Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Основы программной инженерии (ОПИ)

ОТЧЕТ

по лабораторной работе № 1

Вариант 6

Выполнил:

Студент гр. 351051 Носович А.А.

Проверил: Фадеева Е.Е.

Минск 2023

ИНДИВИДУАЛЬНАЯ РАБОТА №1

**ТЕМА: ОРГАНИЗАЦИЯ ТЕХНОЛОГИЧЕСКОГО ПРОЦЕССА РАЗРАБОТКИ**

**ПРОГРАММНЫХ ПРОДУКТОВ.**

**1. Контрольные вопросы:**

1. **Приложение** –программная реализация на компьютере решения задачи.

**Программный продукт** –комплекс взаимосвязанных программ

для решения определенной задачи массового спроса, подготовленный к реализации как любой вид промышленной продукции.

**Утилита** – небольшая вспомогательная программа для решения специализированных задач по настройке, оптимизации, улучшению работы оборудования и программного обеспечения.

1. **Жизненный цикл программного обеспечения** – период времени, который начинается с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

Состоит из следующих основных этапов:

- Анализ и планирование

- Проектирование

- Разработка /Реализация

- Тестирование

- Внедрение / Ввод в эксплуатацию

- Эксплуатация / Сопровождение

1. **Интерфейс** – граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия между элементами системы.

**Пользовательский интерфейс** – интерфейс, обеспечивающий передачу информации между пользователем-человеком и программно-аппаратными компонентами компьютерной системы.

**Графический интерфейс** – система средств для взаимодействия пользователя с электронными устройствами, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана.

1. Основные цели в составлении «**Технического задания»** – назначение и область применения программы, технические, технико-экономические и специальные требования, предъявляемые к программе, необходимые стадии и сроки разработки, виды испытаний.
2. **Программное обеспечение контроля версий** отслеживает все вносимые в код изменения в специальной базе данных. При обнаружении ошибки разработчики могут вернуться назад и выполнить сравнение с более ранними версиями кода для исправления ошибок, сводя к минимуму проблемы для всех участников команды.
3. **Git, SVN, darcs, Bazaar, mercurial, perforce.**
4. Для регистрации, учёта и контроля программных дефектов и иных ошибок на проекте. Также **системы управления дефектами** упрощают отслеживание, классификацию и приоритезацию багов.
5. **Jira, YouTrack, Trello, Bugzilla.**
6. **IDE (integrated development environment)** – комплекс программных средств, используемый программистами для разработки программного обеспечения.
7. Для написания приложения к индивидуальной практической работе я использовал **Visual Studio (а именно Visual Studio Code).** Одним из главных преимуществ Visual Studio является его широкая поддержка языков программирования. Она позволяет разрабатывать приложения на таких языках, как C++, C#, Visual Basic, F#, Python и многих других.

**2. Разработать программное средство с графическим интерфейсом.**

1. Номер моего варианта – 6. Тема приложения - Проверка умения складывать и вычитать числа. Для реализации использовал язык программирования Python.

Программный код имеет следующий вид:

import tkinter as tk

import random

def generate\_question():

num1 = random.randint(1, 10)

num2 = random.randint(1, 10)

operator = random.choice(['+', '-'])

question = f"Решите пример: {num1} {operator} {num2}" + " = "

if operator == '+':

result = num1 + num2

else:

result = num1 - num2

return question, result

def check\_answer():

user\_answer = entry.get()

if user\_answer == str(result):

label.config(text="Отлично! Ответ правильный!")

else:

label.config(text="Неправильно! Пропробуйте ещё раз!")

def next\_question():

global result

question, result = generate\_question()

question\_label.config(text=question)

entry.delete(0, 'end')

label.config(text="")

root = tk.Tk()

root.title("Проверка умения складывать и вычитать")

root.geometry("500x300")

root["bg"] = "#a7c3cf"

question\_label = tk.Label(root, text="", font=(

"Arial", 20), background="#a7c3cf")

question\_label.pack()

entry = tk.Entry(root, font=("Arial", 24))

entry.pack()

check\_button = tk.Button(root, text="Проверить ответ", font=(

"Arial", 20, 'bold'), bg="#a7c3cf", command=check\_answer)

check\_button.pack(anchor="center", padx=20, pady=30)

label = tk.Label(root, text="", font=("Arial", 16), background="#a7c3cf")

label.pack(padx=20, pady=0)

next\_button = tk.Button(root, text="Следующий пример",

font=("Arial", 18), bg="#679fb8", command=next\_question)

next\_button.pack(anchor="s", padx=10, pady=20)

