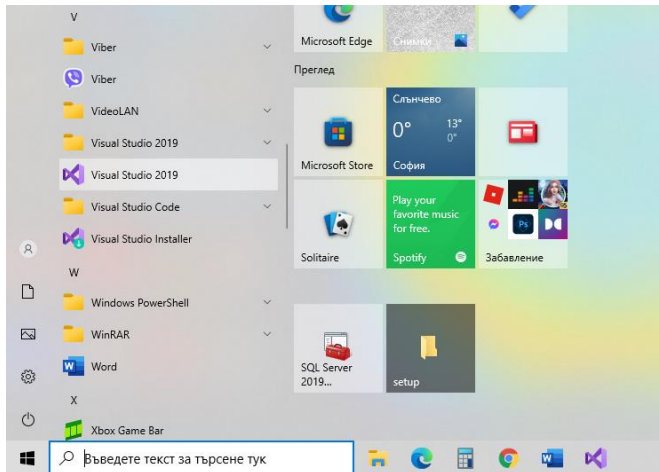


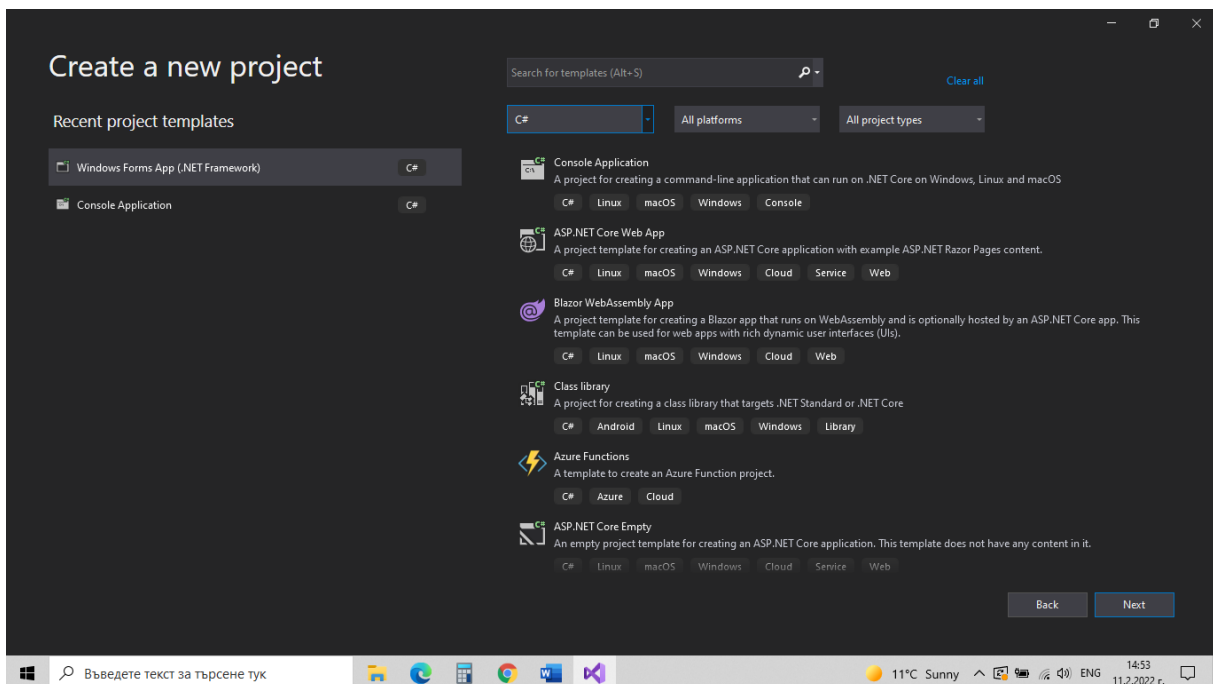
QUIZ GAME USING WINDOWS FORMS APP AND C#

