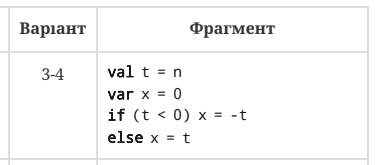
Лабораторна робота №3  
Розгалуження

1. Мета

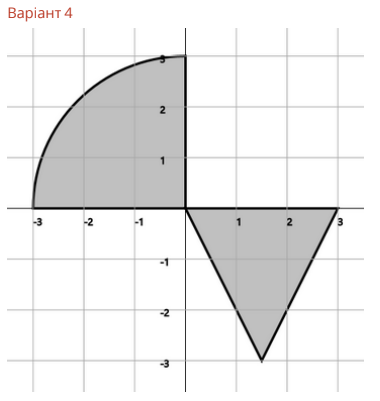
Отримання практичних навичок з розробки програм, що реалізують

алгоритми з розгалуженням.

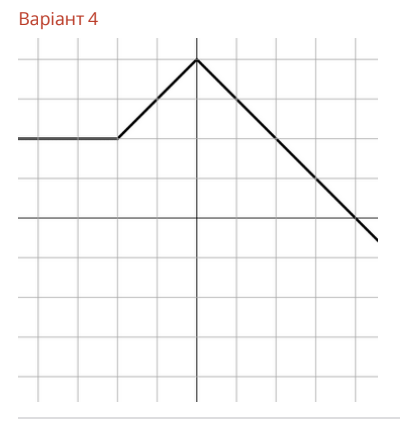
1. Завдання
2. Представити математичний запис фрагмента програми та обчислити значення змінної X після його виконання. Позначення: n – номер варіанту.



1. Написати функцію, що виводить на екран значення true, якщо точка А з координатами х, у належить до заштрихованої області, та false в іншому випадку.



1. Скласти програму, що містить опис функції, яка задана графічно. Доповнити програму функцією main, яка викликає створену функцію для значення аргументу x, що треба вводити з клавіатури, та виводить результат обчислень на консоль.



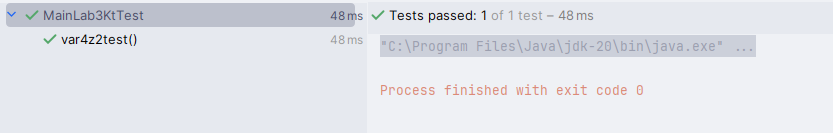
1. Практична частина
2. Лістинг програми:

fun *var4z2*(*x*: *Double*, *y*: *Double*): *Boolean* {  
 val r = 3.0 *// circle radius  
 // points of a triangle* val (Ax, Ay) = *listOf*(0.0, 0.0)  
 val (Bx, By) = *listOf*(3.0, 0.0)  
 val (Cx, Cy) = *listOf*(1.5, -3.0)  
 var *result* = false  
  
 if (*x* < 0.0 && *y* > 0) {  
 *result* = *inCircle*(*x*, *y*, r)  
 } else if (*x* > 0 && *y* < 0) {  
 *result* = *inTriangle*(*x*, *y*, Ax, Ay, Bx, By, Cx, Cy)  
 }  
 return *result*}  
  
fun *inCircle*(*x*: *Double*, *y*: *Double*, *r*:*Double*): *Boolean* {  
 return *x*.*pow*(2) + *y*.*pow*(2) <= *r*.*pow*(2)  
}  
  
fun *inTriangle*(*x*: *Double*, *y*: *Double*, *Ax*: *Double*, *Ay*: *Double*, *Bx*: *Double*, *By*: *Double*, *Cx*: *Double*, *Cy*: *Double*): *Boolean* {  
 return (((*x* - *Ax*) \* (*By* - *Ay*) - (*y* - *Ay*) \* (*Bx* - *Ax*)) \* ((*Cx* - *Ax*) \* (*By* - *Ay*) - (*Cy* - *Ay*) \* (*Bx* - *Ax*))) >= 0 &&  
 (((*x* - *Bx*) \* (*Cy* - *By*) - (*y* - *By*) \* (*Cx* - *Bx*)) \* ((*Ax* - *Bx*) \* (*Cy* - *By*) - (*Ay* - *By*) \* (*Cx* - *Bx*))) >= 0 &&  
 (((*x* - *Cx*) \* (*Ay* - *Cy*) - (*y* - *Cy*) \* (*Ax* - *Cx*)) \* ((*Bx* - *Cx*) \* (*Ay* - *Cy*) - (*By* - *Cy*) \* (*Ax* - *Cx*))) >= 0  
}

Лістинг тестів:

*@Test  
@Tag*("var4")  
fun *var4z2test*() {  
 *assertFalse*(*var4z2*(-1.0, -1.0))  
 *assertTrue*(*var4z2*(1.0, -1.0))  
 *assertTrue*(*var4z2*(-2.0, 2.0))  
 *assertFalse*(*var4z2*(2.0, -3.0))  
}

Результати:



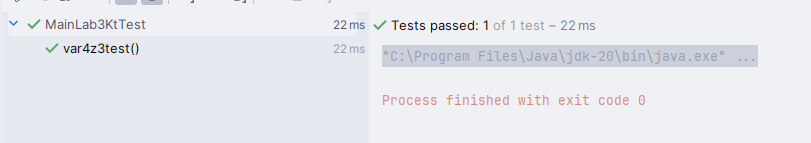
1. Лістинг програми:

fun *var4z3*(*x*: *Double*):*Double* {  
 return when {  
 *x* <= -2 -> 2.0  
 *x* <= 0 -> *x* + 4  
 else -> -*x* + 4  
 }  
}

Лістинг тестів:

*@Test  
@Tag*("var4")  
fun *var4z3test*() {  
 *assertEquals*(2.0, *var4z3*(-3.0))  
 *assertEquals*(3.5, *var4z3*(-0.5))  
 *assertEquals*(3.5, *var4z3*(0.5))  
 *assertEquals*(0.0, *var4z3*(4.0))  
 *assertEquals*(4.0, *var4z3*(0.0))  
}

Результати:



1. Висновки

На цій лабораторній роботі я отримання практичних навичок з розробки програм, що реалізують алгоритми з розгалуженням на мові програмування Kotlin.