

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра информационных систем

ПРАКТИЧЕСКАЯ РАБОТА №7
по дисциплине «Объектно-ориентированное программирование»
ТЕМА: ВЫЧИСЛЕНИЕ ФУНКЦИЙ

Студенты гр. 3372

Преподаватель

Беляев К.В.
Лазарев Ф.Н.
Егоров С.С.

Санкт-Петербург

2024

Задание на Практическую работу

Студенты Беляев К.В., Лазарев Ф.Н.

Группа 3372

Исходные данные: создать распределенное приложение, включающее клиентскую и серверную части, взаимодействующие посредством сетевого обмена сообщениями.

Клиентская часть представляет собой GUI приложение, реализующее интерфейс аналогичный работе №4.

Параметризовать уже существующие классы используя шаблоны, позволяющие работать с разными типами данных

Серверная часть представляет собой консольное приложение, предназначенное для выполнения перечисленных в меню работы №3 функций над полиномом с комплексными элементами, а также вычисление значения функций \sin и \cos .

Спецификации классов

Таблица 1. Протокол класса ClientApplication

Атрибуты(старые)		
идентификатор	область видимости	семантическое описание
*comm	private	Объект класса Communicator
*interface	private	Вывод меню в консоль
Методы(старые)		
идентификатор	область видимости	семантическое описание
ClientApplication	public	Конструктор класса
fromCommunicator	public slot	Принятие ответа из Серверной части
toCommunicator	public slot	Отправление запроса в Серверную часть

Таблица 2. Протокол класса ServerApplication

Атрибуты(старые)		
идентификатор	область видимости	семантическое описание
*comm	private	Объект класса Communicator
*polynom	private	Объект класса Polynom с комплексным коэф.
*floatPolynom	private	Объект класса Polynom с вещественными коэф.
*roots	private	Массив комплексных корней полинома
*floatRoots	private	Массив вещественных корней полинома
An	private	Комплексный старший коэффициент
floatAn	private	Вещественный старший коэффициент
rootsAmount	private	Кол-во корней
Методы(старые)		
идентификатор	область видимости	семантическое описание
pushBack	public	Добавление нового комплексного корня
pushBack	public	Добавление нового вещественного корня
recieve	public slot	Работа с запросами из Клиентской части

Таблица 3. Протокол класса TCommunicator

Атрибуты(старые)		
идентификатор	область видимости	семантическое описание
ready	private	Готовность принять запрос
params	private	Параметры коммуникатора
Методы(старые)		
идентификатор	область видимости	семантическое описание
TCommunicator	public	Объект класса
recieved	public signal	Работа с запросами
send	public slot	Получение запроса из клиентской части
recieve	private slot	Получение ответа из серверной части

Таблица 4. Протокол класса Array

Атрибуты(старые)			
идентификатор	тип	область видимости	семантическое описание
length	int	private	Целочисленная длина массива
arr	number*	private	Указатель на первый элемент массива
Методы(старые)			
идентификатор	область видимости	семантическое описание	
Array	public	Конструктор класса. Создает массив заданной длины, по умолчанию – 0	
~Array	public	Деструктор класса	
getLength	public	Получение длины массива	
fill	public	Заполнение массива числами с консоли	
resize	public	Изменение размера массива	
changeElement	public	Изменение выбранного элемента числом с консоли	
printArray	public	Вывод массива в консоль	

averageValue	public	Подсчет среднего значения элементов массива
SKO	public	Подсчет СКО элементов массива
shakerSort	public	Сортировка массива по убыванию – если передается параметр 1, по возрастанию – если передается 0 или не передается ничего

Таблица 5. Протокол класса TComplex

Атрибуты(старые)			
идентификатор	тип	область видимости	семантическое описание
re	double	private	Вещественная часть комплексного числа
im	double	private	Мнимая часть комплексного числа
separator	Qchar	private	Разделение аргументов в запросе
Методы(старые)			
идентификатор	область видимости		семантическое описание
TComplex()	public		Конструктор класса по умолчанию
TComplex(double re, double im)	public		Конструктора класса, принимающий вещественное и мнимое части комплексного числа
TComplex(double re)	public		Конструктор класса, принимающий вещественную часть комплексного числа
getRe	public		Получение вещественной части комплексного числа
getIm	public		Получение мнимой части комплексного числа
module	public		Вычисление модуля комплексного числа
operator+	public		Оператор сложения
operator-	public		Оператор вычитания
operator/	public		Оператор деления
operator*	public		Оператор умножения
operator+=			Оператор сложения с присваиванием
operator-=	public		Оператор вычитания с присваиванием

operator/=	public	Оператор деления с присваиванием
operator*= operator*=	public	Оператор умножения с присваиванием
operator=	public	Оператор присваивания
operator==	public	Оператор «равно»
operator!=	public	Оператор «неравно»
operator<	public	Оператор «меньше»
operator>	public	Оператор «больше»
pow	public	Вычисление корня из комплексного числа
setSeparator	public	Установление разделителя аргументов
Operator QString()	public	Преобразование в QString
operator >>	public	Оператор “>>” для работы с потоком ввода
operator <<	public	Оператор “<<” для работы с потоком вывода
operator >>	public	Оператор “>>” для работы с QByteArray