Step 1: creating a new project

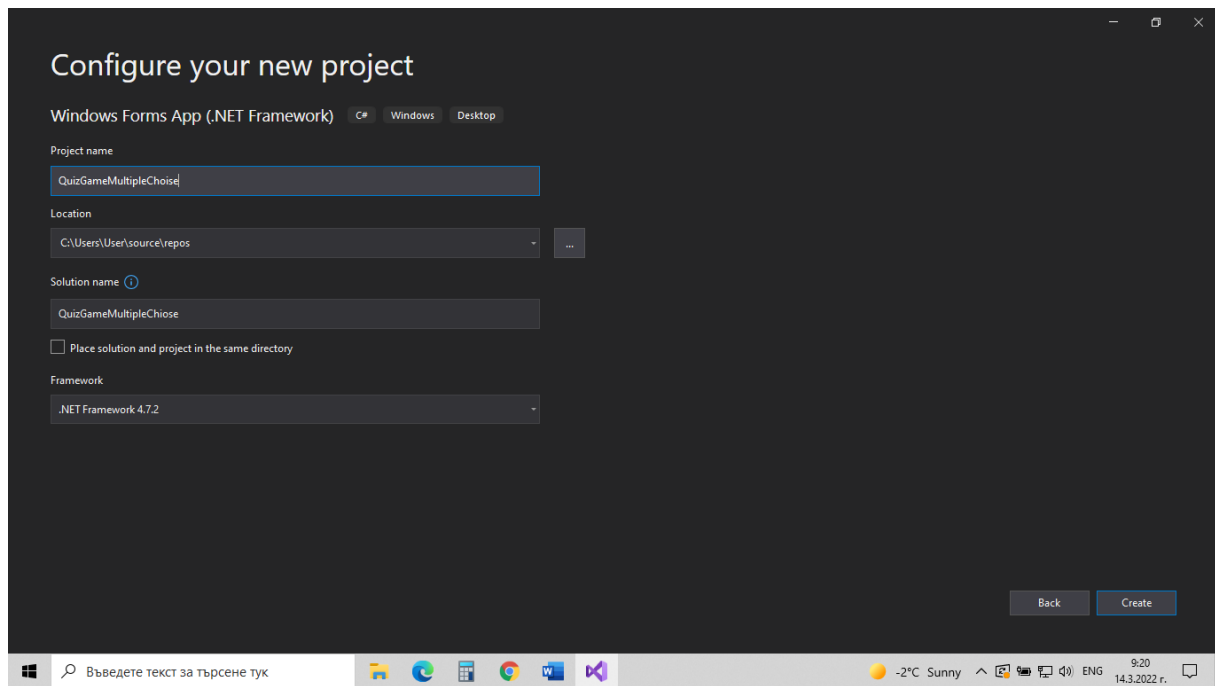
Start the development environment **Visual Studio 2019**.



