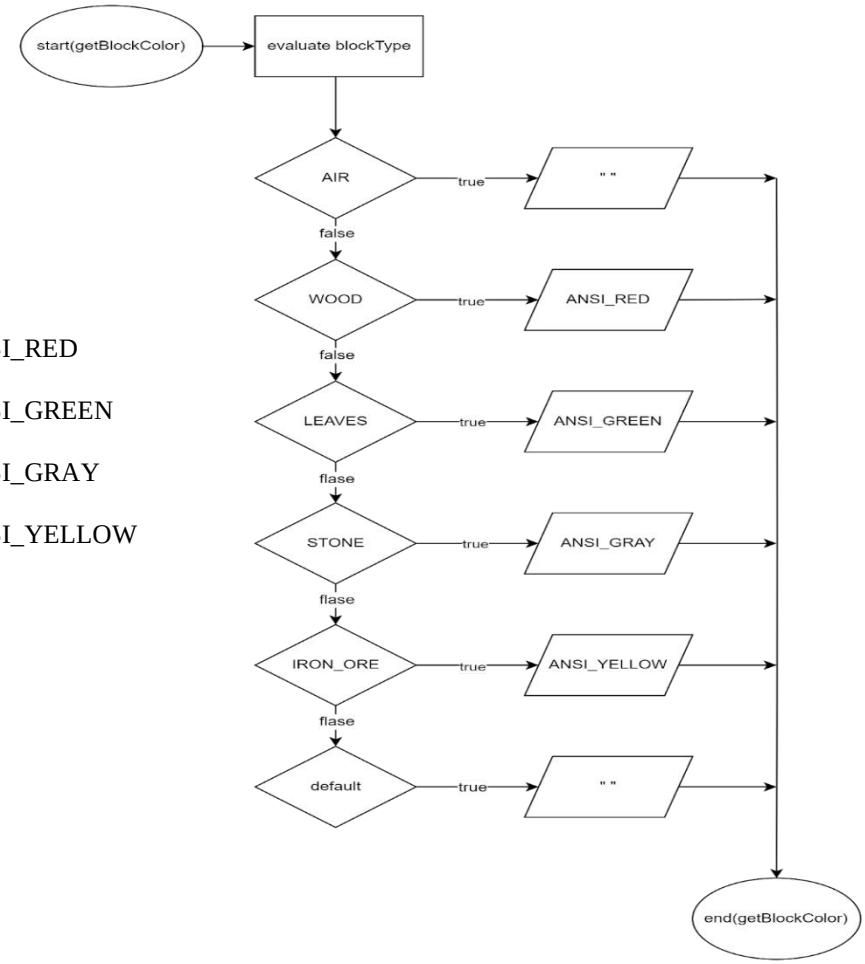
Method: getBlockColor

Start
Switch for the value of blockType

1. Case blockType = AIR
   Return the output: “ “
2. Case blockType = WOOD
   Return the output: ANSI_RED
3. Case blockType = LEAVES
   Return the output: ANSI_GREEN
4. Case blockType = STONE
   Return the output: ANSI_GRAY
5. Case blockType = IRON_ORE
   Return the output: ANSI_YELLOW
6. Otherwise:
   Return the output: “ “

End

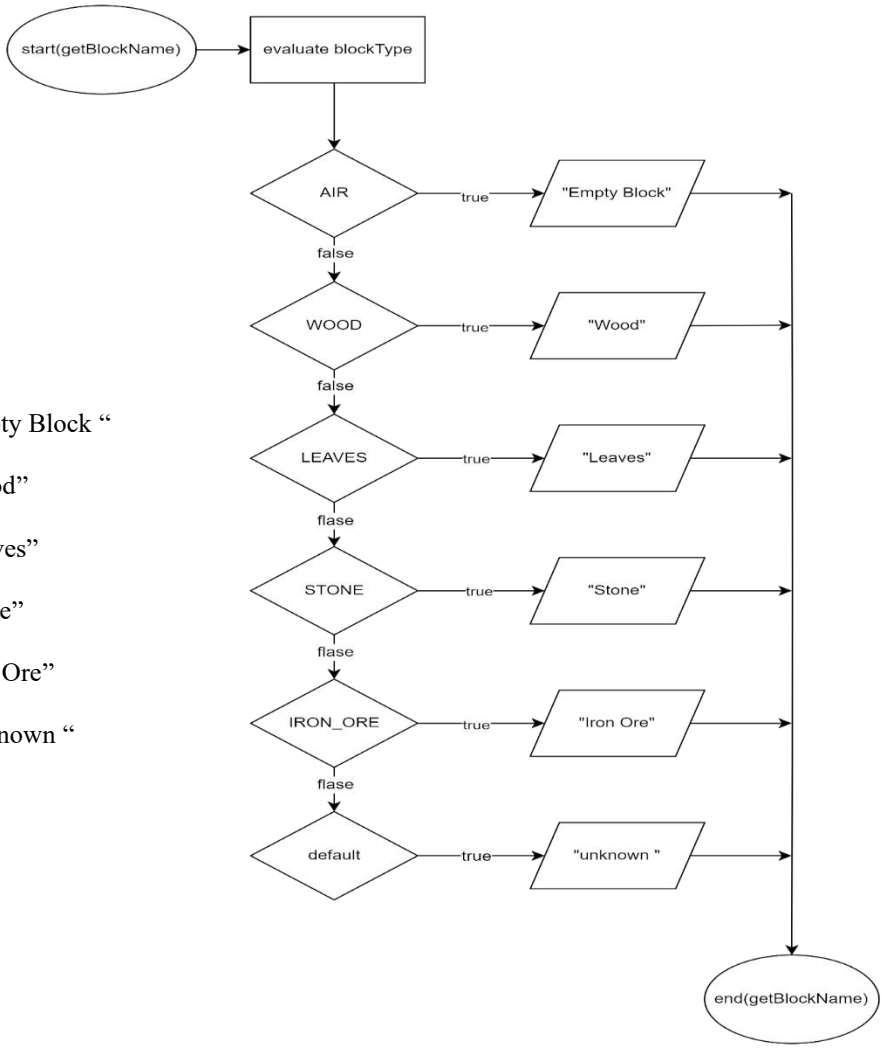


Method: getBlockName

Start
Switch for the value of blockType

1. Case blockType = AIR
   Return the output: “Empty Block “
2. Case blockType = WOOD
   Return the output: “Wood”
3. Case blockType = LEAVES
   Return the output: “Leaves”
4. Case blockType = STONE
   Return the output: “Stone”
5. Case blockType = IRON_ORE
   Return the output: “Iron Ore”
6. Otherwise:
   Return the output: “Unknown “

