

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

ПРАКТИЧЕСКАЯ РАБОТА №6
по дисциплине «Объектно-ориентированное программирование»
ТЕМА: ПАРАМЕТРИЗАЦИЯ КЛАССОВ

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.

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. Первичный 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*=	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 видимост и	семантическое описание
roots	number*	private	Массив корней полинома
coefficients	number*	private	Массив коэффициентов полинами
degree	int	private	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)

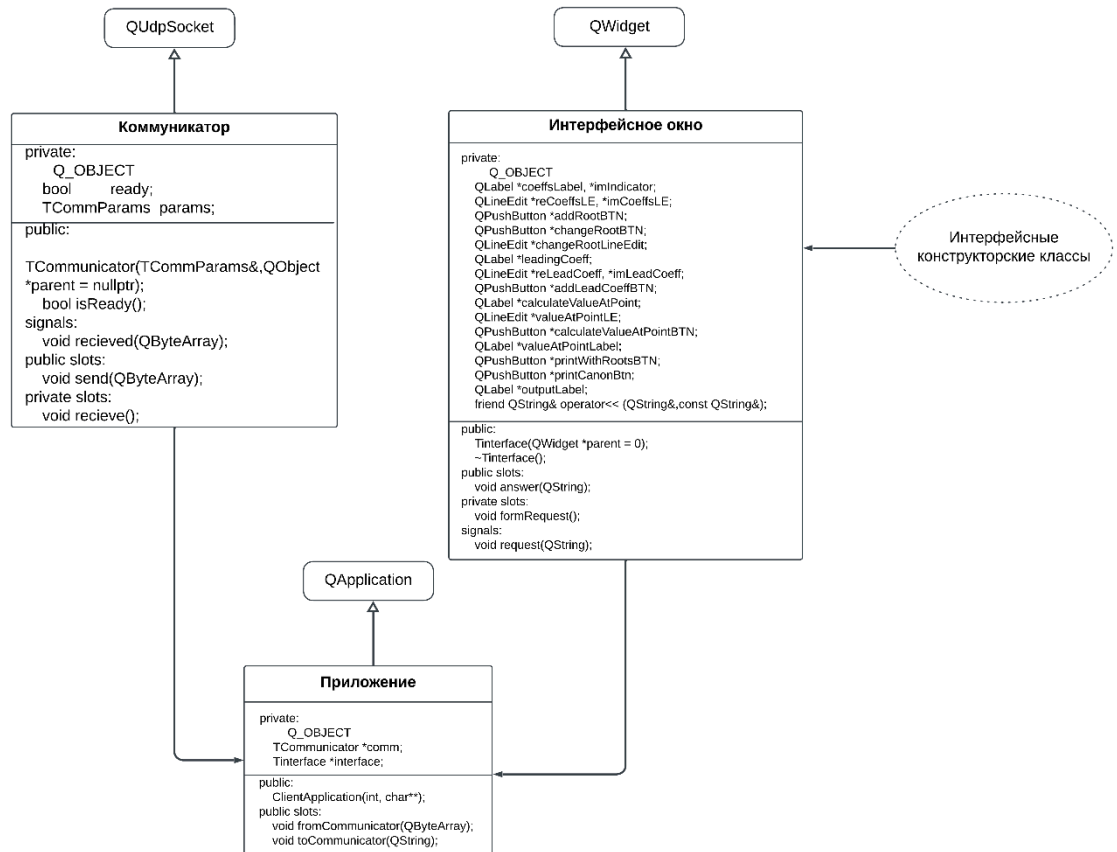
идентификатор	тип	область видимост и	семантическое описание
coeffsLabel	QLabel *	private	Лэйбл коэффициента
imIndicator	QLabel *	private	Лэйбл индикатора i
reCoeffsLE	QLineEdit *	private	Поле ввода действительной части
imCoeffsLE	QLineEdit *	private	Поле ввода мнимой части
addRootBTN	QPushButton *	private	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
Атрибуты(новые)			
RealMode	*QRadioButton	private	кнопка переключения режима
ComplexMode	*QRadioButton	private	кнопка переключения режима
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	

Диаграмма классов

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

Клиентская часть



Серверная часть

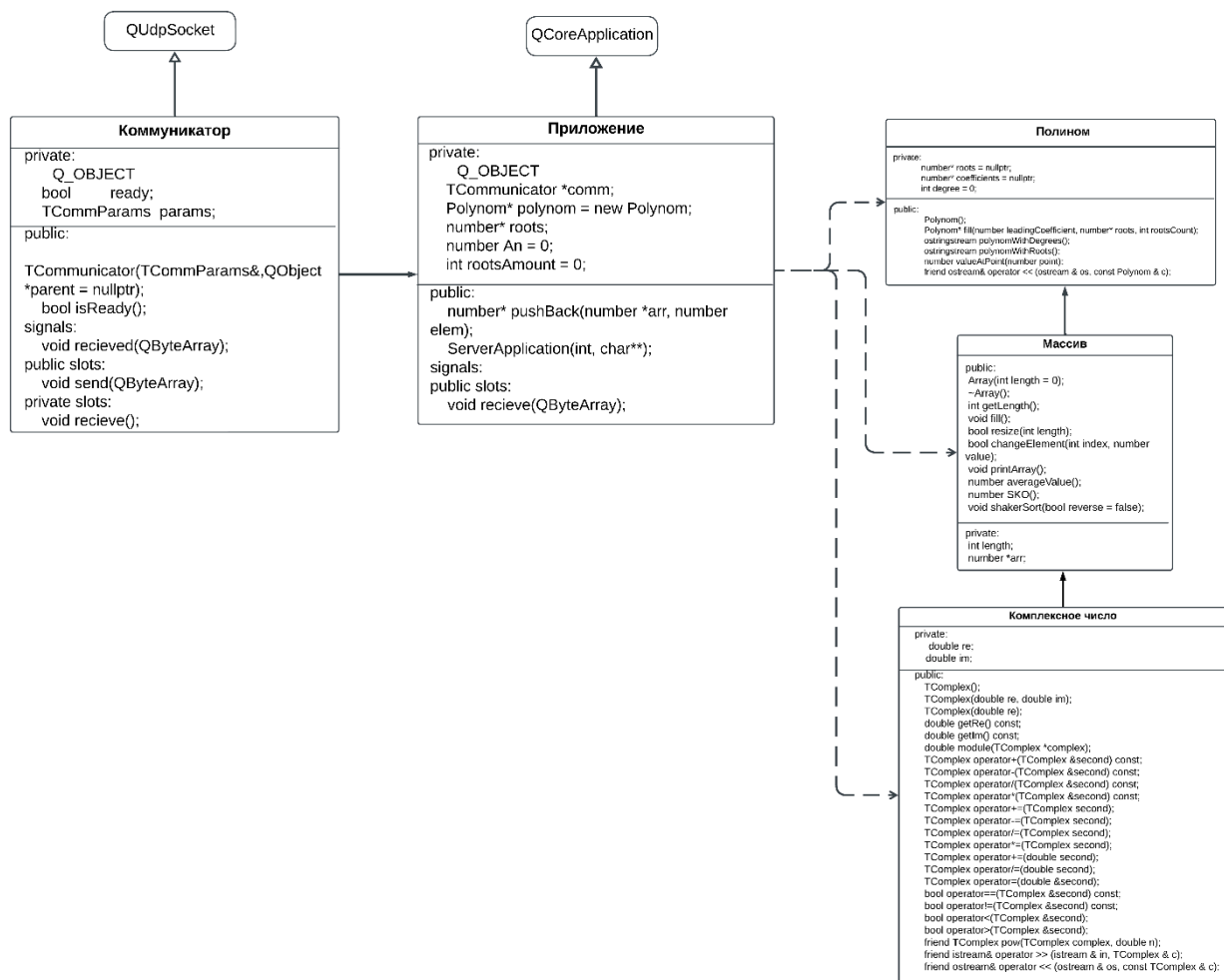


Рисунок 1 – Диаграмма классов.

Описание контрольного примера с исходными и ожидаемыми расчетными данными

1. Выбираем режим вещественных чисел
2. При создании полинома вводится коэффициент a_n : 1 и корни полинома: 2, 3, 4, 5
3. Создается полином $p(x) = x^4 - 14x^3 + 71x^2 - 154x + 120$
4. Изменяется коэффициент a_n на (3). Полином принимает вид:
$$p(x) = 3x^4 - 42x^3 + 213x^2 - 462x + 360$$
5. Предусмотрено изменение корня вводом индекса корня и новым значением: 1 и 10 соответственно. Полином меняет вид на: $p(x) = 3x^4 - 63x^3 + 444x^2 - 1260x + 1200$
6. Для вычисления значения функции в точке, необходимо ввести x : 3.
Значение: $p(3) = -42$

*Для режим комплексных чисел - то же самое, что и в прошлой работе

СКРИНШОТЫ РАБОТЫ ПРОГРАММЫ НА КОНТРОЛЬНЫХ ПРИМЕРАХ

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

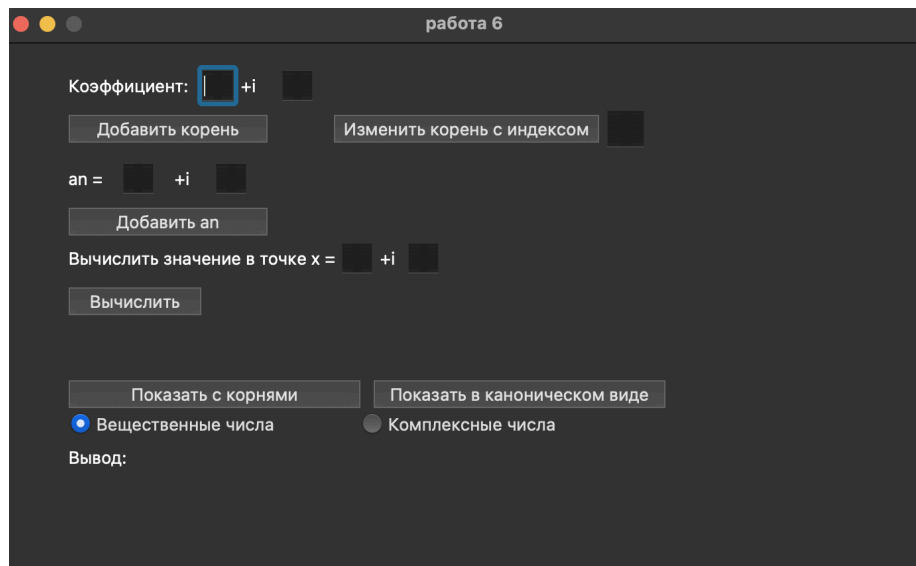


Рисунок 2 – Запуск программы

Для создания полинома нужно коэффициент a_n и корни полинома. Вводятся коэффициент a_n и корни полинома в соответствующие поля. Далее нужно нажать кнопки «Добавить корень» и «Добавить a_n ». Кнопку «Добавить корень» нужно нажимать после каждого введенного нового корня. На рисунке 3 показан ввод коэффициента a_n и корней полинома.

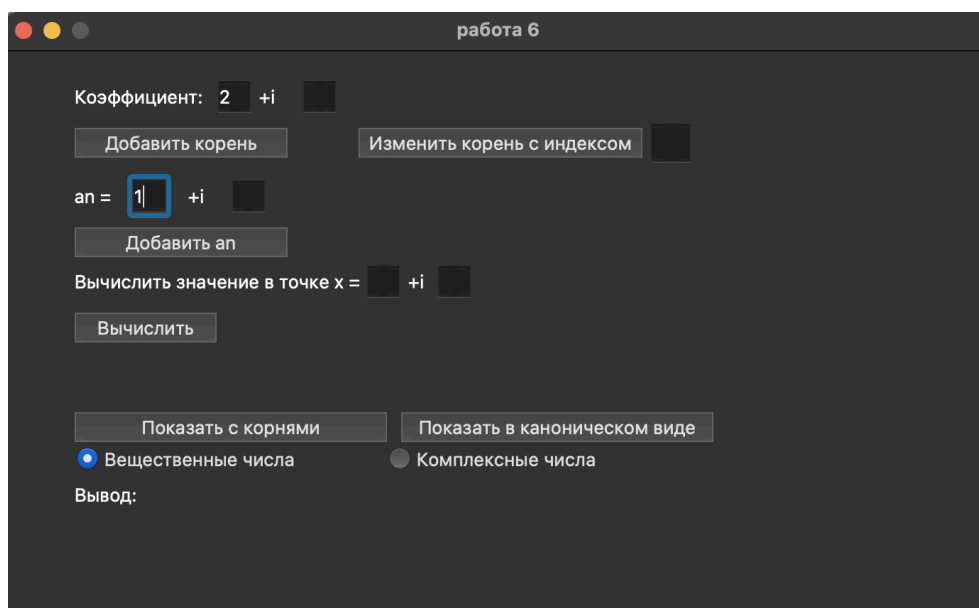


Рисунок 3 – Ввод корня и коэффициента из контрольного примера

Чтобы вывести полином, нажмем кнопку «Показать с корнями» или «Показать в каноническом виде». В окне появится полином с введенными на предыдущем шаге корнями и коэффициентом (рисунок 4).

The screenshot shows a software window titled "работа 6". It contains several input fields and buttons. The "Коэффициент:" field is set to "5" with a blue selection box around it, followed by "+i" and "0". Below it are buttons "Добавить корень" and "Изменить корень с индексом". The "a_n =" field is set to "1" with a blue selection box around it, followed by "+i" and "0". Below it is a button "Добавить a_n". The "Вычислить значение в точке x =" field is empty, followed by "+i" and another empty field. Below it is a button "Вычислить". At the bottom, there are two buttons: "Показать с корнями" and "Показать в каноническом виде". Below these are two radio buttons: "Вещественные числа" (selected) and "Комплексные числа". The polynomial expression $p(x) = x^4 (-14) x^3 + (71) x^2 (-154)x + (120)$ is displayed at the bottom.

Рисунок 4 – Вывод полинома на экран

Изменим старший коэффициент полинома. Введем 3 в поле ввода коэффициента и нажмем кнопку добавить a_n. На рисунке 5 показаны результаты работы программы.

The screenshot shows the same software window "работа 6" as before, but with the "a_n =" field now set to "3" with a blue selection box around it. The polynomial expression at the bottom has been updated to $p(x) = (3) x^4 (-42) x^3 + (213) x^2 (-462)x + (360)$. All other elements, including the coefficient field (5 +i 0), buttons, and radio buttons, remain the same.

Рисунок 5 – Изменение старшего коэффициента

Теперь изменим один из корней. Вводим в соответствующие поля индекс корня и новое значение и нажимаем кнопку «Изменить корень с индексом». Результат замены корня видно на рисунке 6.

The screenshot shows a software window titled "работа 6" with a dark background. It contains several input fields and buttons for polynomial manipulation. At the top, there is a label "Коэффициент:" followed by a text input field containing "10", a "+i" label, and a numeric input field containing "0". Below this are two buttons: "Добавить корень" and "Изменить корень с индексом", with the latter followed by a numeric input field containing "1". Further down, there is a label "an =" followed by a numeric input field containing "3", a "+i" label, and a numeric input field containing "0", with a "Добавить an" button below it. Below that is a label "Вычислить значение в точке x =" followed by two empty numeric input fields separated by a "+i" label, and a "Вычислить" button below it. At the bottom, there are two buttons: "Показать с корнями" and "Показать в каноническом виде". Below these are two radio buttons: "Вещественные числа" (which is selected) and "Комплексные числа". At the very bottom, the polynomial expression is displayed:
$$p(x) = (3) x^4 (-42) x^3 + (213) x^2 (-462)x + (360)$$

Рисунок 6 – Изменение корня полинома

Чтобы вычислить значение полинома в данной точке, необходимо ввести «3» и нажать клавишу Enter. На экране появится запрос точки x. На рисунке 7 показан результат работы программы.

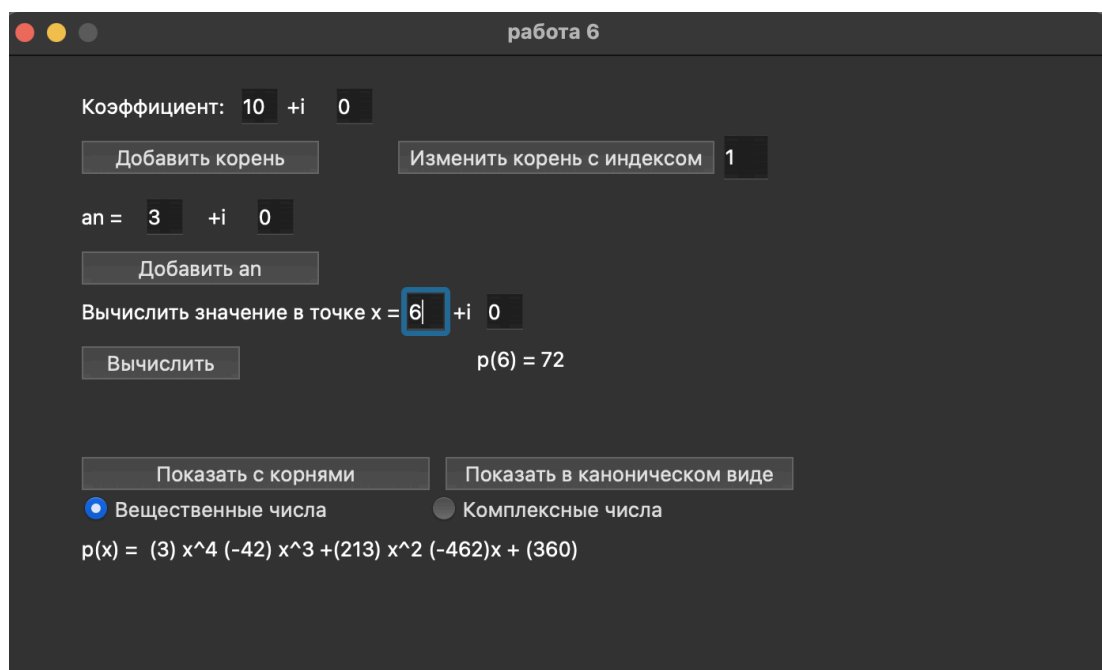


Рисунок 7 – Вычисленное значение

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 №5, но в нём классы “Массив” и “Полином” параметризуются. Также был разработан контрольный пример для проведения проверки, с чем программа справилась успешно.