

# مشروع مادة هياكل البيانات

## توصيف مشروع إدارة المهام

### فكرة المشروع:

مشروع إدارة المهام يهدف إلى مساعدة المستخدمين على تنظيم وإدارة مهامهم اليومية بطريقة بسيطة حيث يوفر البرنامج واجهة تفاعلية تسهل على المستخدم إضافة المهام، عرضها، فرزها حسب الأولوية أو التاريخ، إكمال المهام، إدارة المهام العاجلة، وعرض المهام المكتملة.

يعتمد المشروع على استخدام هياكل البيانات المختلفة مثل المصفوفات، القوائم المرتبطة (Linked Lists)، وقائمة الانتظار (Queue) لتحقيق إدارة فعالة وسهلة للمهام.

### المهام التي يقوم بها المشروع:

1. إضافة مهمة جديدة.
2. عرض جميع المهام.
3. حذف مهمة.
4. فرز المهام حسب الأولوية.
5. فرز المهام حسب التاريخ.
6. إكمال مهمة (أي نقل المهمة من مهمة موجودة في قائمة المهام إلى قائمة المهام التي تم تنفيذها).
7. عرض المهام المكتملة (أي عرض المهام الموجودة في قائمة المهام التي تم تنفيذها).
8. إضافة مهمة عاجلة. (اختياري)
9. عرض المهام العاجلة. (اختياري)

### آلية عمل البرنامج بالتفصيل:

1. إضافة مهمة جديدة:
- الوصف: يسمح للمستخدم بإضافة مهمة جديدة عبر إدخال اسم المهمة، الأولوية (1-عالية، 2-متوسطة، 3-منخفضة)، وتاريخ الإضافة.
  - البنية المستخدمة:
- يتم تخزين المهام في مصفوفة ثابتة `tasks[]` تُستخدم لتخزين البيانات المرتبطة بكل مهمة.

## • الخطوات :

1. إدخال تفاصيل المهمة.
  2. تخزين المهمة في المصفوفة.
  3. زيادة عداد المهام taskCount.
- 

## 2. عرض جميع المهام:

- الوصف: يعرض البرنامج جميع المهام النشطة المخزنة في المصفوفة.
  - البنية المستخدمة :
- يعتمد على قراءة بيانات المهام المخزنة في المصفوفة tasks[]

## • الخطوات :

1. قراءة جميع المهام في المصفوفة حتى آخر مهمة مضافة.
  2. طباعة اسم المهمة، الأولوية، والتاريخ.
- 

## 3. حذف مهمة:

- الوصف: يسمح للمستخدم بحذف مهمة معينة بناءً على رقمها.
  - البنية المستخدمة :
- يتم تعديل المصفوفة tasks[] لسد الفراغ الناتج عن الحذف.

## • الخطوات :

1. إدخال رقم المهمة المراد حذفها.
  2. حذف المهمة.
  3. نقل باقي المهام للأمام لسد الفراغ.
- 

## 4. فرز المهام حسب الأولوية:

- الوصف: يتم ترتيب المهام حسب الأولوية باستخدام خوارزمية الفرز الفقاعي (Bubble Sort)

- البنية المستخدمة :

- المصفوفة `tasks[]` لتخزين المهام.
- خوارزمية Bubble Sort لترتيب المهام.

- الخطوات :

1. مقارنة أولويات المهام.
2. تبديل المواقع بين العناصر إذا لزم الأمر.
3. طباعة المهام بالترتيب الجديد.

---

## 5. فرز المهام حسب التاريخ:

- الوصف : يتم ترتيب المهام حسب تاريخ الإضافة باستخدام خوارزمية الفرز السريع (Quick Sort)

- البنية المستخدمة :

- المصفوفة `tasks[]`.
- خوارزمية Quick Sort لترتيب التواريخ.

- الخطوات :

1. اختيار عنصر محوري (Pivot).
2. تقسيم المصفوفة إلى قسمين (تواريخ أقدم وأحدث).
3. فرز كل قسم بشكل متكرر.
4. دمج النتائج النهائية.

---

## 6. إكمال مهمة:

- الوصف : يتم نقل المهمة المكتملة من قائمة المهام النشطة إلى قائمة المهام المكتملة.

- البنية المستخدمة :

- يتم حذف المهمة من المصفوفة `tasks[]`.
- يتم إضافة المهمة إلى قائمة مرتبطة `CompletedTask` (Linked List)

## • الخطوات :

1. تحديد المهمة المكتملة.
  2. إزالتها من المصفوفة.
  3. إنشاء عقدة جديدة وإضافتها إلى القائمة المرتبطة.
- 

## 7. عرض المهام المكتملة:

- الوصف :يعرض البرنامج جميع المهام المكتملة المخزنة في القائمة المرتبطة.
- البنية المستخدمة :

◦ القائمة المرتبطة. CompletedTask (Linked List)

## • الخطوات :

1. قراءة العقد من القائمة المرتبطة بدءًا من الرأس. completedHead.
  2. طباعة تفاصيل كل مهمة.
- 

## 8. إضافة مهمة عاجلة:

- الوصف :يتم تخصيص قائمة منفصلة للمهام العاجلة لضمان معالجتها بالترتيب الذي أضيفت به.

## • البنية المستخدمة :

◦ قائمة الانتظار urgentTasks (Queue) لتخزين المهام العاجلة.

## • الخطوات :

1. إدخال تفاصيل المهمة العاجلة.
  2. إضافة المهمة إلى نهاية قائمة الانتظار باستخدام Enqueue
- 

## 9. عرض المهام العاجلة:

- الوصف :يعرض البرنامج قائمة المهام العاجلة المخزنة في قائمة الانتظار.
- البنية المستخدمة :

○ قراءة المهام من قائمة الانتظار. urgentTasks.

#### • الخطوات :

1. قراءة جميع المهام المخزنة في قائمة الانتظار.
2. طباعة تفاصيل كل مهمة حسب ترتيب الإدخال.

---

هياكل البيانات المستخدمة في المشروع:

#### 1. المصفوفة: (Array)

- لتخزين المهام النشطة.
- توفر الوصول السريع للبيانات باستخدام المؤشرات.

#### 2. القائمة المرتبطة: (Linked List)

- لتخزين المهام المكتملة بطريقة مرنة وديناميكية.

#### 3. قائمة الانتظار: (Queue)

- لإدارة المهام العاجلة وفق ترتيب الإدخال (FIFO).

#### 4. خوارزميات الفرز :

- **Bubble Sort**: لفرز المهام حسب الأولوية.
- **Quick Sort**: لفرز المهام حسب التاريخ.

---

إضافات ممكنة لتحسين المشروع:

#### 1. التراجع عن آخر عملية: (Undo Last Operation)

الوصف:

تسمح ميزة التراجع للمستخدم بإلغاء آخر عملية تمت على قائمة المهام العادية (مثل إضافة مهمة، حذف مهمة.. إلخ). يتم ذلك عن طريق تسجيل العمليات في مكس (Stack)، ثم عند تنفيذ التراجع يتم استرجاع العملية الأخيرة وعكس تأثيرها.

آلية التنفيذ:

#### 1. استخدام مكس: (Stack)

- يتم تخزين تفاصيل كل عملية (مثل الإضافة، الحذف، أو التعديل) في المكس operations فور تنفيذها.
- يتم تسجيل نوع العملية مع بيانات المهمة (مثل الاسم، الأولوية، والتاريخ) داخل المكس.

## 2. تنفيذ التراجع:

- عند طلب التراجع، يتم استخراج آخر عملية من المكس باستخدام Pop()
- يتم تحليل نوع العملية (إضافة، حذف) واسترجاع الحالة السابقة.
- يتم تنفيذ العكس:
  - لعملية الإضافة: يتم حذف المهمة المضافة.
  - لعملية الحذف: يتم إعادة المهمة المحذوفة.

---

## تسليم المشروع

السبت 2025/5/17

المقابلات تُجرى كل طالب حسب توقيته محاضراته في الأسبوع الذي يلي السبت 2025/5/17.

أي المقابلات ستكون يوم الأحد 2025/5/18

والثلاثاء 2025/5/20.

مع التمنيات بالتوفيق.

م. بيان سقى