End

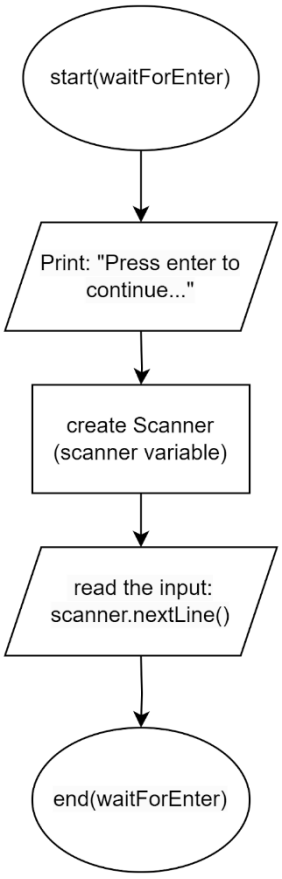


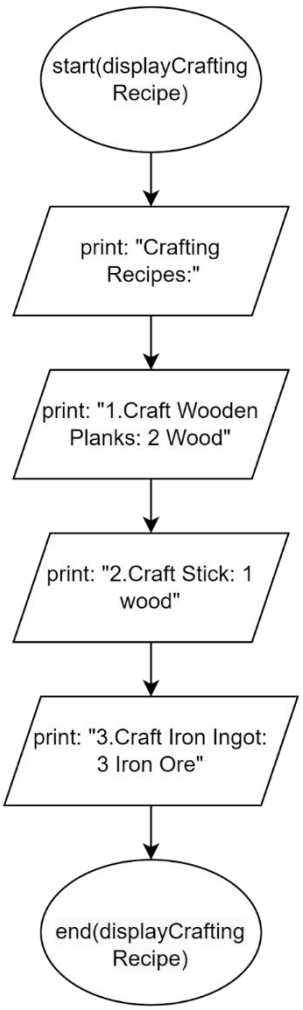
Method: waitForEnter

Start
Print “Press enter to continue…”
Create a new Scanner object named scanner
Call scanner.nextLine(), (getting the input from the user)
end

Method: displayCraftingRecipes

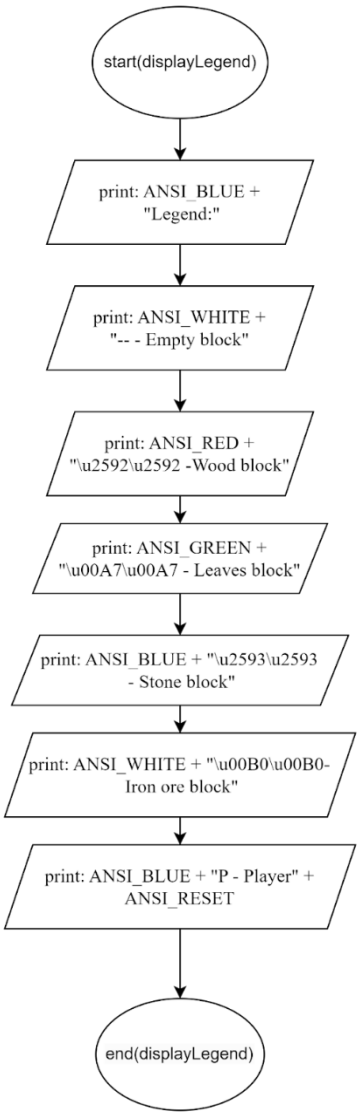
Start
Print “Crafting Recipes:”
Print “1. Craft Wooden Planks: 2 Wood”
Print “2. Craft Stick: 1 Wood”
Print “3. Craft Iron Ingot: 3 Iron Ore”
end





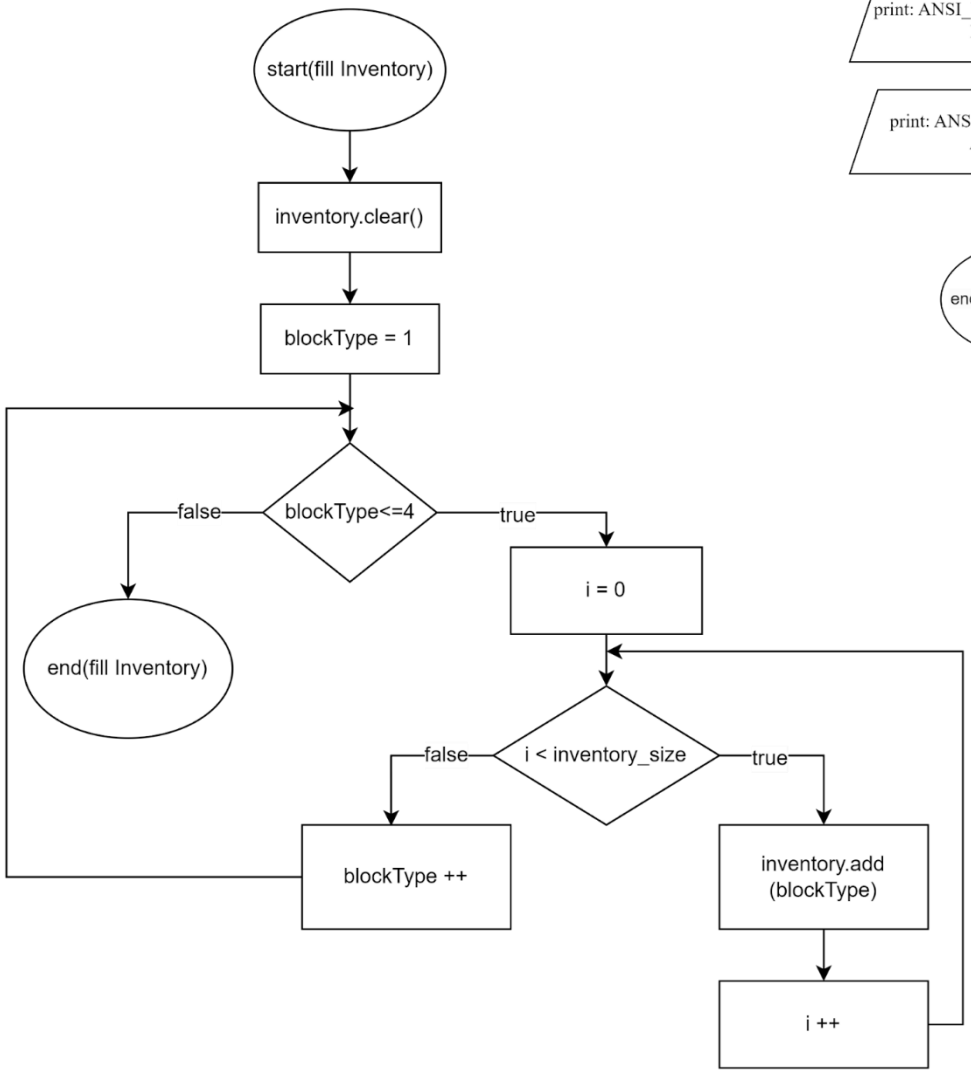
Method: displayLegend

Start
Print ANSI_BLUE + "Legend:"
Print ANSI_WHITE + "-- - Empty block"
Print ANSI_RED + "\u2592\u2592 - Wood block"
Print ANSI_GREEN + "\u00A7\u00A7 - Leaves block"
Print ANSI_BLUE + "\u2593\u2593 - Stone block"
Print ANSI_WHITE + "\u00B0\u00B0- Iron ore block"
Print ANSI_BLUE + "P - Player" + ANSI_RESET
end



Method: fillInventory

Start
Call inventory.clear() method to clear existing inventory
For each blockType from 1 to 4
    For i from 0 to (INVENTORY_SIZE - 1):
        Add blockType to Inventory
end

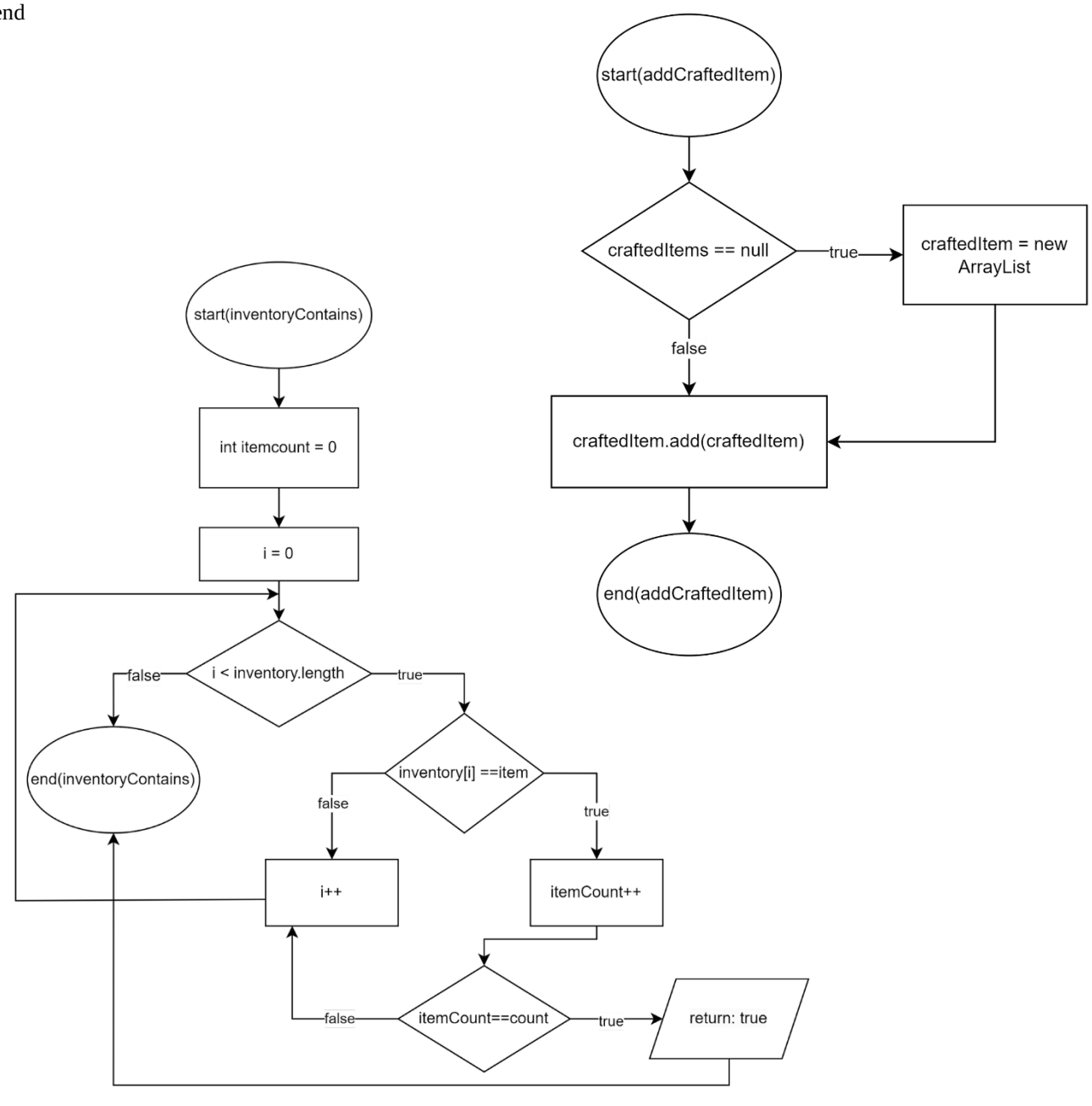


Method: inventoryContains

Start
Set itemCount to 0
For each i in inventory
    If i is equal to item:
        Increment itemCount by 1
        If itemCount is equal to count
            return : true
Return false (if the required count is not reached)
end

Method: addCraftedItem

Start
If craftedItems list is equal to null
    Create a new ArrayList (craftedItem = Create a new ArrayList)
Add the craftedItem to the craftedItem list
end



Method: getCraftedItemName

Start
Switch for the value of craftedItem

1. Case craftedItem = CRAFTED_WOODEN_PLANKS
   Return the output: “Wooden Planks “
2. Case craftedItem = CRAFTED_STICK
   Return the output: “Stick “
3. Case craftedItem = CRAFTED_IRON_INGOT
   Return the output: “Iron Ingot “
4. Otherwise:
   Return the output: “Unknown “

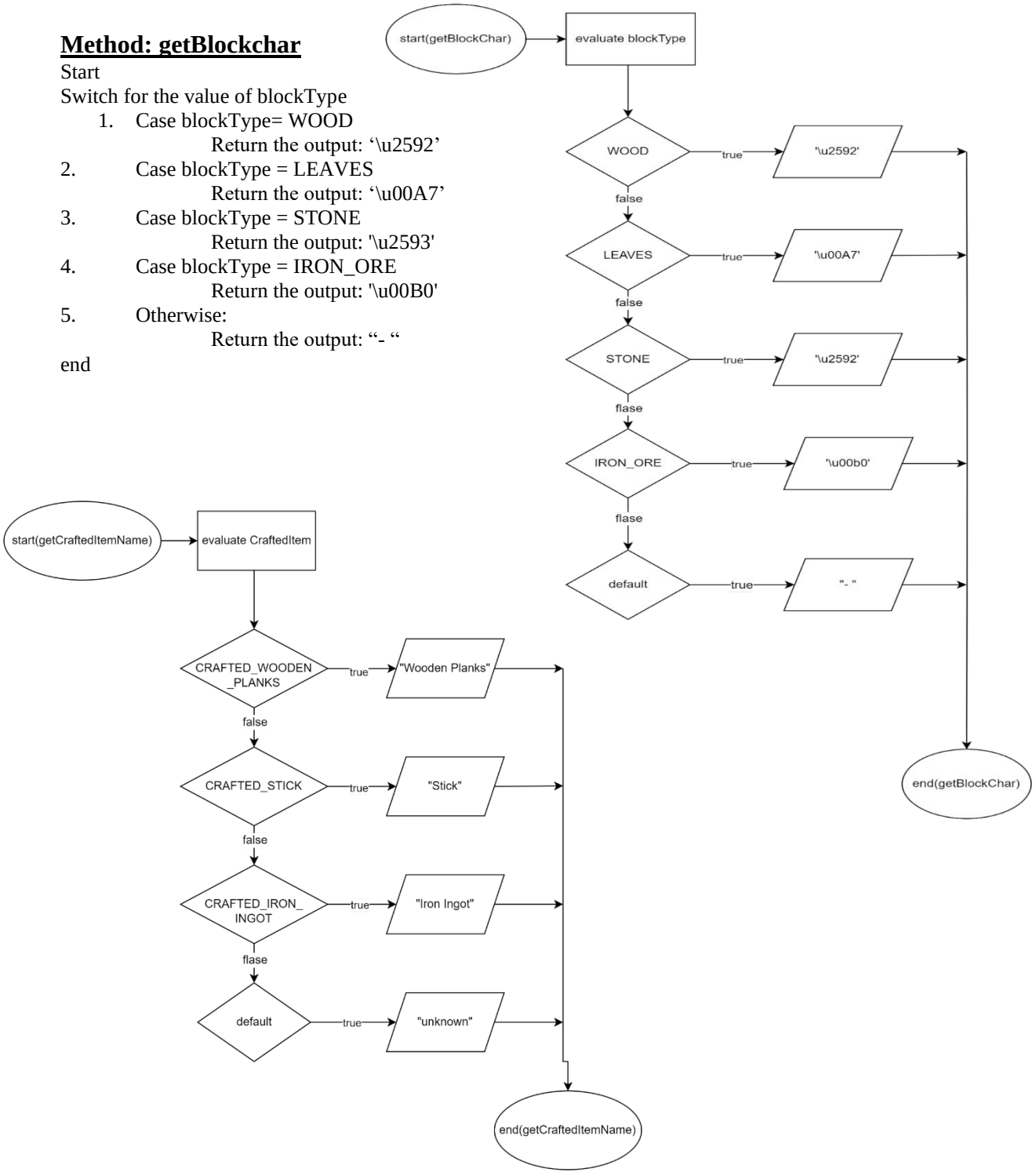
end

Method: getBlockchar

Start
Switch for the value of blockType

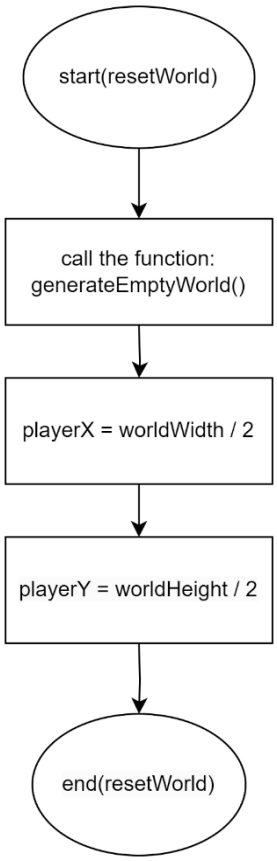
1. Case blockType= WOOD
   Return the output: ‘\u2592’
2. Case blockType = LEAVES
   Return the output: ‘\u00A7’
3. Case blockType = STONE
   Return the output: '\u2593'
4. Case blockType = IRON_ORE
   Return the output: '\u00B0'
5. Otherwise:
   Return the output: “- “