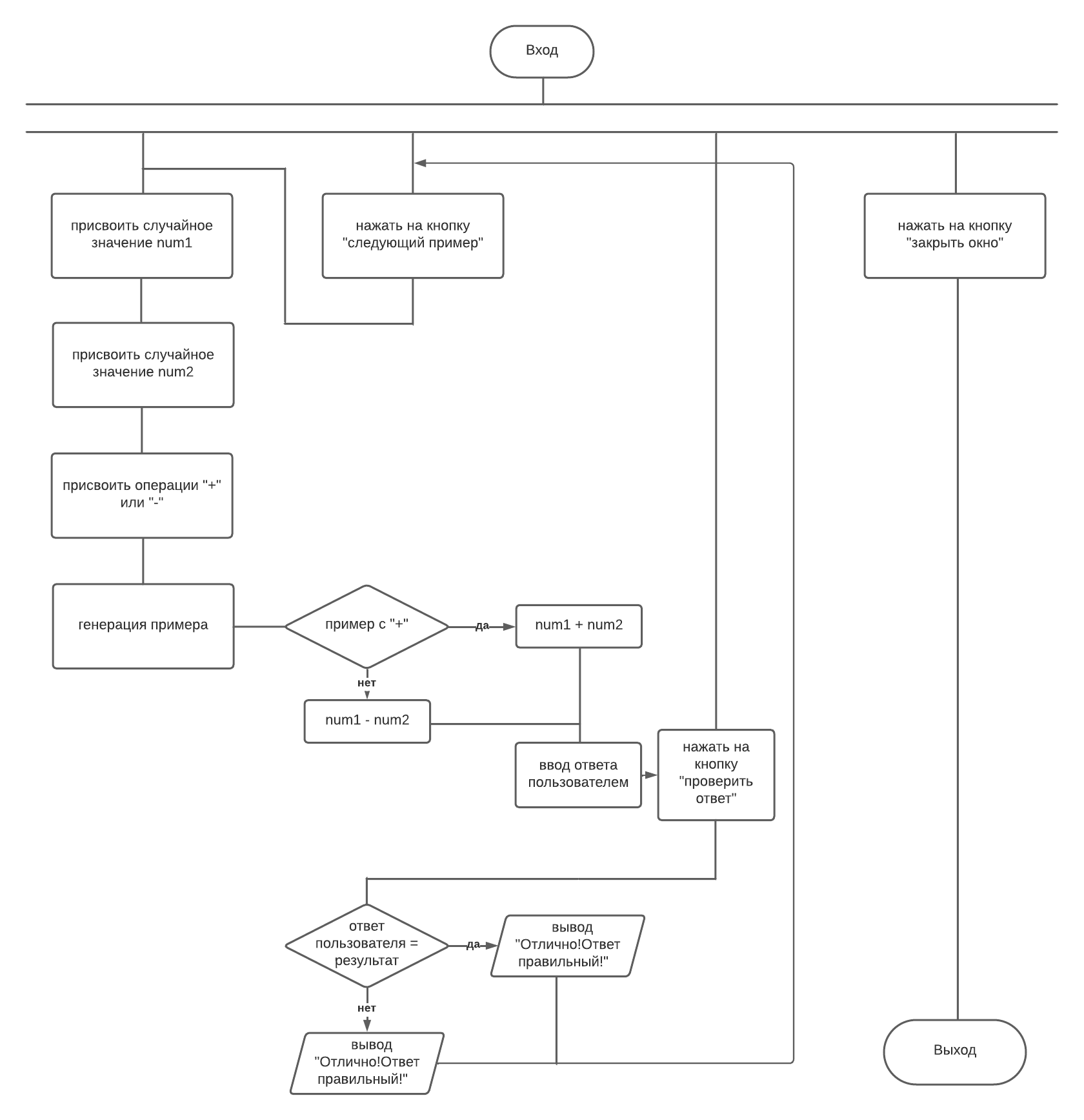
next\_question()

root.mainloop()

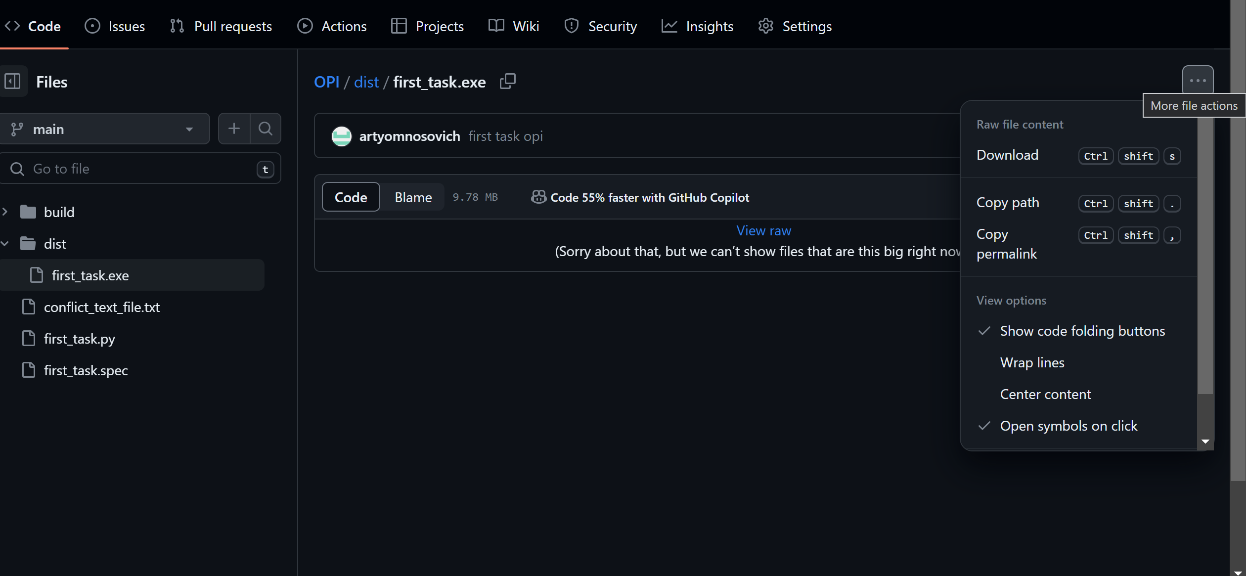
1. Описание применения.