Choose **Create a new project, Windows Forms App (.NET Framework)** and **C#**

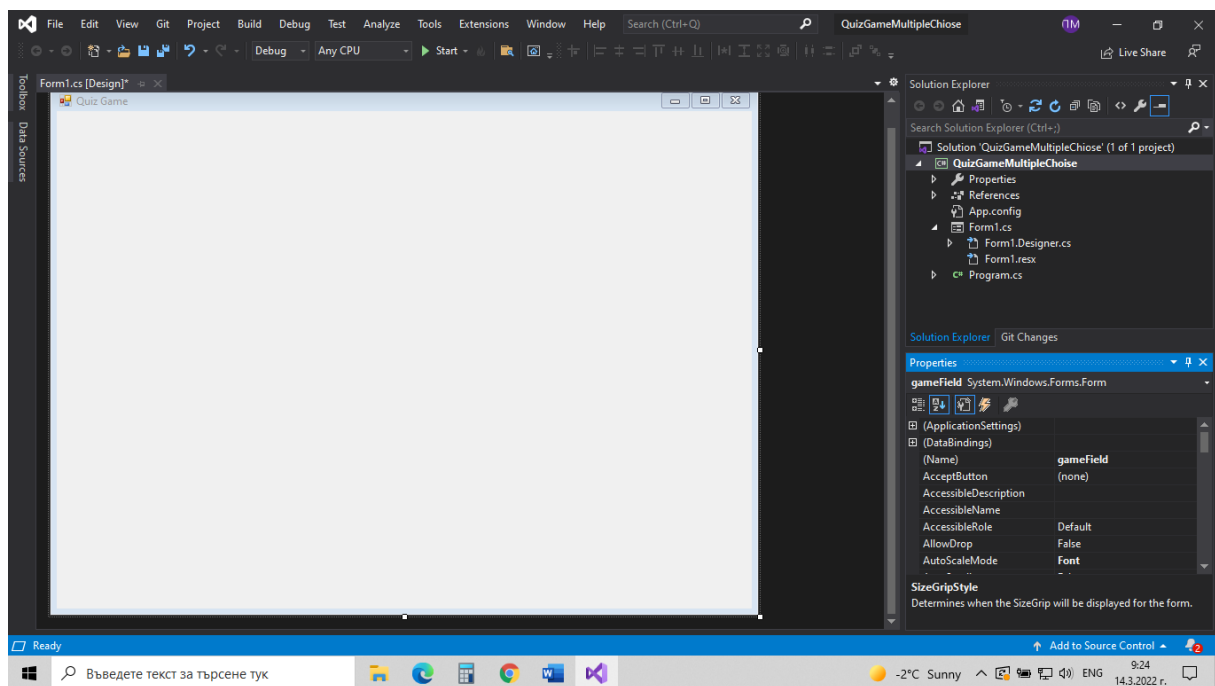


Write **Project name** and **Solution name**. Change the **location** if you want.



Step 2: Setting properties of the Form1:

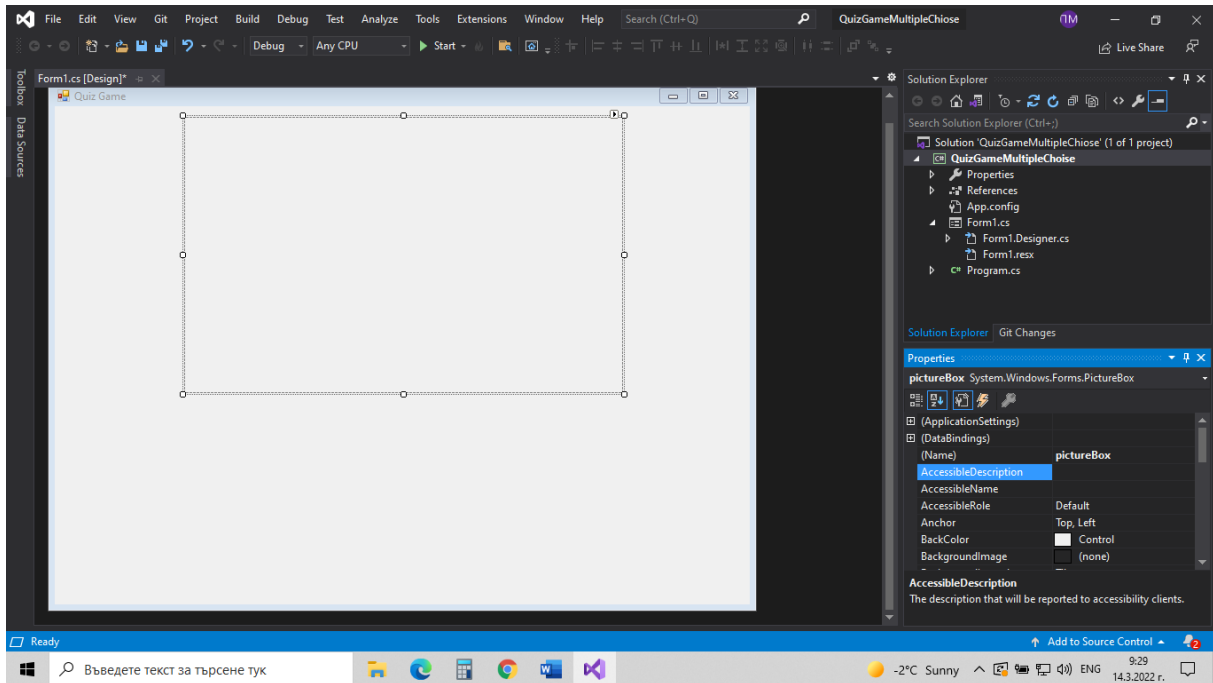
- (Name): gameField
- Size: 800;600
- Text: Quiz Game



