

Projet 1 : Générateur de mots de passe sécurisé

Objectif : Implémenter un programme pour générer des mots de passe aléatoires selon divers critères (longueur, présence de chiffres, symboles, majuscules).

Compétences couvertes :

- Structures de contrôle (`if`, `switch`).
- Manipulation de chaînes de caractères.
- Boucles et conditions.

Fonctionnalités :

1. Permettre à l'utilisateur de choisir la longueur du mot de passe.
2. Inclure des critères personnalisables (majuscules, chiffres, symboles).
3. Générer des mots de passe sécurisés en mélangeant ces caractères.
4. Afficher le mot de passe généré dans la console.

Projet 2 : Analyseur de texte

Objectif : Créer un programme qui analyse un texte donné et affiche des statistiques (nombre de mots, de caractères, de phrases, etc.).

Compétences couvertes :

- Manipulation de chaînes de caractères.
- Boucles (`for`, `while`).
- Conditions (`if`, `else`).
- Gestion de tableaux.

Fonctionnalités :

1. Compter le nombre de mots et de phrases dans un texte donné.
2. Analyser la fréquence des lettres.
3. Détecter les mots les plus utilisés.
4. Afficher les résultats sous forme de résumé dans la console.

Projet 3 : Simulation d'une file d'attente

Objectif : Implémenter une simulation de file d'attente pour gérer les clients dans une file (par exemple, dans une banque).

Compétences couvertes :

- Utilisation de classes et d'objets.
- Structures de données de base (tableaux, files d'attente).
- Algorithmes simples (ajouter, retirer, gérer des priorités).

Fonctionnalités :

1. Ajouter des clients à la file d'attente avec un numéro de ticket.
2. Retirer les clients de la file d'attente.
3. Implémenter une logique pour les clients prioritaires.
4. Afficher l'état actuel de la file d'attente dans la console.

Projet 4 : Calculateur de matrices

Objectif : Créer un programme pour effectuer des opérations sur des matrices (addition, multiplication, transposition).

Compétences couvertes :

- Manipulation de tableaux.
- Utilisation de boucles imbriquées.
- Gestion d'erreurs (dimensions incompatibles).

Fonctionnalités :

1. Créer une matrice en saisissant ses dimensions et ses éléments.
2. Implémenter l'addition et la multiplication de matrices.
3. Afficher les résultats sous forme de matrice.

Projet 5 : Moteur de jeu de plateau

Objectif : Créer un programme qui simule un jeu de plateau avec des règles simples (ex : jeu de l'oie).

Compétences couvertes :

- Structures de contrôle (`if`, `while`, `switch`).
- Boucles pour simuler les tours de jeu.
- Manipulation de tableaux pour les positions des joueurs.

Fonctionnalités :

1. Créer un plateau avec n cases.
2. Simuler le mouvement des joueurs (tirage de dés).
3. Implémenter des cases spéciales (gagne un tour, perd un tour, etc.).
4. Afficher la position des joueurs après chaque tour.

Projet 6 : Jeu du pendu

Objectif : Implémenter un jeu du pendu avec des mots choisis aléatoirement.

Compétences couvertes :

- Manipulation de chaînes de caractères et de tableaux.
- Conditions et boucles (`for`, `while`).
- Gestion des erreurs (limiter le nombre d'essais).

Fonctionnalités :

1. Afficher le mot à deviner sous forme de tirets.
2. Demander à l'utilisateur de saisir une lettre à chaque tour.
3. Mettre à jour le mot affiché si la lettre est correcte, sinon incrémenter le nombre d'erreurs.
4. Afficher un message de victoire ou de défaite.