1. Aiken-Mode

;;; aiken-mode.el --- Major mode for Aiken -\*- lexical-binding: t -\*-

;; Copyright © 2023 Sebastian Nagel <sebastian.nagel@ncoding.at>

;; Author: Sebastian Nagel <sebastian.nagel@ncoding.at>

;; URL: https://github.com/aiken-lang/aiken-mode

;; Keywords: languages aiken

;; Version: 1.0.2

;; Package-Requires: ((emacs "26.1"))

;; SPDX-License-Identifier: MPL-2.0

;; This file is NOT part of GNU Emacs.

;; This Source Code Form is subject to the terms of the Mozilla Public

;; License, v. 2.0. If a copy of the MPL was not distributed with this

;; file, You can obtain one at http://mozilla.org/MPL/2.0/.

;;; Commentary:

;; Provides syntax highlighting for the Aiken smart contract language.

;;; Code:

;; Aiken syntax

(defvar aiken-keywords

'("if"

"else"

"when"

"is"

"fn"

"use"

"let"

"pub"

"type"

"opaque"

"const"

"todo"

"error"

"expect"

"test"

"trace"

"fail"

"validator"

"and"

"or"))

(defvar aiken-operators

'(

"="

"->"

".."

"|>"

">="

"<="

">"

"<"

"!="

"=="

"&&"

"||"

"!"

"+"

"-"

"/"

"\*"

"%"

"?"))

(defvar aiken-font-lock-keywords

(append

`(

;; Keywords

(,(regexp-opt aiken-keywords 'symbols) . font-lock-keyword-face)

;; CamelCase is a type

("[[:upper:]][[:word:]]\*" . font-lock-type-face)

;; Operators

(,(regexp-opt aiken-operators nil) . font-lock-builtin-face))

;; Identifiers after keywords

(mapcar (lambda (x)

(list (concat (car x) "[^(]\\(\\w\*\\)")

1 ;; apply face ot first match group

(cdr x)))

'(("const" . font-lock-type-face)

("type" . font-lock-type-face)

("use" . font-lock-constant-face)

("fn" . font-lock-function-name-face)))))

;; Mode definitions

;;;###autoload

(define-derived-mode aiken-mode prog-mode "Aiken"

"Major mode for Aiken code."

:group 'aiken-mode

(setq-local indent-tabs-mode nil)

;; Syntax highlighting via font-lock

(setq-local font-lock-defaults '(aiken-font-lock-keywords))

;; Syntax: make \_ part of words

(modify-syntax-entry ?\_ "w" aiken-mode-syntax-table)

;; Comment syntax

(modify-syntax-entry ?/ ". 124b" aiken-mode-syntax-table)

(modify-syntax-entry ?\n "> b" aiken-mode-syntax-table)

(modify-syntax-entry ?\^m "> b" aiken-mode-syntax-table)

;; Comment settings

(setq-local comment-start "// ")

(setq-local comment-end "")

(setq-local comment-start-skip "//+ \*")

(setq-local comment-use-syntax t)

(setq-local comment-auto-fill-only-comments t))

;;;###autoload

(add-to-list 'auto-mode-alist '("\\.ak\\'" . aiken-mode))

(provide 'aiken-mode)

;;; aiken-mode.el ends here