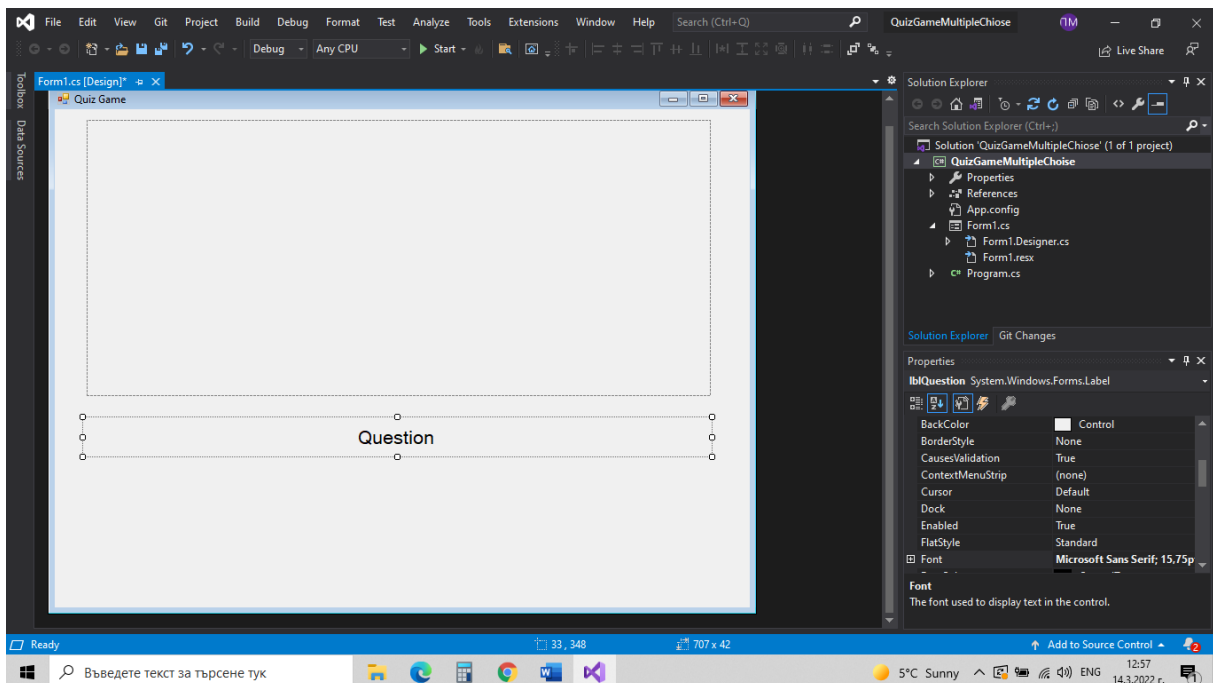
Step3: Adding and setting objects

Setting of pictureBox1:

- (Name): pictureBox



Adding label for the question:

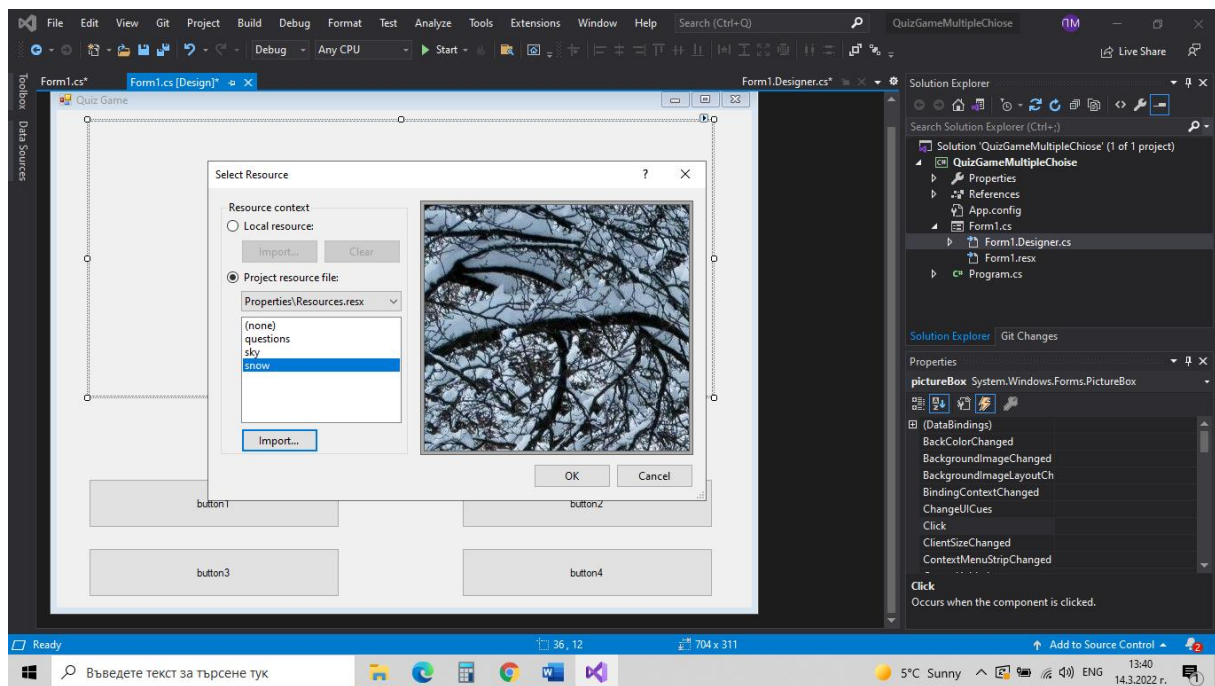


Setting the label:

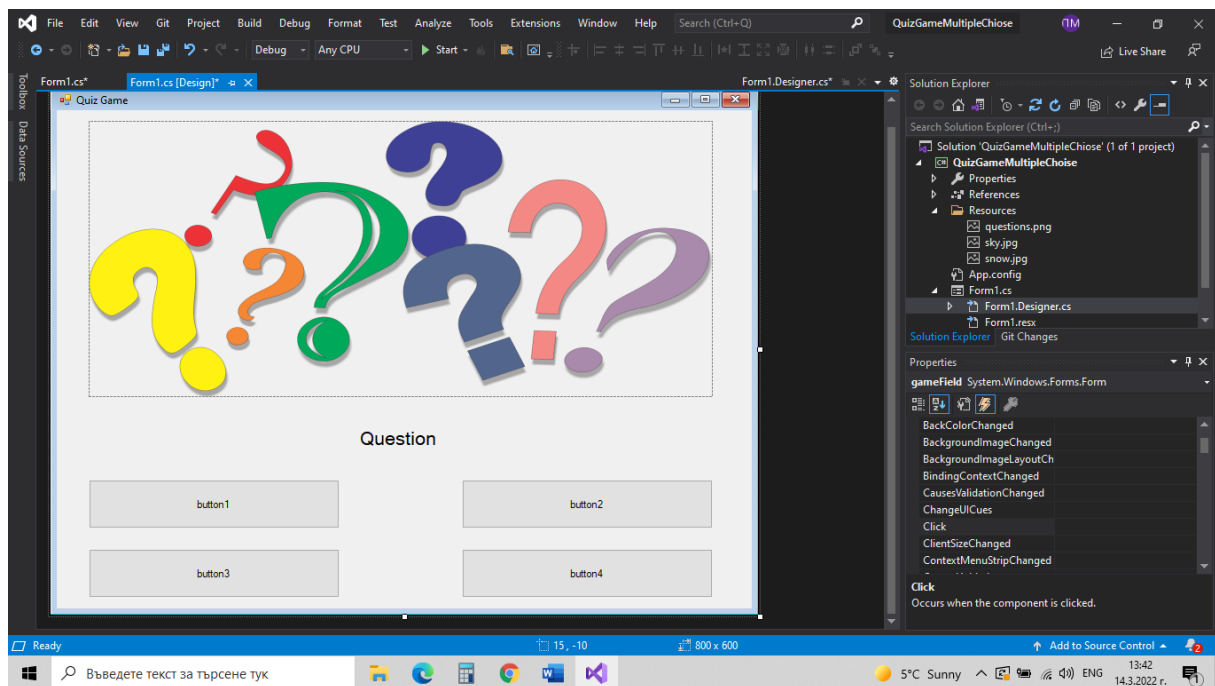
- (Name): lblQuestion
- AutoSize: false

- Font: size 16
- Text: Question
- TextAlign: MiddleCenter

Importing pictures as Project resource file:



Adding 4 buttons:



Setting the buttons:

Every button has the property Tag. We must set it for each of the buttons:

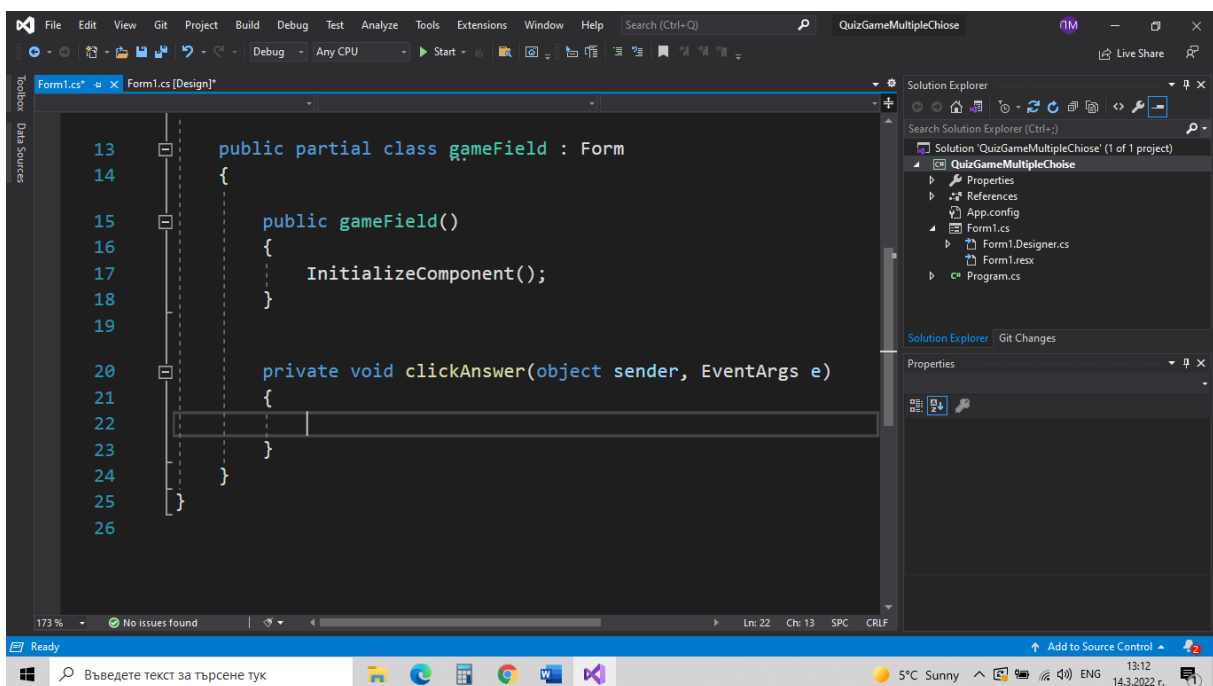
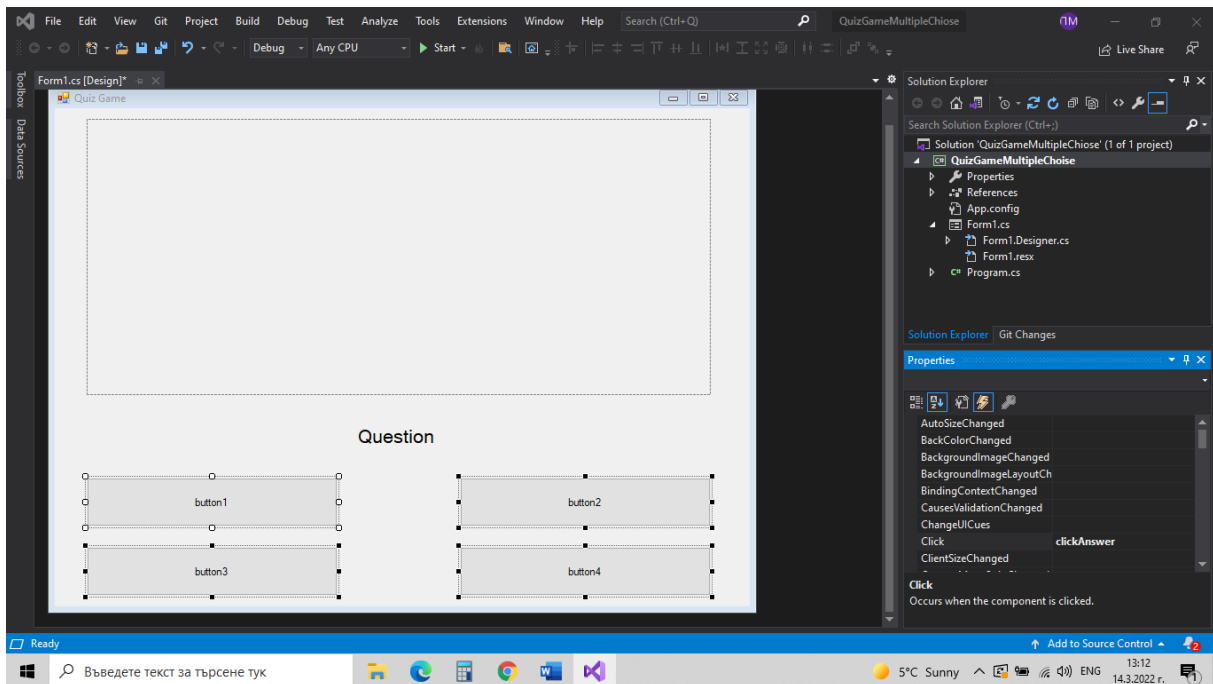
Button1 Tag: 1