Программный продукт предназначен для вывода двух случайных чисел, а также случайного знака между ними в качестве операции сложения или вычитания, и подаётся на вывод в виде примера. Пользователю предлагается ввести ответ на отображаемый пример и проверить свои знания в области сложения и вычитания двух чисел. Запускать на персональных компьютерах под управлением ОС Windows 11 64-битная система.

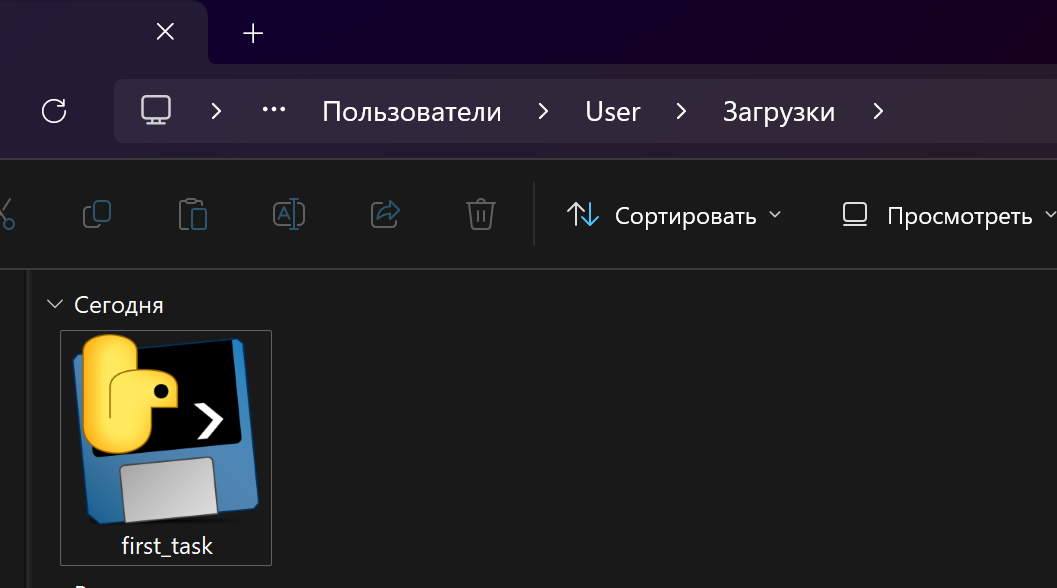
1. Детализированная схема алгоритма работы программного средства в соответствии с ГОСТ 19.701–90:



1. Для установки программы необходимо перейти по ссылке <https://github.com/artyomnosovich/OPI/blob/main/dist/first_task.exe> и скачать файл first\_task.exe нажатием кнопки **«…»** -> **Download:**



Для запуска программы необходимо сделать двойное нажатие левой кнопкой мыши на само приложение first\_task.exe находящийся в локальном хранилище на ПК:



После чего может появится уведомление защитника Windows, в котором нужно нажать «Выполнить в любом случае» (стандартная реакция на скрипты/ приложения написанные на языке Python) для запуска программы.

Руководство по эксплуатации: Пользователю подаётся 2 случайных числа в диапазоне от 1 до 10, а также случайный знак между ними в виде примера, ответ на который программа предлагает дать пользователю в поле для ввода с помощью клавиатуры. После ввода своего ответа пользователю необходимо нажать на кнопку "Проверить ответ" для сравнения действительного решения примера с данными, введенными пользователем. В случае правильного ответа пользователю на экран выводится сообщение "Отлично! Ответ правильный!", в случае неправильного ответа - "Неправильно! Попробуйте ещё раз!" и предоставляется возможность дать ответ на тот же пример вплоть до получения положительного результата или приступить к решению следующего примера нажатием кнопки "Следующий пример". Кнопка "Следующий пример" доступна для нажатия и до первого ввода ответа и также отправляет пользователю новый пример для предоставления решения. В случае оставления поля ввода пустым и нажатия на кнопку "Проверить ответ" на экран выводится сообщение "Неправильно! Попробуйте ещё раз!" Выход из программы происходит с помощью нажатия на кнопку в виде креста в правом верхнем углу окна.

Графический интерфейс приложения выглядит следующим образом:

