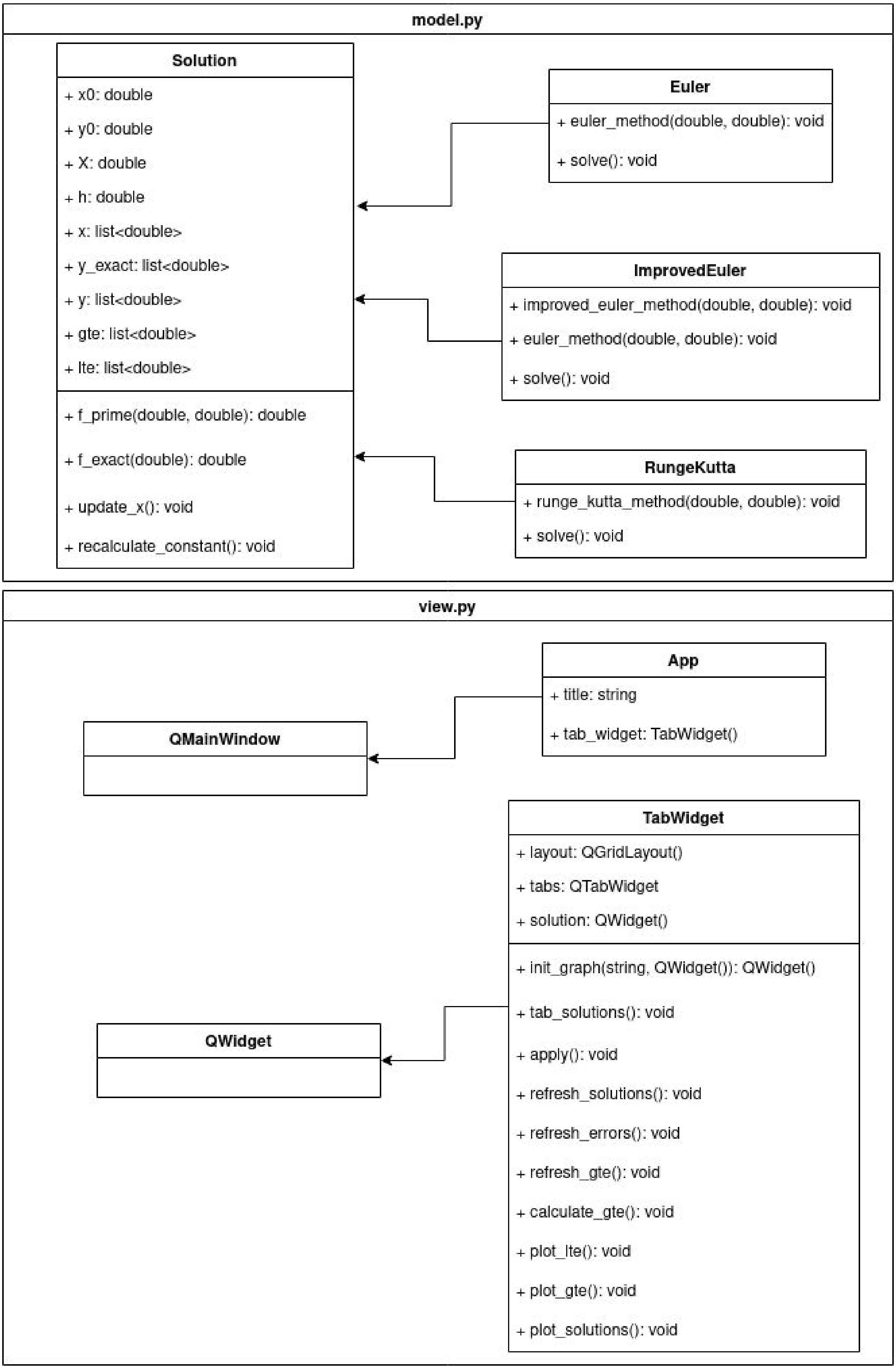