Button2 Tag: 2

Button3 Tag: 3

Button4 Tag: 4

For each button we create an event **clickAnswer** that happens at the click of a mouse:



Step 2: Creating variables

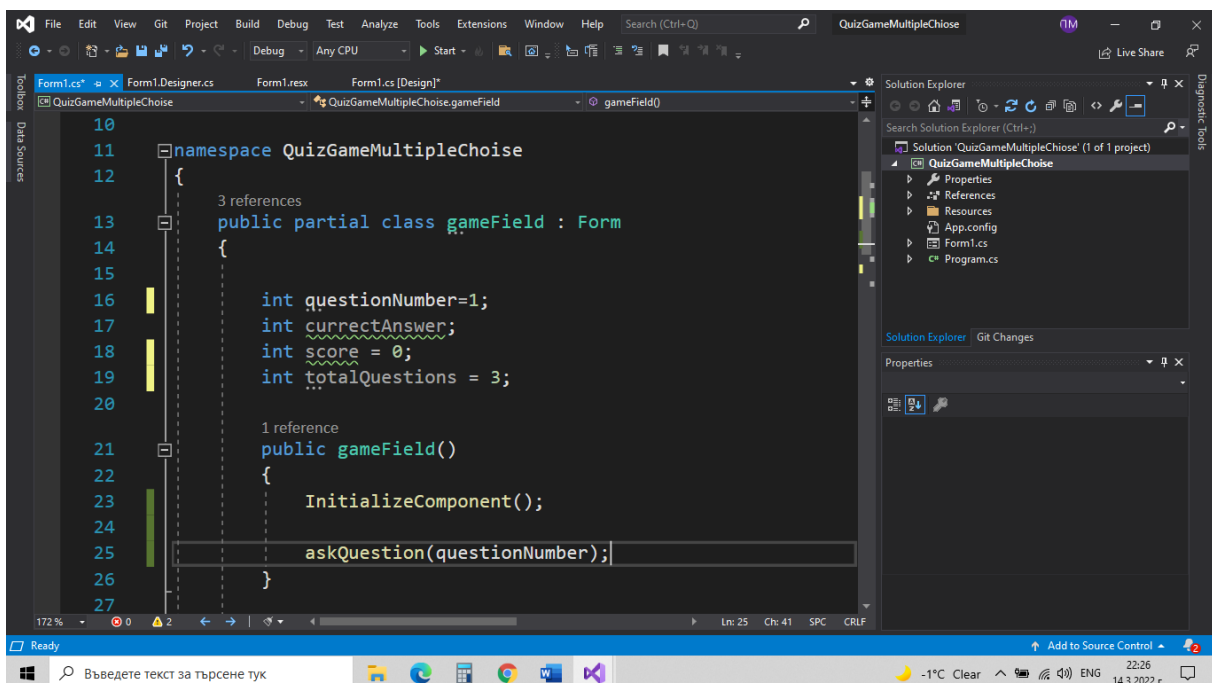
We need 4 variables:

questionNumber – an integer, stores on which question we are at the moment. Question number 1 is loaded at the beginning, so we give a value to this variable 1.

