Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО лабораторной работе №8**

**«Основы событийно-ориентированного программирования»**

**ПО ДИСЦИПЛИНЕ**

**«УП 05.02 Разработка кода информационных систем»**

Выполнила: студентка учебной группы

ИСПк-101-51-00

Дубровина Юлия Константиновна

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

**Цель работы** – получение базовых навыков реализации приложений с графическим интерфейсом пользователя на основе событийно-ориентированной парадигмы.

**Скриншот задания.**

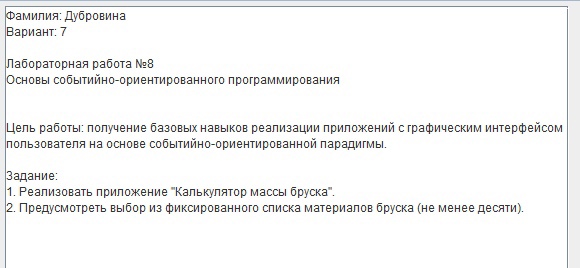
****

Рисунок 1 – Скриншот задания

**Формулировка задания**

1. Реализовать приложение “Калькулятор массы бруска”.
2. Предусмотреть выбор из фиксированного списка материалов бруска (не менее десяти).

**Описание кода.**

Программа реализована на языке программирования Python с помощью библиотеки PyQt5 и ее дополнительных параметров. Введен список материалов бруском в количестве 10 штук с помощью переменной. Так же, реализована функция для расчета массы бруска, проверка на наличие данных в полях ввода и вывод сообщения об ошибке. Создание стиля для окна создавалось с помощью CSS.

**Код программы**

import sys  
  
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QLineEdit, QComboBox, QPushButton, QVBoxLayout, QHBoxLayout, QSizePolicy, QMessageBox  
  
  
  
# Список материалов брусков  
  
materials = {  
  
 'Сосна': 450,  
  
 'Ель': 520,  
  
 'Дуб': 830,  
  
 'Береза': 670,  
  
 'Лиственница': 550,  
  
 'Кедр': 470,  
  
 'Осина': 330,  
  
 'Ясень': 670,  
  
 'Клен': 550,  
  
 'Груша': 670,  
  
 'Яблоня': 670,  
  
 'Вишня': 550,  
  
 'Красное дерево': 710,  
  
}  
  
  
  
# Функция для расчета массы бруска  
  
def calc\_mass():  
  
 length = length\_input.text()  
  
 width = width\_input.text()  
  
 height = height\_input.text()  
  
  
  
 # Проверка на наличие данных в полях ввода  
  
 if not length or not width or not height:  
  
 # Вывод сообщения об ошибке  
  
 QMessageBox.warning(window, 'Ошибка', 'Введите значения длины, ширины и высоты бруска.')  
  
 return  
  
  
  
 length = float(length)  
  
 width = float(width)  
  
 height = float(height)  
  
 material = material\_select.currentText()  
  
 volume = length \* width \* height  
  
 density = materials[material]  
  
 mass = volume \* density / 1000000  
  
 mass\_label.setText(f'Масса бруска из {material}, размером {length}x{width}x{height} см, составляет {mass:.2f} кг.')  
  
  
  
# Создание графического интерфейса  
  
app = QApplication(sys.argv)  
  
window = QWidget()  
  
window.setWindowTitle('Калькулятор массы бруска')  
  
  
  
# Создание стилевой таблицы для установки размеров текста, фона и цвета текста  
  
style\_sheet = """  
  
QLabel {  
  
 font-size: 24px;  
  
 color: black;  
  
 margin-top: 10px;  
  
 margin-bottom: 10px;  
  
  
  
}  
  
QLineEdit {  
  
 background-color: rgba(255, 255, 255, 150);  
  
 font-size: 24px;  
  
 color: black;  
  
 margin-top: 10px;  
  
 margin-bottom: 10px;  
  
}  
  
QComboBox {  
  
 font-size: 24px;  
  
 color: black;  
  
}  
  
QPushButton {  
  
 font-size: 24px;  
  
 color: black;  
  
 background-color: rgba(255, 0, 255, 150);  
  
}  
  
QWidget {  
  
 background-image: url(backgroun1.jpg);  
  
 background-position: center;  
  
 background-repeat: no-repeat;  
  
 background-size: cover;  
  
}  
  
"""  
  
  
  
# Установка стилевой таблицы для окна  
  
window.setStyleSheet(style\_sheet)  
  
  
  
# Создание макета для полей ввода  
  
input\_layout = QVBoxLayout()  
  
input\_layout.addWidget(QLabel('Длина (см):'))  
  
length\_input = QLineEdit()  
  
length\_input.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Fixed)  
  
input\_layout.addWidget(length\_input)  
  
input\_layout.addWidget(QLabel('Ширина (см):'))  
  
width\_input = QLineEdit()  
  
width\_input.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Fixed)  
  
input\_layout.addWidget(width\_input)  
  
input\_layout.addWidget(QLabel('Высота (см):'))  
  
height\_input = QLineEdit()  
  
height\_input.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Fixed)  
  
input\_layout.addWidget(height\_input)  
  
  
  
# Создание макета для выбора материала  
  
material\_layout = QHBoxLayout()  
  
material\_layout.addWidget(QLabel('Материал:'))  
  
material\_select = QComboBox()  
  
material\_select.addItems(materials.keys())  
  
material\_select.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Fixed)  
  
material\_layout.addWidget(material\_select)  
  
  
  
# Создание макета для кнопки расчета и метки с результатом  
  
button\_layout = QVBoxLayout()  
  
calc\_button = QPushButton('Рассчитать')  
  
calc\_button.clicked.connect(calc\_mass)  
  
calc\_button.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Fixed)  
  
button\_layout.addWidget(calc\_button)  
  
mass\_label = QLabel()  
  
mass\_label.setSizePolicy(QSizePolicy.Expanding, QSizePolicy.Expanding)  
  
mass\_label.setStyleSheet('margin-top: 10px; margin-bottom: 10px;')  
  
button\_layout.addWidget(mass\_label)  
  
  
  
# Создание главного макета и добавление в него других макетов  
  
main\_layout = QVBoxLayout()  
  
main\_layout.addLayout(input\_layout)  
  
main\_layout.addLayout(material\_layout)  
  
main\_layout.addLayout(button\_layout)  
  
  
  
# Установка главного макета для окна  
  
window.setLayout(main\_layout)  
  
  
  
# Назначение минимального размера окна  
  
window.setMinimumSize(300, 200)  
  
  
  
window.showMaximized()  
  
  
  
sys.exit(app.exec\_())

**Результат выполнения программы**

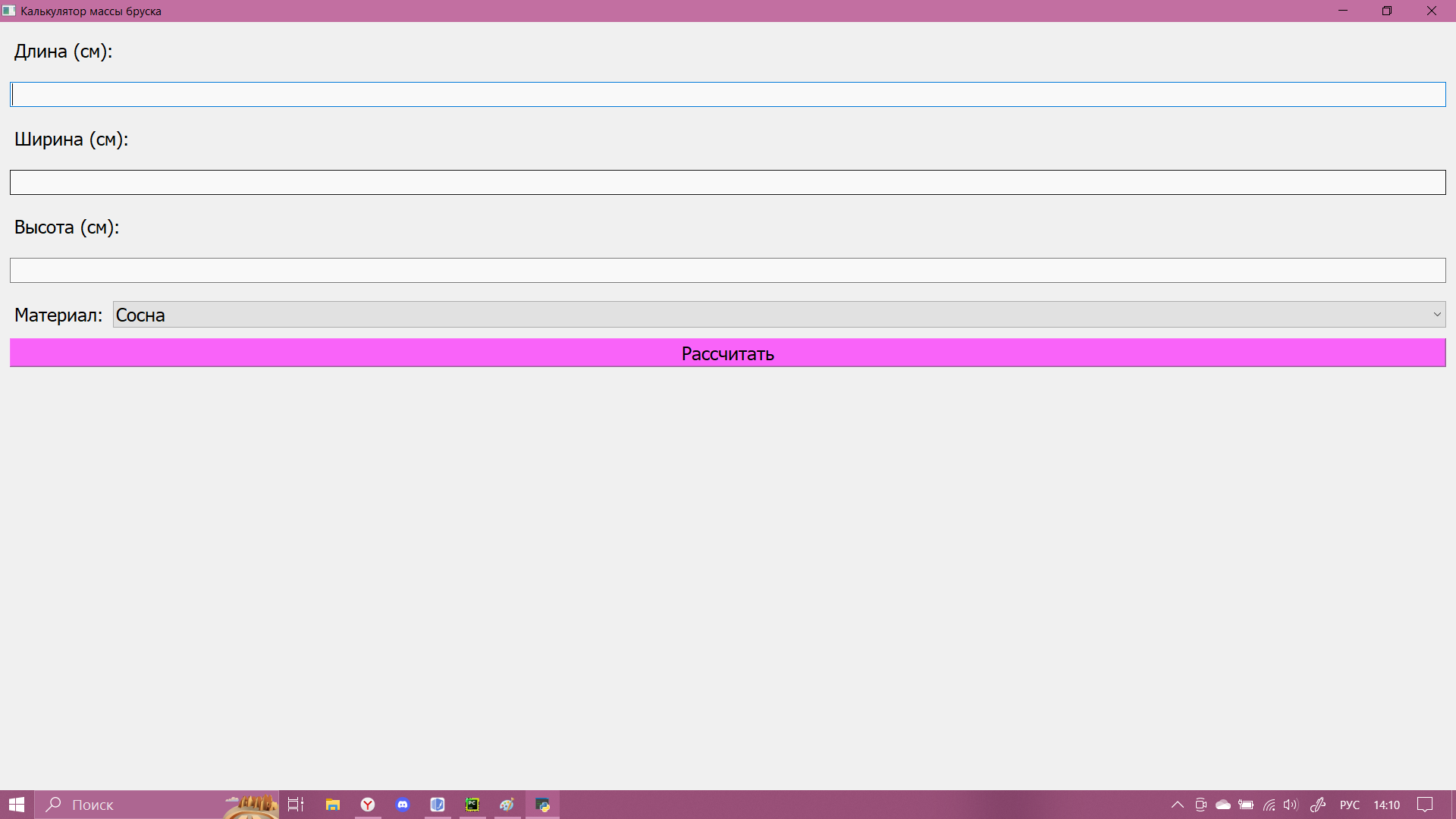
****

Рисунок 2 – Результат запуска программы

(См. рис. 3) Так же, можно вводить длину, ширину и высоту материала бруска, сего моей программе их 10 видов.

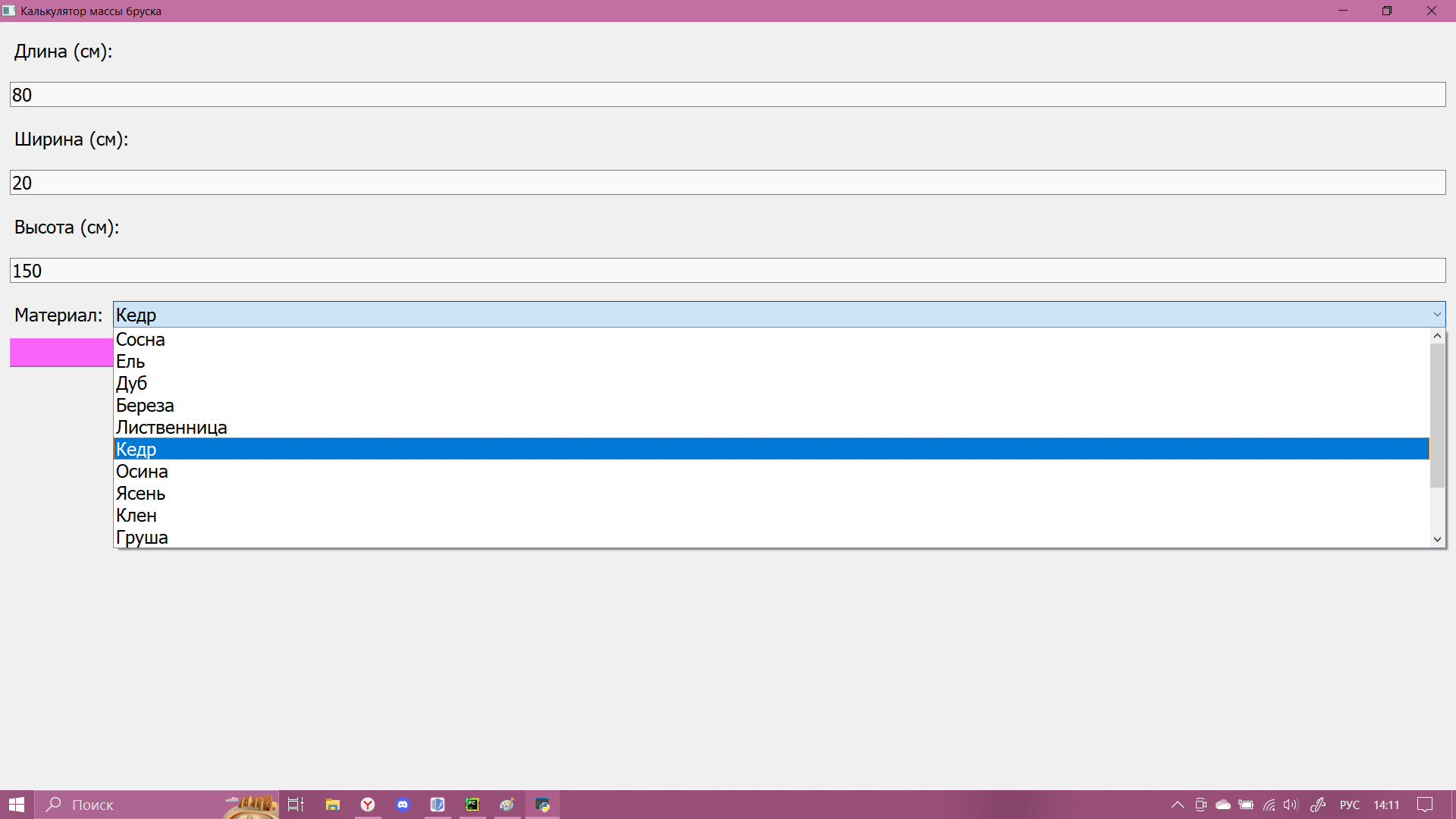


Рисунок 3 – ввод данных и выбор материала бруска

(См. рис. 4) При нажатии на кнопку “расчитать” и выбрав любой материал с вводом любых значений данных, снизу будет выводиться вес бруска.

