**УРОК 11-16: Оператори за цикъл – преговор с допълнение**

**1. Въведение**

От урока за алгоритми знаем, че един от най-често използваните алгоритми е цикличният, при който няколко действия се повтарят многократно, докато е изпълнено даденото условие. Например, цикъл има при вход в платформа. Цикличният алгоритъм се реализира в програма чрез оператори за цикъл. Всеки език за програмиране притежава поне един такъв оператор. В езика С# има три оператора за цикъл – операторът for, операторът while и операторът do...while.

**2. Структура на цикъл** – състои се от 4 части

* *Инициализация* – задава се начална стойност на променливите, участващи в цикъла
* *Тяло на цикъла* – съдържа действията, които се повтарят многократно
* *Актуализация* – обновяване на стойностите на участващите променливи и подготовка за следващото изпълнение на тялото на цикъла
* *Условие за край на цикъла* – логически израз, чрез който се продължава или прекратява действието на цикъла

**3. Оператор за цикъл с брояч**

**а)** **Предназначение** – използва се за създаване на цикли, в които тялото на цикъла трябва да се изпълни за зададени стойности на променлива или променливи

**б)** **Синтаксис**

*for (<инициализация>; <условие>; <обновяване>)*

*{ <тяло на цикъла> }*

**в)** **Действие** - изпълнението на оператора започва с инициализация, в която се задава начална стойност на променливата, която управлява изпълнението на цикъла. След това започва повтаряне на следните три действия:

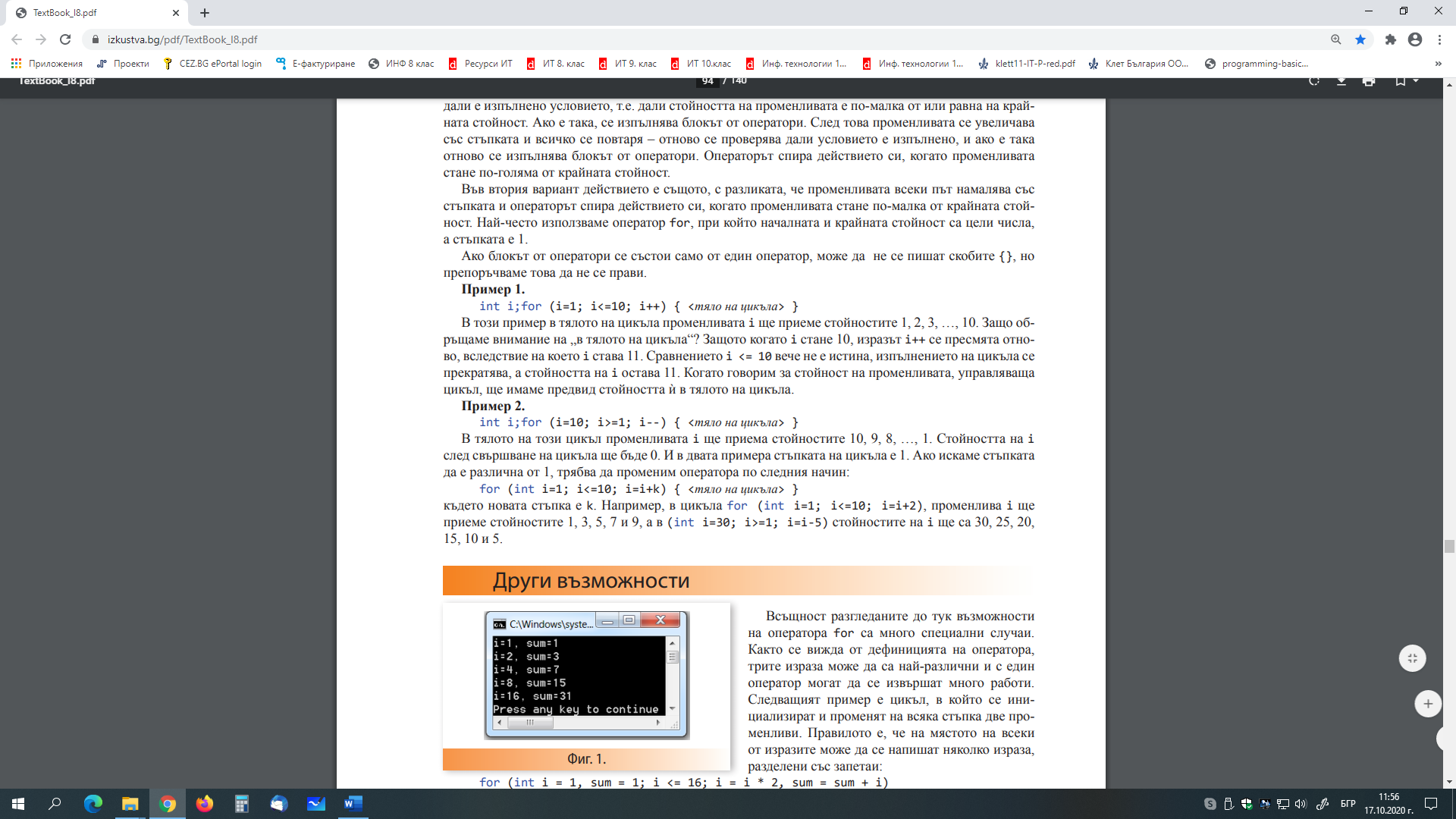
* изчислява се логическият израз. Ако стойността му е false – изпълнението на оператора за цикъл се прекратява. Ако стойността му е true, се продължава със следващите две стъпки;
* изпълнява се блокът от оператори, наричан *<тяло на цикъла>*;
* изчислява се изразът *<обновяване>,* чрез койтосе променя стойността на променливата, управляваща изпълнението на цикъла. След това отново се изчислява изразът и т.н.