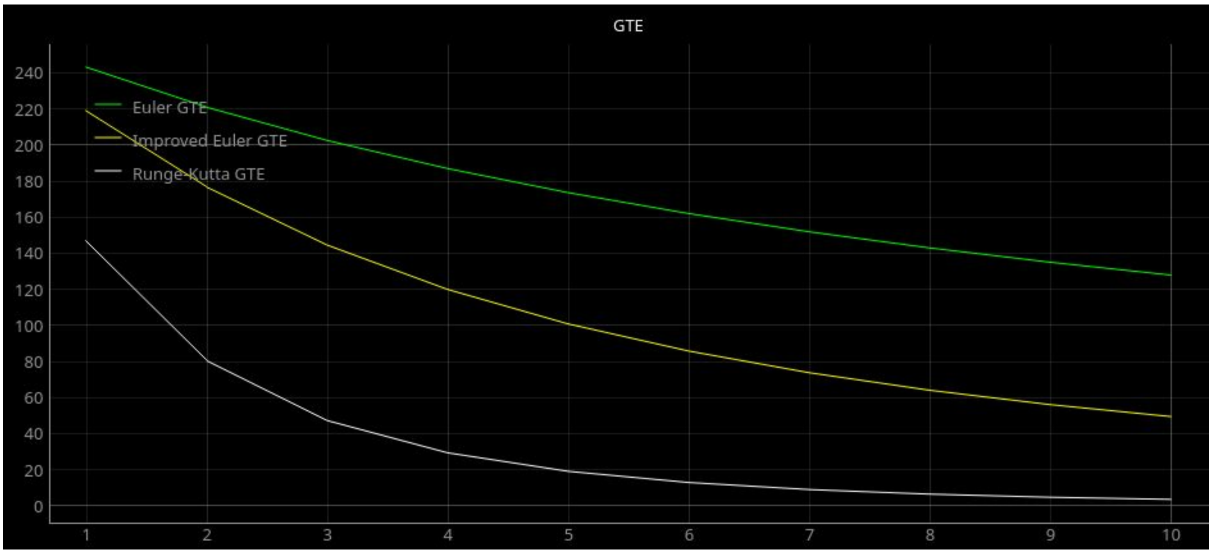
Computational Practicum. Variant 1