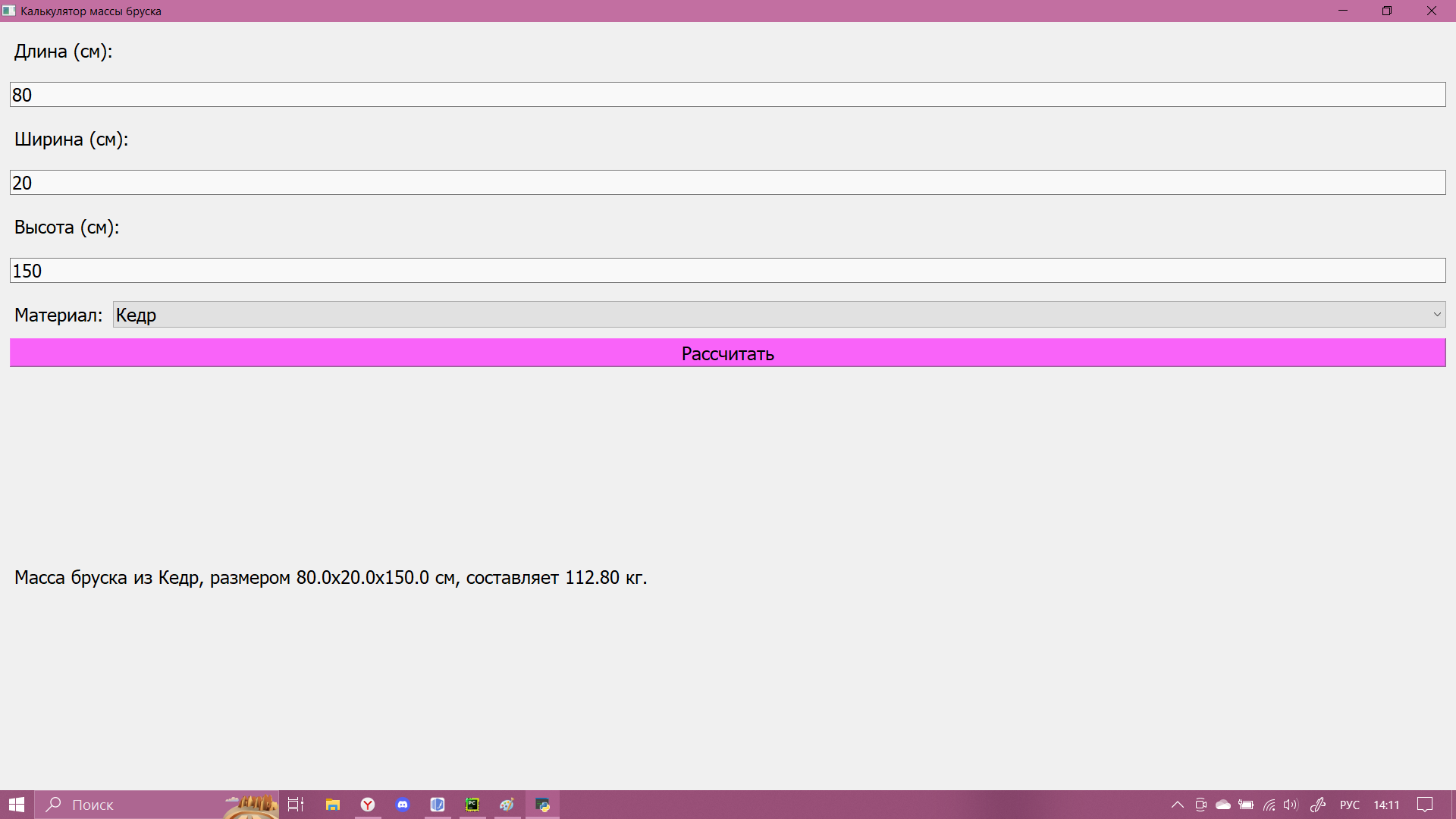


Рисунок 4 – Масса бруска.

**Вывод**

Входе работы над программой, я освоила навыки создания графических приложений с помощью библиотеки PyQt5, так же изучила формулы расчета массы разных материалов брусков. Вывела результат выполнения программы с помощью скриншотов и объяснила все назначения кнопок и поля ввода данных. Работа была выполнена мною полном объеме.