Diagram

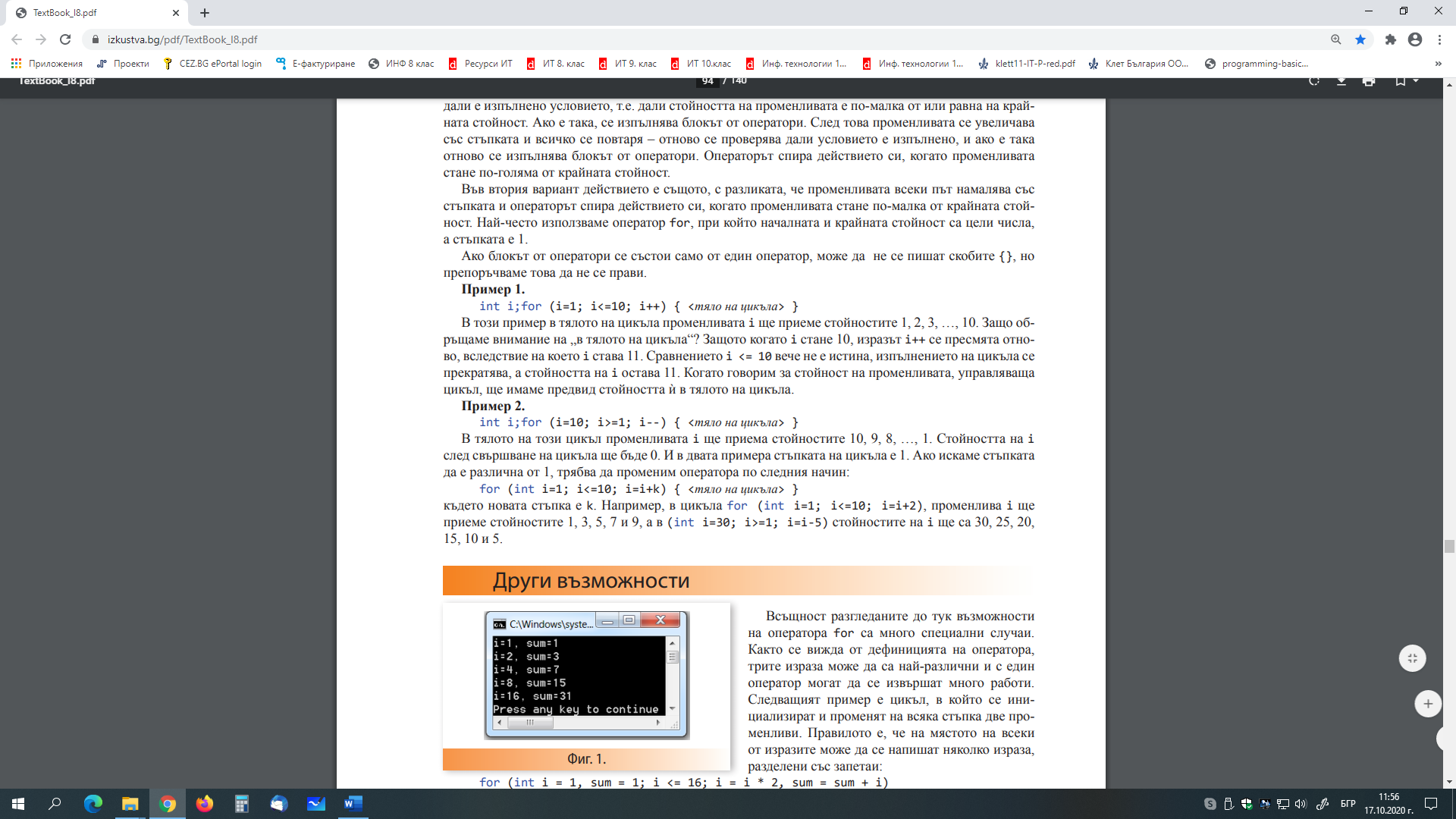
Description automatically generated

**г)** **Примери**

**Пример 1.**



**Пример 2.**



**4. Оператор за цикъл с предусловие**

**а)** **Синтаксис**

*<инициализация>;*

*while( <условие>)*

*{*

*<тяло на цикъла>*

*<обновяване>;*

*}*

**б)** **Действие** – Операторът while (англ. докато) проверява стойността на логическия израз и ако тя е true, тогава изпълнява тялото на цикъла. Операторът продължава да прави това, докато условието е изпълнено. Когато за пръв път стойността на израза стане false, операторът прекратява работа и предава управлението на следващия оператор. Тъй като в този оператор условието се проверява преди да се изпълни тялото на цикъла, за оператора казваме, че е с предусловие.

Diagram

Description automatically generated

**5. Оператор за цикъл с постусловие**

Разликата между него и оператора while е, че условието за край на цикъла се проверява след всяко изпълнение на тялото на цикъла. Затова в този случай казваме, че операторът за цикъл е с постусловие.

