Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Метрология, стандартизация и сертификация (в инормационных технологиях)»

**ОТЧЕТ**

к лабораторной работе №1

на тему:

**«Метрики размера программ»**

БГУИР 6-05-0612-02

|  |
| --- |
| Выполнили студентки группы 353501  ГОРОДЕЦКАЯ Диана Анатольевна  МАЛИНОВСКАЯ Евгения Александровна |
| Проверила доцент каф. Информатики  БОЛТАК Светлана Владимировна |
|  |
| (дата, подпись преподавателя) |

Минск 2025

Листинг 1 – Исходный код программы

case class Task(description: String, var isCompleted: Boolean = false)

class TodoList {

private var tasks: List[Task] = List()

def addTask(description: String): Unit = {

tasks = Task(description) :: tasks

println(s"Task added: '$description'")

}

def completeTask(description: String): Unit = {

tasks.find(\_.description == description) match {

case Some(task) =>

task.isCompleted = true

println(s"Task completed: '$description'")

case None =>

println(s"Task '$description' not found")

}

}

def showTasks(): Unit = {

if (tasks.isEmpty) println("Task list is empty.")

else {

println("Task list:")

tasks.zipWithIndex.foreach { case (task, index) =>

val status = if (task.isCompleted) "[x]" else "[ ]"

println(s"${index + 1}. $status ${task.description}")

}

}

}

def removeTask(description: String): Unit = {

tasks = tasks.filterNot(\_.description == description)

println(s"Task removed: '$description'")

}

}

object TodoApp extends App {

val todoList = new TodoList

def printHelp(): Unit = {

println("""Available commands:

|add <description> - add a task

|complete <description> - mark a task as completed

|remove <description> - remove a task

|list - show the list of tasks

|help - show help

|exit - exit""".stripMargin)

}

def processCommand(command: String): Unit = {

val parts = command.split(" ", 2)

parts(0) match {

case "add" if parts.length > 1 => todoList.addTask(parts(1))

case "complete" if parts.length > 1 => todoList.completeTask(parts(1))

case "remove" if parts.length > 1 => todoList.removeTask(parts(1))

case "list" => todoList.showTasks()

case "help" => printHelp()

case "exit" => System.exit(0)

case \_ => println("Unknown command. Type 'help' for assistance.")

}

}

println("Welcome to the To-Do List!")

printHelp()

while (true) {

print("> ")

scala.io.StdIn.readLine() match {

case command if command.nonEmpty => processCommand(command)

case \_ => // Ignore empty input

}

}

}

Словарь программы:

Длина программы:

Объем программы: