования к надежности

Программа должна быть стабильной и не выдавать ошибок при работе.

02

## Требования к производительности

Программа должна работать быстро и выдавать результат расчета за 5 секунд.

03

## Требования к техническому обеспечению

Для создания графического интерфейса можно использовать готовые библиотеки.

## Т р е б о в а н и я   к   и н т е р ф е й с у

1. Интерфейс программы должен быть интуитивно понятным и удобным в использовании.
2. Интерфейс программы должен содержать элементы для ввода данных, требуемых для расчета налогового вычета:

Стоимость лечения. Не менее 1 символа и не более 10-ти символов. Допустимы для ввода значения: только цифры и разделитель дробной части – точка(.). Точность ввода значений до 2 знаков после запятой.

Сумма уже полученных вычетов на лечение за год. Не менее 1 символа и не более 10-ти символов. Допустимы для ввода значения: только цифры и разделитель дробной части – точка(.). Точность ввода значений до 2 знаков после запятой.

3. Интерфейс должен содержать элементы для запуска процесса расчета «Рассчитать НВ».
4. Интерфейс должен содержать элементы для отражения результата расчёта.

Показать результат расчета как «Сумма налогового вычета **XX XXX рублей XX копеек**».

Ограничение. Если в результате расчета Сумма налогового вычета получилась отрицательной, то в результате показать

«Сумма налогового вычета **0 рублей 00 копеек**».

## Используемые технологии

На весь проект у меня ушло 10 часов и в процессе разработки я столкнулся с некоторыми трудностями, но смог их решить и в большинстве случаев мне помогла документация. В создании веб-приложения я использовал только базовые возможности языка JS, HTML, CSS.

```
1  var language = "ru";
2  var maxDeduction = 120000;
3
4  document.getElementById("calculate").addEventListener("click", function() {
5      var expensesInput = document.getElementById("expenses");
6      var expenses = parseFloat(expensesInput.value);
```

СИРИУС

# Анализ процесса

Буду показывать  
наиболее важные  
фрагменты кода



## Структура 1

`var language = "ru";` - объявляет переменную `language` и присваивает ей значение `"ru"`. Эта переменная указывает на текущий язык интерфейса.

`var maxDeduction = 120000;` - объявляет переменную `maxDeduction` и присваивает ей значение `120000`. Это максимальное значение налогового вычета.

`document.getElementById("calculate").addEventListener("click", function() {...})` - добавляет обработчик событий клика на кнопку "Calculate". Когда пользователь нажимает на эту кнопку, выполняется функция, которая рассчитывает налоговый вычет и выводит результат на страницу.

```
1  var language = "ru";
2  var maxDeduction = 120000;
3
4  document.getElementById("calculate").addEventListener("click", function() {
5      var expensesInput = document.getElementById("expenses");
6      var expenses = parseFloat(expensesInput.value);
7      if (isNaN(expenses) || expenses < 1 || expenses > 999999999) {
8          alert(getLocalizedString("Expenses must be between 1 and 10 characters."));
9      } else if (expensesInput.value.split(".").length > 2) {
10         alert(getLocalizedString("Invalid format for expenses."));
11     } else if (expensesInput.value.indexOf(".") !== -1 && expensesInput.value.split(".")[1].length > 2) {
12         alert(getLocalizedString("Precision should be no more than 2 decimal places."));
13     } else {
14         var previousDeductionsInput = document.getElementById("previous-deductions");
15         var previousDeductions = parseFloat(previousDeductionsInput.value) || 0;
16         if (isNaN(previousDeductions) || previousDeductions < 0 || previousDeductions > 999999999) {
17             alert(getLocalizedString("Previous deductions must be between 0 and 10 characters."));
18         } else {
19             if (maxDeduction === 0) {
20                 maxDeduction = 999999999
21             }
22             var remainingDeduction = maxDeduction - previousDeductions;
23             expenses = Math.min(expenses, maxDeduction);
24             var deduction = Math.min(expenses * 0.13, remainingDeduction) - previousDeductions;
25             var rubles = Math.floor(Math.max(deduction, 0));
26             var kopecks = Math.abs(Math.round((deduction - rubles) * 100));
27             if (kopecks === 100) {
28                 rubles += 1;
29                 kopecks = 0;
30             }
31         }
32     }
33 }
```

## Структура 2

`document.getElementById("save").addEventListener("click", function() {...}` - добавляет обработчик событий клика на кнопку "Save". Когда пользователь нажимает на эту кнопку, выполняется функция, которая сохраняет результат расчета в текстовый файл.

`document.getElementById("expenses").addEventListener("input", function() {...}` - добавляет обработчик событий ввода для поля "Expenses". Когда пользователь вводит значение в это поле, выполняется функция, которая проверяет корректность введенных данных и выводит сообщение об ошибке при необходимости.

```
document.getElementById("save").addEventListener("click", function() {
    var expensesInput = document.getElementById("expenses");
    var expenses = parseFloat(expensesInput.value);
    var previousDeductionsInput = document.getElementById("previous-deductions");
    var previousDeductions = parseFloat(previousDeductionsInput.value) || 0;
    var maxDeduction = 120000;
    var remainingDeduction = maxDeduction - previousDeductions;
    expenses = Math.min(expenses, maxDeduction);
    var deduction = Math.min(expenses * 0.13, remainingDeduction) - previousDeductions;
    console.log(deduction);
    var rubles = Math.floor(deduction);
    var kopecks = Math.round((deduction - rubles) * 100);
    var result = "Сумма расходов на лечение: " + expenses + "\nНалоговый вычет составляет " + rubles + " рублей " + kopecks;
    if (deduction < 0){
        var result = "Сумма расходов на лечение: " + expenses + "\nНалоговый вычет составляет 0 рублей 0 копеек";
    }
    var blob = new Blob([result], {type: "text/plain;charset=utf-8"});
    saveAs(blob, "result.txt");
});

document.getElementById("expenses").addEventListener("input", function() {
    var expensesInput = document.getElementById("expenses");
    expensesInput.value = expensesInput.value.replace(/^[^0-9.]/g, "");
    if (expensesInput.value.split(".").length > 2) {
        alert(getLocalizedString("Invalid format for expenses."));
    } else if (expensesInput.value.indexOf(".") !== -1 && expensesInput.value.split(".")[1].length > 2) {
        alert(getLocalizedString("Precision should be no more than 2 decimal places."));
    }
});
```

## Структура 3

`document.getElementById("previous-deductions").addEventListener("input", function() {...})` - добавляет обработчик событий ввода для поля "Previous deductions". Когда пользователь вводит значение в это поле, выполняется функция, которая проверяет корректность введенных данных и выводит сообщение об ошибке при необходимости.

`document.getElementById("language-selector").addEventListener("click", function() {...})` - добавляет обработчик событий клика на кнопку "Language selector". Когда пользователь нажимает на эту кнопку, выполняется функция, которая меняет язык интерфейса.

```
document.getElementById("previous-deductions").addEventListener("input", function() {
    var previousDeductionsInput = document.getElementById("previous-deductions");
    previousDeductionsInput.value = previousDeductionsInput.value.replace(/^[^0-9.]/g, "");
    if (previousDeductionsInput.value.split("").length > 2) {
        alert(getLocalizedString("Invalid format any previous deductions."));
    } else if (previousDeductionsInput.value.indexOf(".") !== -1 && previousDeductionsInput.value.split(".")[1].length > 2) {
        alert(getLocalizedString("Precision should be no more than 2 decimal places."));
    }
});

document.getElementById("language-selector").addEventListener("click", function() {
    if (language === "ru") {
        language = "en";
        document.getElementById("language-selector").textContent = "Русский";
    } else {
        language = "ru";
        document.getElementById("language-selector").textContent = "English";
    }
    updateInterfaceLanguage();
});

function getLocalizedString(text) {
    if (language === "ru") {
```

## Структура 4

`function getLocalizedString(text) {...}` - функция, которая возвращает локализованный текст в зависимости от текущего языка интерфейса.