correctAnswer – an integer that keeps the correct answer. At the beginning this variable has no value.

score – an integer that remembers the result. For each correct answer the score is increased by 1.

totalQuestions – the number of all questions.



```
10
11 namespace QuizGameMultipleChoice
12 {
13     3 references
14     public partial class gameField : Form
15     {
16         int questionNumber=1;
17         int correctAnswer;
18         int score = 0;
19         int totalQuestions = 3;
20
21         1 reference
22         public gameField()
23         {
24             InitializeComponent();
25             askQuestion(questionNumber);
26         }
27
```

Step 2: creating a method askQuestion()

This is a void method that does the following:

- sets a picture of the question
- sets the text of the question
- sets the text of the buttons
- sets which is the correct answer

He takes an integer, the question number, as an argument. We will use a switch construction to consider all the options.

```

0 references
private void askQuestion(int qnum)
{
    switch (qnum)
    {
        case 1:
            pictureBox.Image = Properties.Resources.questions;

            lblQuestion.Text = "In which country is the next project mobility?";

            button1.Text = "Serbia";
            button2.Text = "Italy";
            button3.Text = "Turkey";
            button4.Text = "Portugal";

            correctAnswer = 1;
            break;
        case 2:
            pictureBox.Image = Properties.Resources.sky;

            lblQuestion.Text = "What is the color of the sky?";

            button1.Text = "yellow";
            button2.Text = "red";
            button3.Text = "blue";
            button4.Text = "orange";

            correctAnswer = 3;
            break;
        case 3:
            pictureBox.Image = Properties.Resources.snow;

            lblQuestion.Text = "Which season is most likely in the picture?";

            button1.Text = "spring";
            button2.Text = "summer";
            button3.Text = "autumn";
            button4.Text = "winter";

            correctAnswer = 4;
            break;
    }
}

```

Now we have to call this method into the main method – GameField():

```
12 {
13     3 references
14     public partial class gameField : Form
15     {
16         int questionNumber=1;
17         int correctAnswer;
18         int score = 0;
19
20         1 reference
21         public gameField()
22         {
23             InitializeComponent();
24             askQuestion(questionNumber);
25         }
26 }
```

Step 3: creating a method clickAnswer()

What should happen when we click a button?

- we need to be able to determine which button we clicked. Therefore, we will create a variable to associate with the button:

```
4 references
private void clickAnswer(object sender, EventArgs e)
{
    var senderObject = (Button)sender;
}
```

- the created variable is a button object, and it has its own tag property, which is actually the answer number. Let's take the property and store it in an integer variable so that we can do operations with it.

4 references

```
private void clickAnswer(object sender, EventArgs e)
{
    var senderObject = (Button)sender;

    int buttonTag = Convert.ToInt32(senderObject.Tag);
}
```

- We need to check if the chosen answer is the correct answer. If so, increase the score.

4 references

```
private void clickAnswer(object sender, EventArgs e)
{
    var senderObject = (Button)sender;

    int buttonTag = Convert.ToInt32(senderObject.Tag);

    if (buttonTag == correctAnswer)
    {
        score++;
    }
}
```

- We need to move on to the next question:

```
4 references
private void clickAnswer(object sender, EventArgs e)
{
    var senderObject = (Button)sender;

    int buttonTag = Convert.ToInt32(senderObject.Tag);

    if (buttonTag == correctAnswer)
    {
        score++;
    }
    questionNumber++;
    askQuestion(questionNumber);
}
```

- When we reach the end of the test, we need to display a message about the result, reset the score and go back to the first question:

```
4 references
private void clickAnswer(object sender, EventArgs e)
{
    var senderObject = (Button)sender;

    int buttonTag = Convert.ToInt32(senderObject.Tag);

    if (buttonTag == correctAnswer)
    {
        score++;
    }
    questionNumber++;
    askQuestion(questionNumber);
    if (questionNumber == totalQuestions+1)
    {
        MessageBox.Show(
            "The Quiz Ended!" + Environment.NewLine +
            "You have ansered " + score + " answers correctly" + Environment.NewLine +
            "Click OK to play again!"
        );
        score = 0;
        questionNumber = 1;
        askQuestion(questionNumber);
    }
}
```