**МИНИCTEPCTBO НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение высшего образования**

**«Северо-Кавказский федеральный университет» Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №2**

**по дисциплине «Проектирование человеко-машинного интерфейса»**

|  |
| --- |
| Выполнил студент группы ПИЖ-б-о-20-1 |
| Ваньянц И.М. « » 20 г. |
| Подпись студента |
| Работа защищена « » \_20 г. |
| Проверил Мельников С.В.  (подпись) |

Ставрополь 2022

**ХОД РАБОТЫ**

Задание 1. Макет

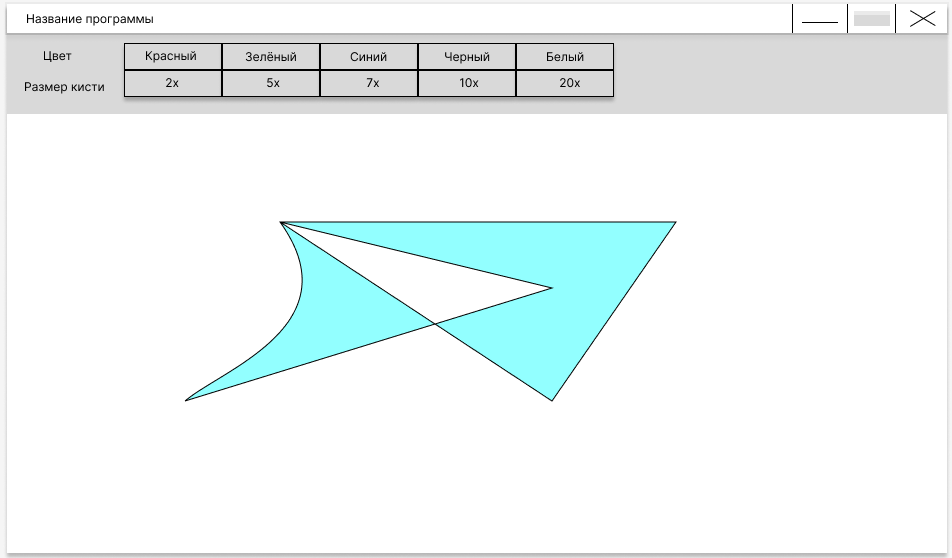


Рисунок 1 – Макет пробной версии

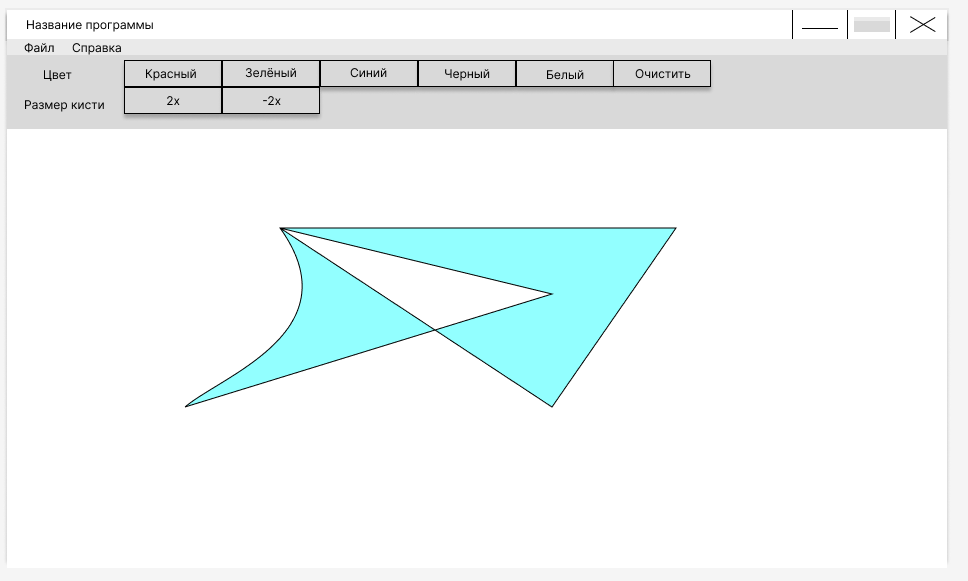


Рисунок 2 – Макет финальной версии

Задание 2. Программа

Код пробной версии программы:

# -\*- coding: utf-8 -\*-  
from tkinter import \*  
from tkinter.messagebox import \*  
  
  
class Paint(Frame):  
 def \_\_init\_\_(self, parent):  
 Frame.\_\_init\_\_(self, parent)  
 self.parent = parent  
 # Параметры кисти по умолчанию  
 self.brush\_size = 10  
 self.brush\_color = "red"  
 self.color = "red"  
 # Компоненты UI  
 self.setUI()  
  
 # Метод рисования на холсте  
  
 def draw(self, event):  
 self.canv.create\_oval(event.x - self.brush\_size,  
 event.y - self.brush\_size,  
 event.x + self.brush\_size,  
 event.y + self.brush\_size,  
 fill=self.color, outline=self.color)  
  
 # Изменение цвета кисти  
  
 def set\_color(self, new\_color):  
 self.color = new\_color  
  
 # Изменение размера кисти  
  
 def set\_brush\_size(self, new\_size):  
 self.brush\_size = new\_size  
  
 def setUI(self):  
 # Название окна  
 self.parent.title("Painter")  
 # Активные элементы на родительском окне  
 self.pack(fill=BOTH, expand=1)  
  
 self.columnconfigure(6, weight=1)  
 self.rowconfigure(2, weight=1)  
  
 # Холст с белым фоном  
 self.canv = Canvas(self, bg="white")  
  
   
 # Канвас  
  
 self.canv.grid(row=2, column=0, columnspan=7, padx=5, pady=5, sticky=E + W + S + N)  
  
 # Реакция холста на нажатие левой кнопки мыши  
 self.canv.bind("<B1-Motion>", self.draw)  
  
 # Метка для кнопок изменения цвета кисти  
 color\_lab = Label(self, text="Цвет: ")  
  
 # Метка в первый ряд и первую колонку,  
 # горизонтальный отступ в 6 пикселей  
 color\_lab.grid(row=0, column=0, padx=6)  
  
 # Кнопки размеров   
 red\_btn = Button(self, text="Красный", width=10, command=lambda: self.set\_color("red"))  
 red\_btn.grid(row=0, column=1)

green\_btn = Button(self, text="Зеленый", width=10, command=lambda: self.set\_color("green"))  
 green\_btn.grid(row=0, column=2)  
  
 blue\_btn = Button(self, text="Синий", width=10, command=lambda: self.set\_color("blue"))  
 blue\_btn.grid(row=0, column=3)  
  
 black\_btn = Button(self, text="черный", width=10, command=lambda: self.set\_color("black"))  
 black\_btn.grid(row=0, column=4)  
  
 white\_btn = Button(self, text="Белый", width=10, command=lambda: self.set\_color("white"))  
 white\_btn.grid(row=0, column=5)  
  
 # Кнопки изменения размера кисти  
 size\_lab = Label(self, text="Размер кисти: ")  
 size\_lab.grid(row=1, column=0, padx=5)  
 one\_btn = Button(self, text="2x", width=10, command=lambda: self.set\_brush\_size(2))  
 one\_btn.grid(row=1, column=1)  
  
 two\_btn = Button(self, text="5x", width=10, command=lambda: self.set\_brush\_size(5))  
 two\_btn.grid(row=1, column=2)  
  
 five\_btn = Button(self, text="7x", width=10, command=lambda: self.set\_brush\_size(7))  
 five\_btn.grid(row=1, column=3)  
  
 seven\_btn = Button(self, text="10x", width=10, command=lambda: self.set\_brush\_size(10))  
 seven\_btn.grid(row=1, column=4)  
  
 ten\_btn = Button(self, text="20x", width=10, command=lambda: self.set\_brush\_size(20))  
 ten\_btn.grid(row=1, column=5)  
  
  
  
  
  