`function updateInterfaceLanguage() {...}` - функция, которая обновляет язык интерфейса, изменяя текст на элементах страницы в соответствии с текущим языком.

```
function getLocalizedString(text) {  
  if (language === "ru") {  
    switch(text) {  
      case "Expenses must be between 1 and 10 characters.":  
        return "Размер расходов должен составлять от 1 до 10 символов.";  
      case "Previous deductions must be between 0 and 10 characters.":  
        return "Предыдущие вычеты должны содержать от 0 до 10 символов.";  
      case "Invalid formatfor expenses.":  
        return "Недопустимый формат для расходов.";  
      case "Precision should be no more than 2 decimal places.":  
        return "Точность должна составлять не более 2 знаков после запятой.";  
      case "Tax deduction is ":  
        return "Налоговый вычет - ";  
      case " kopecks.":  
        return " копеек.";  
      case " rubles ":  
        return " рублей "  
      default:  
        return text;  
    }  
  } else {  
    return text;  
  }  
}
```

## Frontend HTML

Создается HTML документ со стандартным DOCTYPE. В <head> подключается CSS файл на котором и будет основная верстка. После в теге <body> пишется вся структура сайта.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Calculator</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/FileSaver.js/2.0.5/FileSaver.min.js"></script>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Calculator</h1>
    <label for="expenses">Expenses:</label>
    <input type="text" id="expenses" name="expenses">
    <br>
    <label for="previous-deductions">Previous deductions:</label>
    <input type="text" id="previous-deductions" name="previous-deductions">
    <br>
    <label for="max-deduction">Max deduction:</label>
    <input type="range" id="max-deduction" name="max-deduction" min="0" max="100" value="50">
    <span id="max-deduction-value">50</span>
    <br>
    <button id="calculate">Calculate</button>
    <br>
    <div id="result"></div>
    <button id="save" style="display: none;">Save</button>
    <br>
    <button id="language-selector">English</button>
    <script src="script.js"></script>
  </body>
</html>
```

## Frontend CSS 1

Общая стилизация всей страницы

```
1  /* General styles */
2  * {
3      box-sizing: border-box;
4      margin: 0;
5      padding: 0;
6  }
7
8  body {
9      background-color: #f5f5f5;
10     font-family: 'Roboto', sans-serif;
11     font-size: 16px;
12     line-height: 1.5;
13     color: #333;
14 }
15
```

### Стилизация тегов <input>

```
input[type="text"] {  
  padding: 10px;  
  border: none;  
  border-radius: 5px;  
  font-size: 16px;  
  font-weight: 500;  
  color: ☐ #333;  
  background-color: ☐ #fff;  
  width: 100%;  
  max-width: 400px;  
  margin-bottom: 20px;  
  box-shadow: 0 0 10px ☐ rgba(0, 0, 0, 0.2);  
}  
  
input[type="text"]::placeholder {  
  color: ☐ #999;  
}
```

http://codepen.io



### Стилизация тегов <button>

```
button {  
  padding: 10px 20px;  
  border-radius: 5px;  
  font-size: 16px;  
  font-weight: 500;  
  text-transform: uppercase;  
  cursor: pointer;  
  transition: all 0.3s ease-in-out;  
  color: ■ #fff;  
  background-color: ■ #007bff;  
  border: none;  
  box-shadow: 0 0 5px ■ rgba(0, 0, 0, 0.2);  
}  
  
button:hover {  
  opacity: 0.8;  
}  
  
button:focus {  
  outline: none;  
  box-shadow: none;  
}
```



Стилизация id #language-selector

```
#language-selector {  
  margin-left: auto;  
  background-color: #007bff;  
  color: #fff;  
  border: none;  
  border-radius: 5px;  
  padding: 10px 20px;  
  cursor: pointer;  
  transition: all 0.3s ease-in-out;  
}  
  
#language-selector:hover {  
  background-color: #0062cc;  
}
```

## Анализ результатов

Из полученных результатов можно сделать вывод, что данные пункты выполнены и проект можно считать полностью рабочим!

1. Веб-приложение работает стабильно и не выдает ошибок.
2. Полное соответствие ТЗ
3. Удобство использования веб-приложения
4. Интуитивно понятный интерфейс
5. Программа совместима с операционными системами Windows, MacOS и Linux
6. Программа имеет возможность выбора языка интерфейса (Русский, Английский).
7. Программа работает быстро и выдает ответ меньше чем за 5 секунд.
8. Возможность сохранения результатов расчета в файле для последующего использования.

Веб-приложение получилось довольно быстрым.

Веб-приложение совместимо с ОС Windows, Linux

Данное веб-приложение работает исправно и готово для общественного пользования!

КРАСНОВ БОГДАН  
АНДРЕЕВИЧ

Почта:  
[bogd4n.krasnov@yandex.ru](mailto:bogd4n.krasnov@yandex.ru)

Телефон:  
+7 988 350 84 37

**We can do it!!!**

**Сириус навсегда**