end



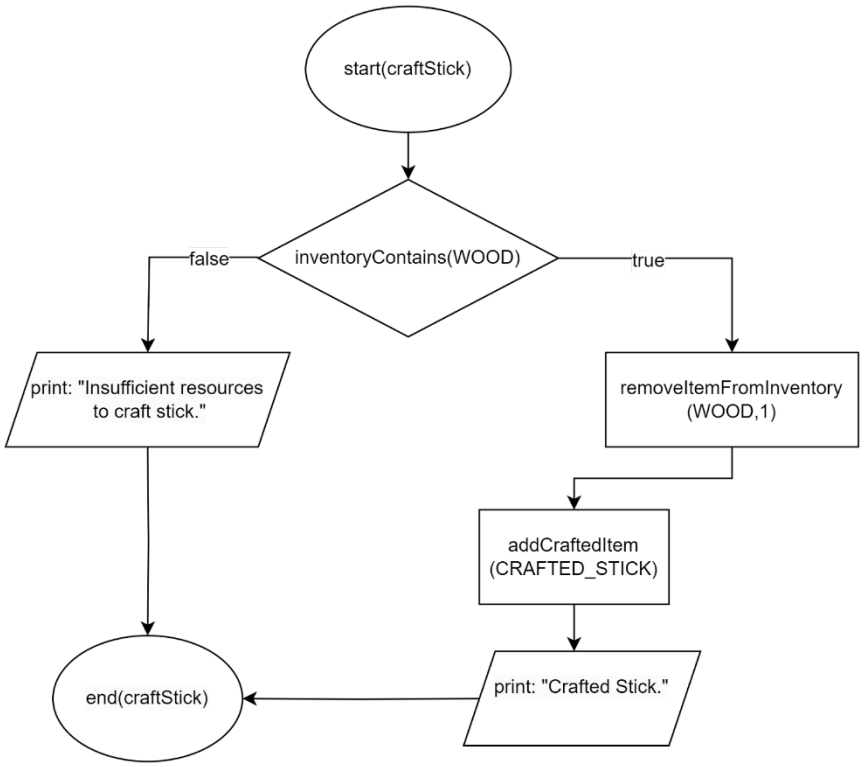
Method: resetWorld

Start
Call function to generate an empty world (generateEmptyWorld())
Set player's X coordinate to: worldWidth / 2
Set player's Y coordinate to: worldHeight / 2
end



Method: craftStick

Start
If InventoryContains(WOOD):
    Remove 1 wood from the inventory //use removeItemsFromInventory(WOOD,1)
    Add the crafted stick to the inventory //use addCraftedItem(CRAFTED_STICK)
    Print the output : "Crafted Stick"
Else:
    Print the output: "Insufficient resources to craft Stick."
end