# функция для создания главного окна  
def main():  
 global root  
 root = Tk()  
 root.geometry("800x600+300+300")  
 app = Paint(root)  
 m = Menu(root)  
 root.config(menu=m)  
  
   
 root.mainloop()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

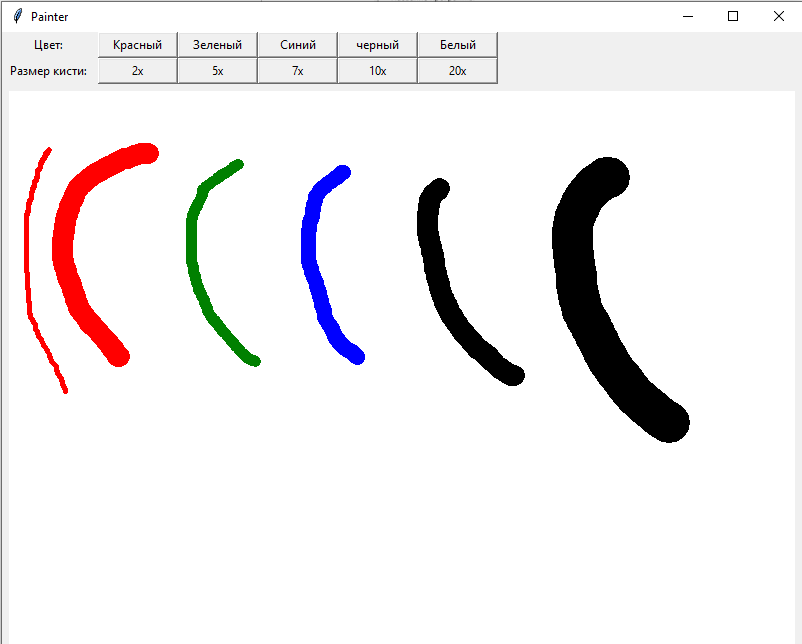


Рисунок 3 – Работа пробной версии

Код финальной версии программы:  
# -\*- coding: utf-8 -\*-  
from tkinter import \*  
from tkinter.messagebox import \*  
  
  
  
class Paint(Frame):  
 def \_\_init\_\_(self, parent):  
 Frame.\_\_init\_\_(self, parent)  
 self.parent = parent  
 # Параметры кисти по умолчанию  
 self.brush\_size = 10  
 self.brush\_color = "red"  
 self.color = "red"  
 # Компоненты UI  
 self.setUI()  
  
 # Метод рисования на холсте  
  
 def draw(self, event):  
 self.canv.create\_oval(event.x - self.brush\_size,  
 event.y - self.brush\_size,  
 event.x + self.brush\_size,  
 event.y + self.brush\_size,  
 fill=self.color, outline=self.color)  
  
 # Изменение цвета кисти  
  
 def set\_color(self, new\_color):  
 self.color = new\_color  
  
  
 # Изменение размера кисти  
  
 def set\_brush\_size\_plus(self, brush\_size):  
 self.brush\_size = self.brush\_size\*2  
  
 def set\_brush\_size\_minus(self, brush\_size):  
 self.brush\_size = self.brush\_size/2  
  
 def setUI(self):  
 # Название окна  
 self.parent.title("Painter")  
 # Активные элементы на родительском окне  
 self.pack(fill=BOTH, expand=1)  
  
 self.columnconfigure(6, weight=1)  
 self.rowconfigure(2, weight=1)  
  
 # Холст с белым фоном  
 self.canv = Canvas(self, bg="white")  
  
   
 self.canv.grid(row=2, column=0, columnspan=7, padx=5, pady=5, sticky=E + W + S + N)  
  
 # Реакция холста на нажатие левой кнопки мыши  
 self.canv.bind("<B1-Motion>", self.draw)  
  
 # Кнопки изменения цвета кисти  
 color\_lab = Label(self, text="Цвет: ")  
  
 # Первый ряд и первая колонка,  
 # горизонтальный отступ в 6 пикселей  
 color\_lab.grid(row=0, column=0, padx=6)  
  
 # Кнопки цвета

red\_btn = Button(self, text="Красный", width=10, command=lambda: self.set\_color("red"))  
 red\_btn.grid(row=0, column=1)  
  
 green\_btn = Button(self, text="Зеленый", width=10, command=lambda: self.set\_color("green"))  
 green\_btn.grid(row=0, column=2)  
  
 blue\_btn = Button(self, text="Синий", width=10, command=lambda: self.set\_color("blue"))  
 blue\_btn.grid(row=0, column=3)  
  
 black\_btn = Button(self, text="черный", width=10, command=lambda: self.set\_color("black"))  
 black\_btn.grid(row=0, column=4)  
  
 white\_btn = Button(self, text="Белый", width=10, command=lambda: self.set\_color("white"))  
 white\_btn.grid(row=0, column=5)  
  
 # Кнопки размера кисти  
 size\_lab = Label(self, text="Размер кисти: ")  
 size\_lab.grid(row=1, column=0, padx=5)  
  
 one\_btn = Button(self, text="2x", width=10, command=lambda: self.set\_brush\_size\_plus(self.brush\_size))  
 one\_btn.grid(row=1, column=1)  
  
 two\_btn = Button(self, text="-2x", width=10, command=lambda: self.set\_brush\_size\_minus(self.brush\_size))  
 two\_btn.grid(row=1, column=2)  
  
 clear\_btn = Button(self, text="Очистить", width=10, command=lambda: self.canv.delete("all"))  
 clear\_btn.grid(row=0, column=6, sticky=W)  
  
  
# выход из программы  
def close\_win():  
 if askyesno("Выход", "Вы уверены?"):  
 root.destroy()  
  
  
# вывод справки  
def about():  
 showinfo("Demo Paint", "Выполнил Ваньянц Илья Михайлович, ПИЖ-Б-О-20-1")  
  
  
# Главное окно  
def main():  
 global root  
 root = Tk()  
 root.geometry("800x600+300+300")  
 app = Paint(root)  
 m = Menu(root)  
 root.config(menu=m)  
  
 fm = Menu(m)  
 m.add\_cascade(label="Файл", menu=fm)  
 fm.add\_command(label="Выход", command=close\_win)  
  
 hm = Menu(m)  
 m.add\_cascade(label="Справка", menu=hm)  
 hm.add\_command(label="О программе", command=about)  
 root.mainloop()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

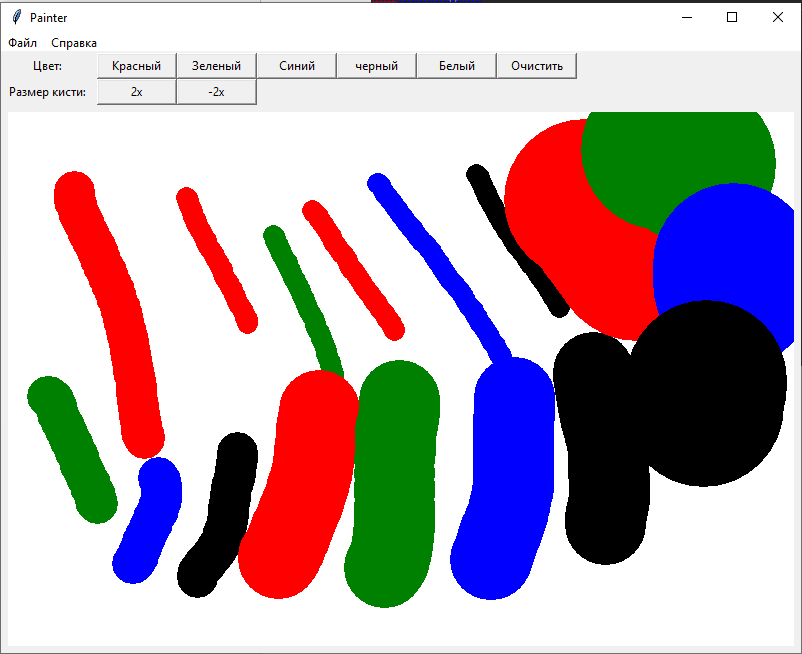


Рисунок 4 – Работа программы