**а)** **Синтаксис**

*<инициализация>;*

*do*

*{*

*<тяло на цикъла>*

*<обновяване>;*

*}*

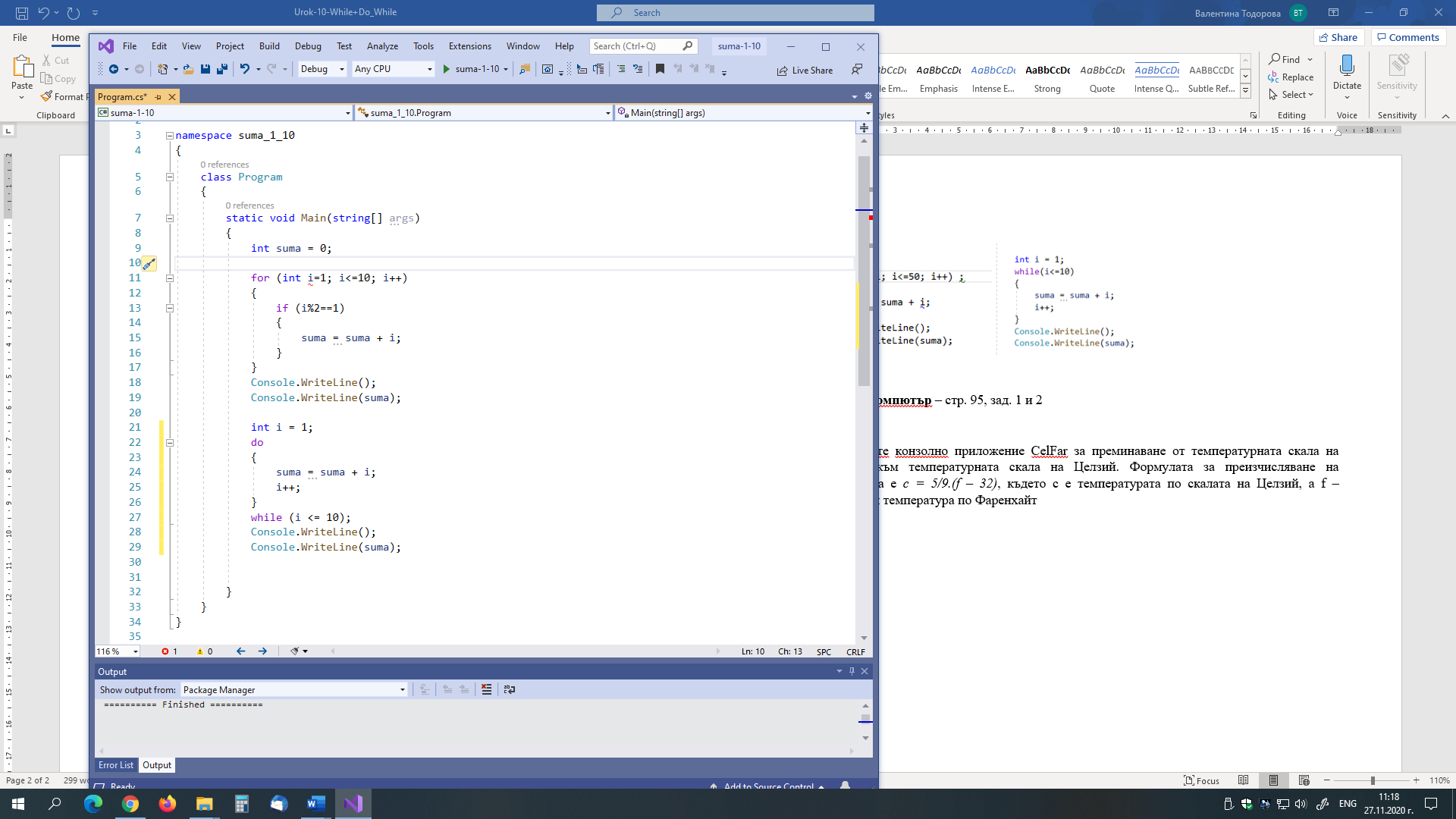
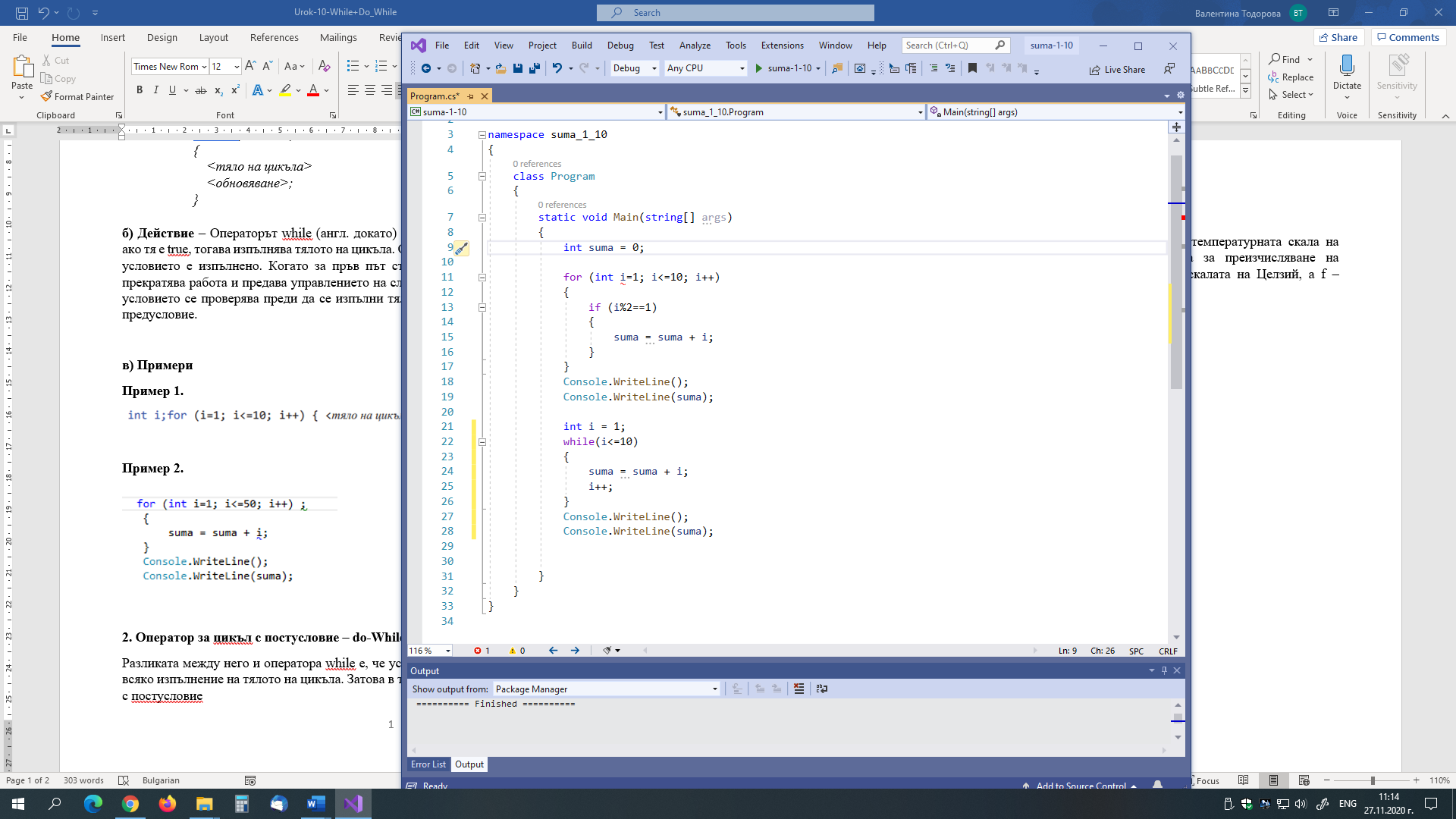
