```
#!/bin/bash
# shellcheck source=kz_common.sh
################################################################################
# Check file names.
#
# Written in 2012 by Karel Zimmer <info@karelzimmer.nl>, Creative Commons
# Public Domain Dedication <https://creativecommons.org/publicdomain/zero/1.0>.
################################################################################

set -o errexit
program_path=$(cd "$(dirname "$(realpath "$0")")" && pwd)
source "$program_path"/kz_common.sh


################################################################################
# Constants
################################################################################

declare program_name='kz-ckname'
declare program_desc
program_desc=$(gettext 'Check file names')
declare display_name=${program_name/kz-/kz }

declare usage=
usage=$(eval_gettext "Usage: \$display_name \$options_usage [DIRECTORY...]")
declare help
help="$(eval_gettext "Usage: \$display_name [OPTION...] [DIRECTORY...]")

$program_desc.

$(gettext 'Options:')
$options_help

$(gettext 'Arguments:
  DIRECTORY      start checking from DIRECTORY')"


################################################################################
# Variables
################################################################################

declare argument_map=false
declare -a map_arguments=()


################################################################################
# Functions
################################################################################

function check_input {
    local -i getopt_rc=0
    local -i map_arg_num=0
    local parsed=''

    parsed=$(
        getopt  --alternative                       \
                --options       "$options_short"    \
                --longoptions   "$options_long"     \
                --name          "$display_name"     \
                --              "$@"
        ) || getopt_rc=$?
    if [[ $getopt_rc -ne $ok ]]; then
        info "$usage_line"
        exit $error
```

```
    fi
    eval set -- "$parsed"
    kz_common.process_options "$@"

    while true; do
        case $1 in
            --)
                shift
                break
                ;;
            *)
                shift
                ;;
        esac
    done

    while [[ "$*" ]]; do
        argument_map=true
        map_arguments[map_arg_num]=$1
        ((++map_arg_num))
        shift
    done
    if ! $argument_map; then
        map_arguments[0]=$HOME
    fi
    for dir in "${map_arguments[@]}"; do
        if ! [[ -d $dir ]]; then
            printf  "$display_name: $dir: %s\n$usage_line\n" \
                    "$(gettext 'directory does not exist')"
            exit $error
        fi
    done
}


function process_input {
    check_files
}


function check_files {
    local basename=''
    local -i count=0
    local dirname=''
    local file=''
    local -i filename_maxlen=142
    local good=''

    # It doesn't matter if the file name contains special characters like tab,
    # space, etc. The find with print0 takes care of this, and the read with
    # IFS= and as delimiter the null character that may not occur in a
    # filename. Note: In Linux everything is a file!
    while IFS= read -r -d $'\0' file; do

        dirname=$(dirname "$file")
        basename=$(basename "$file")

        # Remove the bad characters, \ is escape for \.
        good=$(printf '%s'"$file" | tr --delete '?"\\<>*|:')

        if ! [[ $file = "$good" ]]; then
            ((++count))
            if [[ -d "$file" ]]; then
                info "$(gettext 'BadName  DIR'): $file"
```

```
            elif [[ -f "$file" ]]; then
                info "$(gettext 'BadName FILE'): $basename ($dirname)"
            else
                info "$(gettext 'BadName SYML'): $file"
            fi
        fi

        if [[ ${#basename} -gt $filename_maxlen ]]; then
            ((++count))
            if [[ -d "$file" ]]; then
                info "$(gettext 'BadLen.  DIR'): $basename"
            else
                info "$(gettext 'BadLen. FILE'): $basename ($dirname)"
            fi
        fi

    done < <(
        find    "${map_arguments[@]}"   \
                -not -name '.*'         \
                -not -path '*/.*'       \
                -type f                 \
                -print0                 \
                -or                     \
                -not -name '.*'         \
                -not -path '*/.*'       \
                -type d                 \
                -print0                 \
                -or                     \
                -not -name '.*'         \
                -not -path '*/.*'       \
                -type l                 \
                -print0                 2> /dev/null || true
        )

    if [[ $count -eq 0 ]]; then
        return $ok
    elif [[ $count -eq 1 ]]; then
        info "$(gettext 'An error has been found.')"
    else
        info "$(eval_gettext "\$count errors were found.")"
    fi
}


function term_script {
    exit $ok
}


################################################################################
# Script
################################################################################

function main {
    kz_common.init_script "$@"
    check_input "$@"
    process_input
    term_script
}

main "$@"
```