Таблица 6. Протокол класса Polynom

Атрибуты(старые)			
идентификатор	тип	область видимости	семантическое описание
roots	number*	private	Массив корней полинома
coefficients	number*	protected	Массив коэффициентов полинами
degree	int	protected	Степень полинома
Методы(старые)			
идентификатор	область видимости		семантическое описание
Polynom()	public		Конструктор класса по умолчанию
polynomWithDegrees	public		Вывод полинома в каноническом виде
polynomWithRoots	public		Вывод полинома с корнями
valueAtPoint	public		высчитать значение в точке

Таблица 7. Протокол класса Tinterface

Атрибуты(старые)			
идентификатор	тип	область видимост и	семантическое описание
coeffsLabel	QLabel *	private	Лэйбл коэффициента
imIndicator	QLabel *	private	Лэйбл индикатора i
reCoeffsLE	QLineEdit *	private	Поле ввода действительной части
imCoeffsLE	QLineEdit *	private	Поле ввода мнимой части
addRootBTN	QPushButton *	private	Кнопка добавления корня
changeRootBTN	QPushButton *	private	Кнопка изменения корня
changeRootLineEdit	QLineEdit *	private	Поле ввода индекса корня
leadingCoeff	QLabel *	private	Лэйбл старшего коэффициента
reLeadCoeff	QLineEdit *	private	Поле ввода действительной части старшего коэффициента
imLeadCoeff	QLineEdit *	private	Поле ввода мнимой части старшего коэффициента
addLeadCoeffBTN	QPushButton *	private	Кнопка добавления старшего коэффициента
calculateValueAtPoint	QLabel *	private	Лэйбл значения полинома в точке
valueAtPointLE	QLineEdit *	private	Поле ввода точки
calculateValueAtPointBTN	QPushButton *	private	Кнопка вычисления значения полинома в точке
valueAtPointLabel	QLabel *	private	Лэйбл значения полинома в точке

printWithRootsBTN	QPushButton *	private	Кнопка вывода полинома с корнями
printCanonBtn	QPushButton *	private	Кнопка вывода полинома в каноническом виде
outputLabel	QLabel *	private	Лэйбл с полиномом
polynom	Polynom *	private	Указатель на объект полинома
roots	number *	private	Массив с корнями полинома
rootsAmount	int	private	Количество корней полинома
An	number	private	Старший коэффициент полинома
RealMode	*QRadioButton	private	кнопка переключения режима
ComplexMode	*QRadioButton	private	кнопка переключения режима
Атрибуты(новые)			
trigonometry	*QLabel	private	Описание функционала
SinMode	*QRadioButton	private	Переключение на функцию Sin
SiMode	*QRadioButton	private	Переключение на функцию Si
decompositionDegreeLabel	*QLabel	private	label перед вводом
printFuncDecomposition	*QPushButton	private	вывод значения
funcDegree	*QLineEdit	private	ввод степени разложения
decompositionLabel	*QLabel	private	Label перед выводом
valueToDecomposeLabel	*QLabel	private	label перед вводим
ReValueToDecompose	*QLineEdit	private	вещественная часть X
ImValueToDecompose	*QLineEdit	private	мнимая часть X
Методы(старые)			
идентификатор	область видимости		семантическое описание
Tinterface	public		Конструктор класса

~Tinterface	public	Деструктор класса
answer	public slot	Вывод ответа из серверной части
formRequest	private slots	Формирование запроса
request	public signal	Отправление запроса в серверную часть

Таблица 8. Первичный протокол класса TFunction

Методы		
идентификатор	область видимости	семантическое описание
TFunction()	public	Конструктор класса по умолчанию
value()	public	Вычисление значения функции в точке

Таблица 9. Первичный протокол класса TFSin

Атрибуты			
идентификатор	тип	область видимости	семантическое описание
derivatives	number*	private	Массив производных
Методы			
идентификатор	область видимости		семантическое описание
TFSin()	public		Конструктор класса по умолчанию
derivativesCalc()	public		Вычисление производных
~TFSin()	public		Деструктор класса по умолчанию

Таблица 10. Первичный протокол класса TFSin

Атрибуты			
идентификатор	тип	область видимости	семантическое описание
derivatives	number*	private	Массив производных

Методы		
идентификатор	область видимости	семантическое описание
TFSi()	public	Конструктор класса по умолчанию
derivativesCalc()	public	Вычисление производных
~TFSi()	public	Деструктор класса по умолчанию

Диаграмма классов

На рисунке 1 представлена диаграмма классов, дополненная атрибутами и методами.

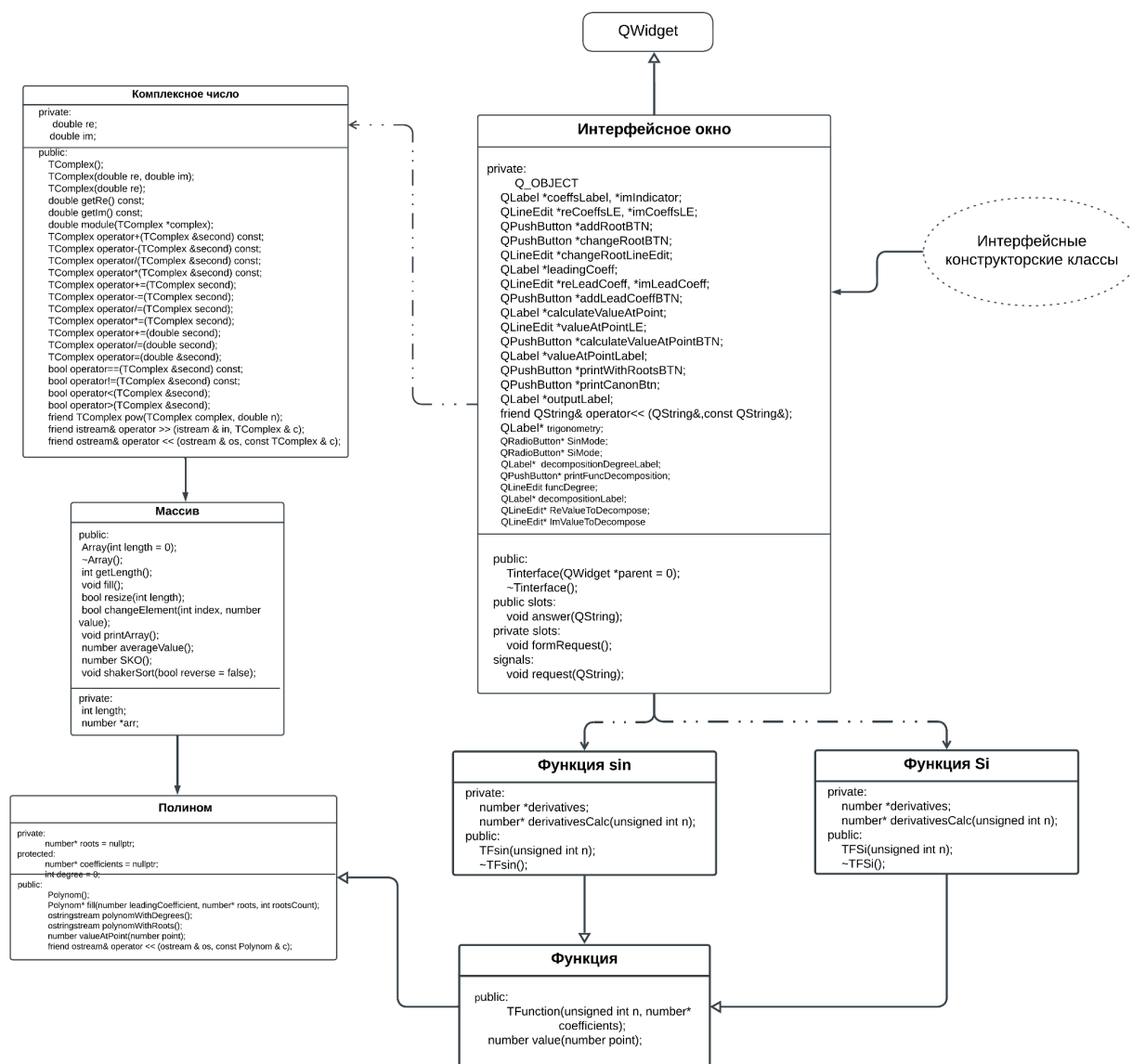


Рисунок 1 – Диаграмма классов.

Описание контрольного примера с исходными и ожидаемыми расчетными данными

1. Выбираем функцию для разложения: Sin и множество разложения: Комплексные числа
2. Вводим степень разложения функции: 5 и точку $X_0 : 5 + 6i$
3. Нажимаем кнопку показать разложение, и выводится вычисление функции в заданной точке: $74.1667 - 33i$
4. Изменяем степень разложения на 10 и выводим значение:
 $-361.643 + 88.749i$

*Для режим вещественных чисел - функционал аналогичен

* Для функции Si - функционал аналогичен

СКРИНШОТЫ РАБОТЫ ПРОГРАММЫ НА КОНТРОЛЬНЫХ ПРИМЕРАХ

После запуска программы на экране появляется окно, с кнопками, полями ввода и полями с текстом на рисунке 2.

The screenshot shows a dark-themed application window with the following elements:

- Кoefficient:** A label followed by a text input field containing '1' and a '+i' label, and another empty text input field.
- Buttons:** 'Добавить корень' (Add root) and 'Изменить корень с индексом' (Change root with index).
- an =**: A label followed by two empty text input fields and a '+i' label.
- Button:** 'Добавить an' (Add an).
- Вычислить значение в точке x =**: A label followed by two empty text input fields and a '+i' label.
- Button:** 'Вычислить' (Calculate).
- Buttons:** 'Показать с корнями' (Show with roots) and 'Показать в каноническом виде' (Show in canonical form).
- Radio buttons:** 'Вещественные числа' (Real numbers) - selected, and 'Комплексные числа' (Complex numbers).
- Text:** 'Вывод:' (Output:).
- Section:** 'Функции Sin и Si' (Functions Sin and Si).
- Radio buttons:** 'Функция Sin' (Sin function) - selected, and 'Функция Si' (Si function).
- Text:** 'Степень разложения:' (Degree of expansion:).
- Text:** 'X:' followed by two empty text input fields and a '+i' label.
- Button:** 'Показать разложение' (Show expansion).
- Text:** 'Вывод:' (Output:).

Рисунок 2 – Запуск программы

Для вычисления значения функции необходимо ввести степень разложения и точку X_0 .

Функции Sin и Si

☒ Функция Sin ☐ Функция Si

Степень разложения: 5 X: 5 + i 6|

Показать разложение

Вывод:

Рисунок 3 – Ввод степени и точки X_0 из контрольного примера

Чтобы вывести значение функции, нажмем кнопку «Показать разложение». В окне появится значение функции с учетом введенных на предыдущем шаге степенью и точкой X_0 (рисунок 4).

Функции Sin и Si

☒ Функция Sin ☐ Функция Si

Степень разложения: 5 X: 5 + i 6|

Показать разложение

Вывод: 74.1667 -33i

Рисунок 4 – Вывод разложения на экран

Изменим степень разложения функции. Введем 10 в поле ввода степени и нажмем кнопку “Показать разложение”. На рисунке 5 показаны результаты работы программы.

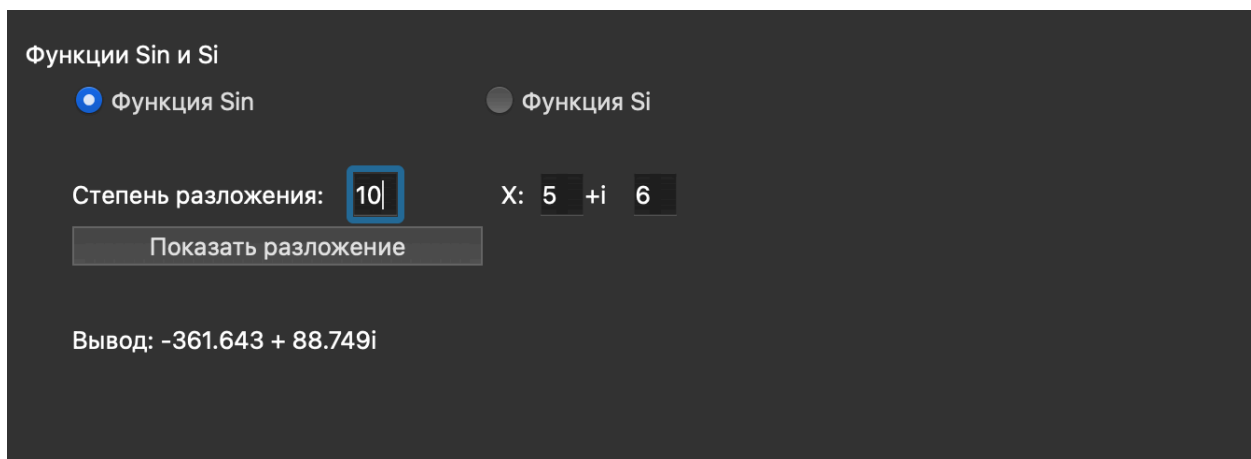


Рисунок 5 – Изменение Степени разложения

Наконец, чтобы выйти из программы, нужно нажать на крестик в правом верхнем углу, и программа автоматически закроется.

ВЫВОДЫ ПО ВЫПОЛНЕНИЮ РАБОТЫ

В рамках данной практической работы была реализована и отлажена программа, предназначенная для создания GUI приложения с клиентской и серверной частью, реализующего функции перечисленные в описании работы №6, но был добавлен функционал вычисления функции в точке X_0 путем разложения в степенной ряд и его вычисления. Также был разработан контрольный пример для проведения проверки, с чем программа справилась успешно.