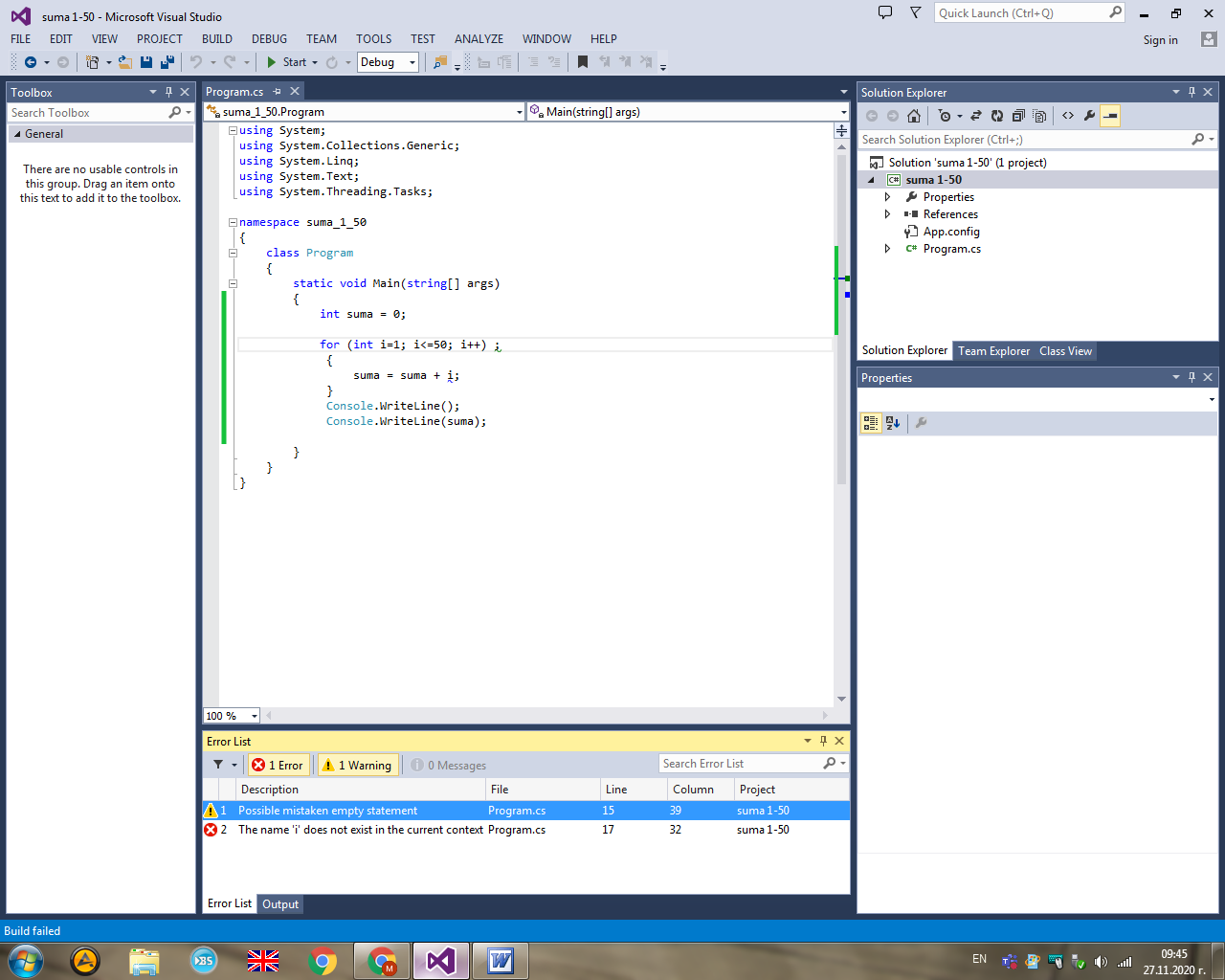
*while( <условие>)*

**б)** **Действие** – Операторът do-while първо изпълнява тялото на цикъла, след което проверява стойността на логическия израз. Ако тя е true отново се изпълнява тялото на цикъла. Операторът продължава да прави това, докато условието е изпълнено. Когато за пръв път стойността на израза стане false, операторът прекратява работа и предава управлението на следващия оператор.

Diagram

Description automatically generated

**в) Примери**

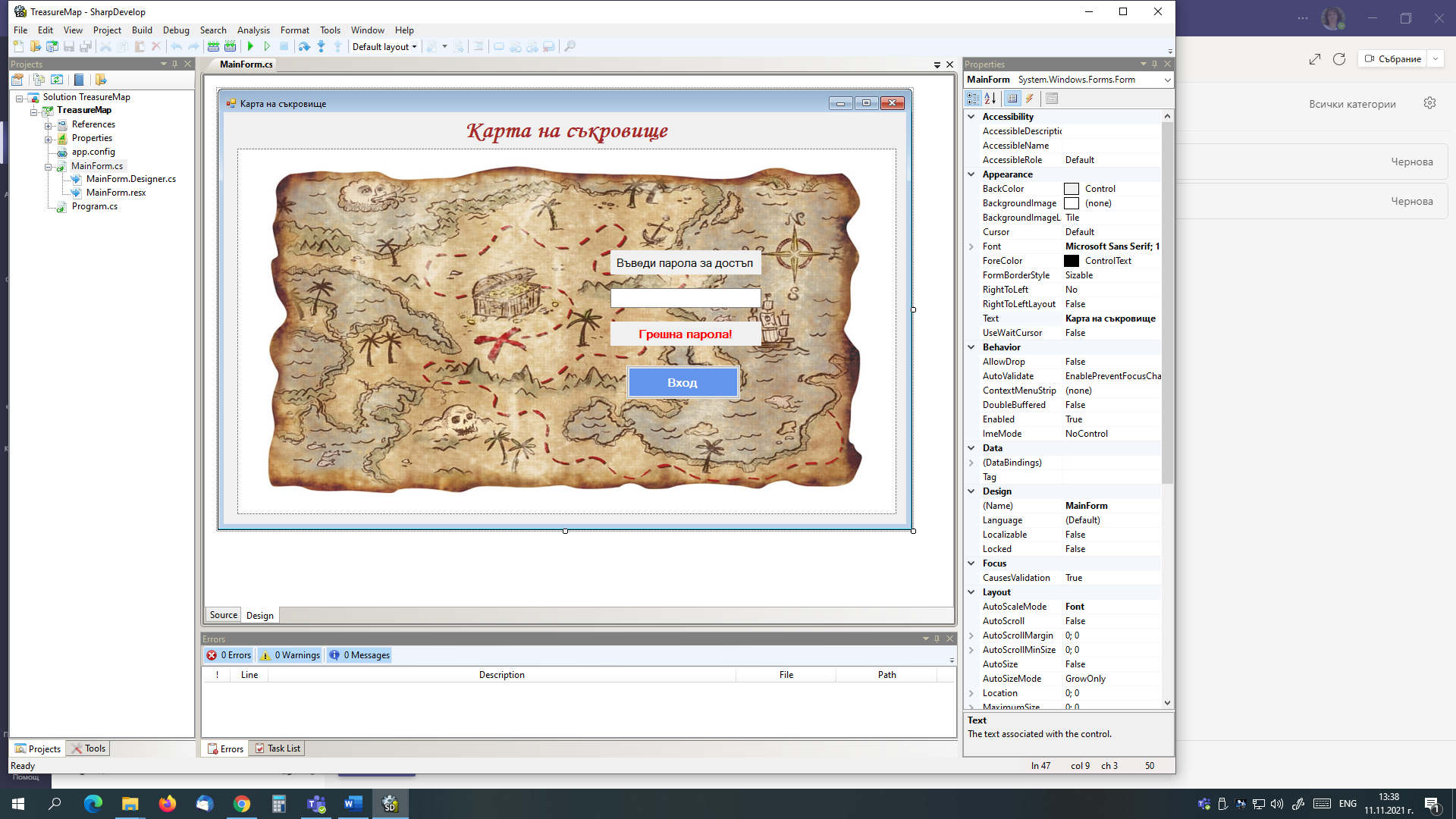


**6. Задачи**

**Задача 1.** Създайте графично приложение **ТreasureMap**, което ще „разкрие“ карта на съкровище, но само при правилно въведена парола за достъп.

**1.1.** Задайте за надпис в заглавната лента на формата текста „Карта на съкровища“

**1.2.** Вмъкнете необходимите елементи със съответните имена, за да получите следната форма:



* За полето за въвеждане паролата се използва елемент **textBox**, като в свойството ***PasswordChar*** се указва символа, който да замества въведените символи от паролата – например \*.
* Намерете в Интернет изображение с карта на съкровище и я вмъкнете като използвате елемента **pictureBox**.

**1.3.** За етикета, съдържащ текста „Въведи парола за достъп“ и за елемента с картата се задава стойност False на свойството Visible.

**1.4.** Създайте процедура за бутона „Вход“, която:

* при въвеждане на парола различна от 123456 да показва етикета „Грешна парола“ и да изчиства съдържанието на полето за паролата
* при въвеждане на правилната парола да се показват само етикета „Карта на съкровище“ и изображението с картата.

***Подсказка:*** Използвайте оператор за цикъл while.

***Решение:***

void **button1\_Click**(object sender, EventArgs e)  
        {  
            **int** pass=**int**.**Parse**(*textBox1*.Text);

**while** (pass != 123456)  
            {  
                *label3*.Visible=**true**;  
                *textBox1*.Text="";  
                pass=**int**.**Parse**(*textBox1*.Text);             
            }  
            *pictureBox1*.Visible=**true**;

*label2*.Visible=**false**;

*label3*.Visible=**false**;

*textBox1*.Visible=**false**;

*button1*.Visible=**false**;  
        }

**Задача 2.** Създайте графично приложение **Coding**, чрез което се въвежда дума и я извежда ***криптирана*** (кодиране на текст, при което всяка буква се замества със следващата в азбуката).

**Примерен вход: роза Примерен изход: спиб**

**кола лпмб**

**Задача 3.** Криптирането през една буква се нарича *със стъпка 1*. При криптиране със стъпка 3 всяка буква се заменя с третата след нея в азбуката - а→г, б→д и т.н. Създайте графично приложение **Coding2**, което по зададена стъпка **n** (0 < n < 31) кодира въведена дума.

**Примерен вход: 2 Примерен изход: грйв**

**боза**