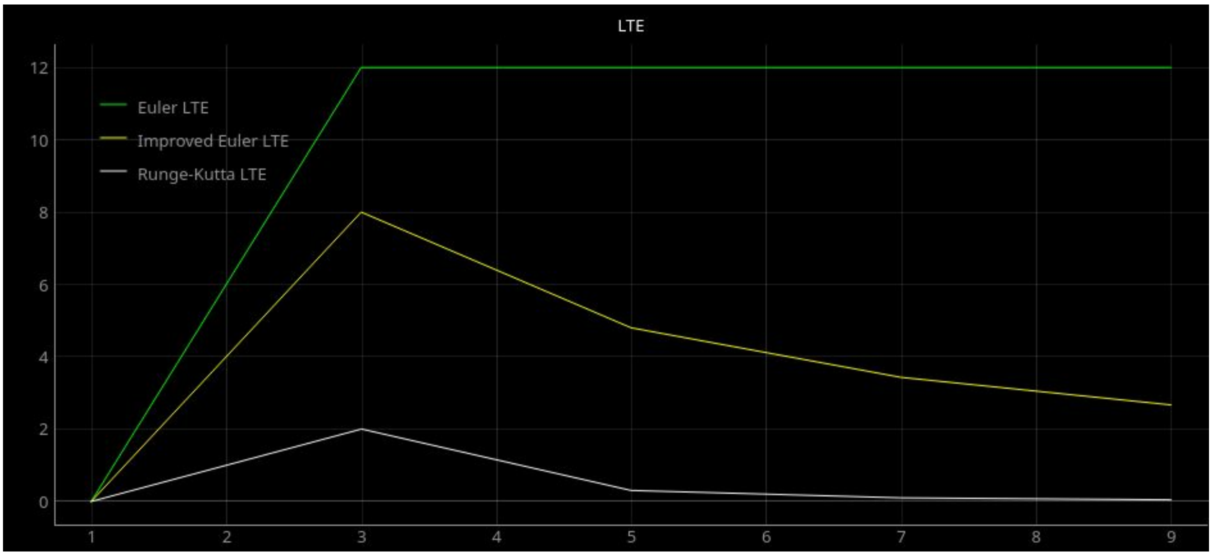
```
class ImprovedEuler(Solution):
	def __init__(self, x0, y0, h, X):
	super().__init__(x0, y0, h, X)

	def improved_euler_method(self, X, Y):
	return (Y + 0.5 * self.h * (self.f_prime(X, Y) + self.f_prime(X + 
self.h, self.euler_method(X, Y))))

	def euler_method(self, X, Y):
	return (Y + self.h * (self.f_prime(X, Y)))

	def solve(self):
	while (self.x[-1] < self.X - self.h/2):
		y_improved = self.improved_euler_method(self.x[-1], 
self.y[-1])
		self.y.append(y_improved)
		self.update_x()
		exact = self.f_exact(self.x[-1])
		self.y_exact.append(exact)
		self.gte.append(abs(exact - self.y[-1]))
		self.lte.append(abs(exact - 
self.improved_euler_method(self.x[-2], self.f_exact(self.x[-2]))))

class RungeKutta(Solution):
	def __init__(self, x0, y0, h, X):
	super().__init__(x0, y0, h, X)

	def runge_kutta_method(self, X, Y):
	k1 = self.f_prime(X, Y)
	k2 = self.f_prime(X + 0.5 * self.h, Y + 0.5 * self.h * k1)
	k3 = self.f_prime(X + 0.5 * self.h, Y + 0.5 * self.h * k2)
	k4 = self.f_prime(X + self.h, Y + self.h * k3)
	return (Y + self.h * (k1 + 2 * k2 + 2 * k3 + k4) / 6)

	def solve(self):
	while (self.x[-1] < self.X - self.h/2):
		y_runge = self.runge_kutta_method(self.x[-1], self.y[-1])
		self.y.append(y_runge)
		self.update_x()
		exact = self.f_exact(self.x[-1])
		self.y_exact.append(exact)
		self.gte.append(abs(exact - self.y[-1]))
		self.lte.append(abs(exact - 
self.runge_kutta_method(self.x[-2], self.f_exact(self.x[-2]))))
```