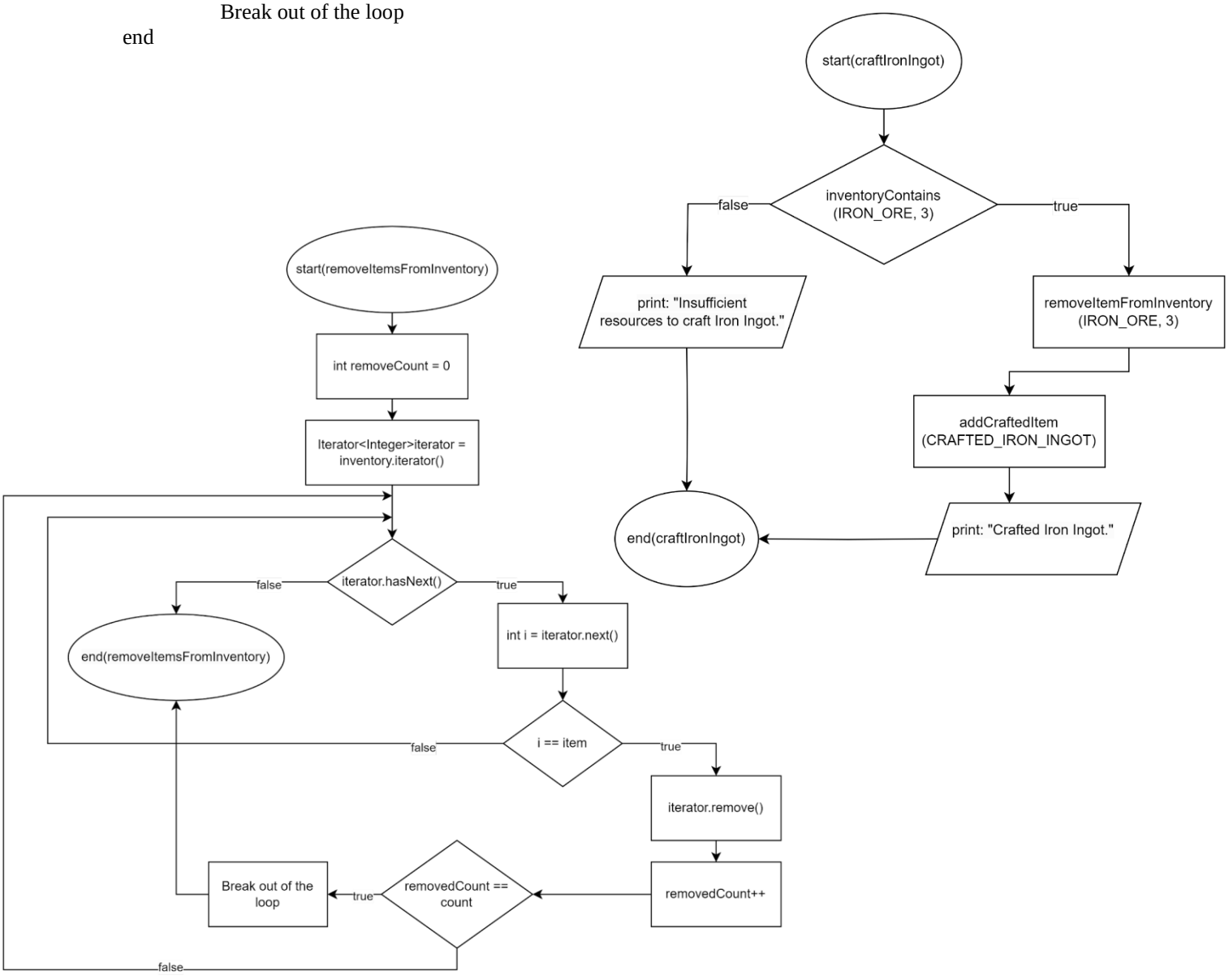


Method: craftIronIngot

Start
If InventoryContains(IRON_ORE, 3):
    Remove 3 iron ores from the inventory //use removeItemsFromInventory(IRON_ORE, 3)
    Add a crafted iron ore ingot to the inventory //use addCraftedItem(CRAFTED_IRON_INGOT)
    Print the output : “Crafted Iron Ingot.”
Else:
    Print the output: “Insufficient resources to craft Iron Ingot.”
End

Method: removeItemsFromInventory

Start
Set removedCount to 0
Create iterator and set it to the beginning of inventory
While iterator has next element
    Set i to iterator.next()
    If i is equal to item
        Remove the current item using iterator
        Increment removeCount by 1
        If removeCount is equal to Count (meaning the required count of items is reached)
            Break out of the loop
end



Method: mineBlock

Start
Get the type of block at player's position
If the blockTupe is not equal to air
    Add blockType to inventory
    Set the block at player's position to air
    Print: "Mined"+ getBlockName(blockType) + "."(Indicating the mined block)
Else
    print : "No block to mine here."
Wait for user input (waitForEnter())
End