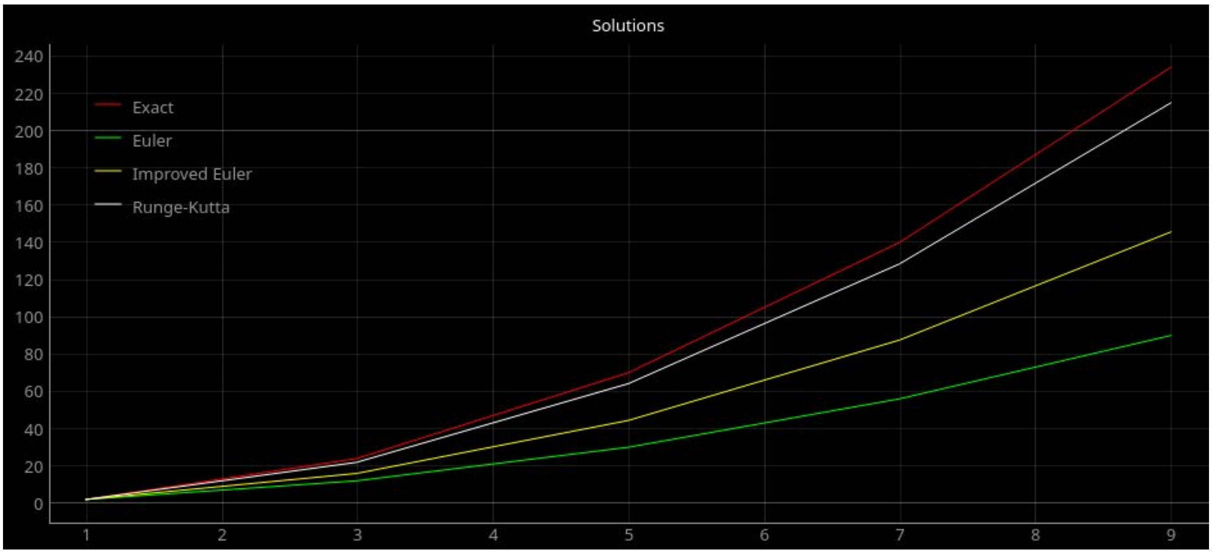
UML DIAGRAM


model.py
Solution
+ x0: double
+ y0: double
+ X: double
+ h: double
+ x: list<double>
+ y_exact: list<double>
+ y: list<double>
+ gte: list<double>
+ lte: list<double>
+ f_prime(double, double): double
+ f_exact(double): double
+ update_x(): void
+ recalculate_constant(): void
Euler
+ euler_method(double, double): void
+ solve(): void
ImprovedEuler
+ improved_euler_method(double, double): void
+ euler_method(double, double): void
+ solve(): void
RungeKutta
+ runge_kutta_method(double, double): void
+ solve(): void
view.py
App
+ title: string
+ tab_widget: TabWidget()
QMainWindow
TabWidget
+ layout: QGridLayout()
+ tabs: QTabWidget
+ solution: QWidget()
+ init_graph(string, QWidget()): QWidget()
+ tab_solutions(): void
+ apply(): void
+ refresh_solutions(): void
+ refresh_errors(): void
+ refresh_gte(): void
+ calculate_gte(): void
+ plot_lte(): void
+ plot_gte(): void
+ plot_solutions(): void
QWidget

Graphs


GTE
Euler GTE
Improved Euler GTE
Runge-Kutta GTE
240
220
200
180
160
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9
10


LTE
Euler LTE
Improved Euler LTE
Runge-Kutta LTE
12
10
8
6
4
2
0
1
2
3
4
5
6
7
8
9


Solutions
Exact
Euler
Improved Euler
Runge-Kutta
240
220
200
180
160
140
120
100
80
60
40
20
0
1
2
3
4
5
6
7
8
9

x0 = 1.0
y0 = 2.0
X = 10.0
h = 1.0

Euler Method

	X	Y (exact)	LTE	Y (approximated)	GTE
0	1	2.0	0.0	2.0	0.0
1	3	24.0	12.0	12.0	12.0
2	5	70.0	12.0	30.0	40.0
3	7	140.0	12.0	56.0	84.0
4	9	234.0	12.0	90.0	144.0

Improved Euler Method

	X	Y (exact)	Y (approximated)	LTE	GTE
0	1	2.0	2.0	0.0	0.0
1	3	24.0	16.0	8.0	8.0
2	5	70.0	44.4	4.799999999999997	25.6
3	7	140.0	87.56571428571428	3.428571428571445	52.43428571428572
4	9	234.0	145.60743764172335	2.666666666666657	88.39256235827665

Runge-Kutta Method

	X	Y (exact)	Y (approximated)	LTE	GTE
0	1	2.0	2.0	0.0	0.0
1	3	24.0	22.0	2.0	2.0
2	5	70.0	64.16666666666666	0.3000000000001137	5.833333333333343
3	7	140.0	128.47883597883595	0.095238095238102	11.521164021164054
4	9	234.0	214.91640946502054	0.041666666666685614	19.083590534979464