# 数字

**标题：闹钟说明：*说明：***

创建一个使用标签显示当前时间的程序。允许用户选择给定时间并指定消息。当时钟到达该时间时，让它弹出一条消息提醒用户他们的消息和/或播放特定的声音文件。

***提示：***

您需要能够获取当前时间并将其显示给用户。您还需要一种机制来保存具有特定时间和关联消息的“事件”。尝试创建一个自定义类来保存此信息。在时钟的每个滴答声中，将时间与每个保存的计划事件进行比较。如果其中一条匹配，则显示关联的消息。确保在给定的时间内检查多个事件，如果不需要，可能会丢弃系统中的任何旧事件（过期）。

***增加的困难：***

使用数字图片使时钟显示为图形。您还可以集成一个数据库，以便更轻松地跟踪多个事件。在消息中添加一个图标，并可能指定播放哪个声音文件作为警报。提醒用户事件是否“过期”以及过期时间。

# 文本

**标题：文本编辑器说明：*说明：***

制作一个简单的记事本样式编辑器，用户可以使用它创建、打开和编辑简单的基于文本的文件。提供典型编辑器中的功能，包括保存、复制/粘贴和打印功能。

***提示：***

从简单的事情做起，一步步向上。创建应用程序首先能够打开一个窗口，允许用户在其中键入并保存。此窗口可能包含多行文本框或文本区域（如果您正在使用web应用程序）、工具栏和状态栏。考虑一下如何将文件读入这个新窗口并将其写回。一旦你完成了读入和写出的基础知识，你就可以进行保存和打开对话框，并从那里复制/粘贴和打印。想一想如果有人试图打开一个非文本文件，你的编辑器会有什么反应。

***增加的困难：***

附加格式选项，如粗体、斜体和下划线。您还可以添加一个选项卡式界面，允许用户同时打开多个文本文档并单独编辑它们。

# 班级

**职位：银行账户经理**

***说明：***

设计一个充当ATM机的程序。用户可以指定PIN，并向他们显示帐户类型（支票和储蓄）的菜单。允许用户从所选帐户类型存款或取款。一定要检查他们不能提取超过自己拥有的钱，或者如果他们存款超过1万美元，它会让用户知道必须联系银行经理结清存款，然后不会让他们提取任何钱。

***提示：***

这是类继承的一个经典示例。创建一个Account类，作为这两种类型帐户的基础。支票账户和储蓄账户都是对的吗？它们将从基本Account类继承。在基类中保留基本功能，如存款（）和取款（），然后在特定帐户类型中重写它们。储蓄账户也可能有利息或提款罚款等内容。重写这些方法时要考虑到这一点。变量“balance”也是基类的受保护成员。

***增加的困难：***

允许支票账户转为负数，并收取透支费

10美元。不要让用户进入超过100美元的透支保护。如果用户已经透支，您还可以防止用户多次提款。

**标题：处理大数字的类说明：*说明：***

创建一个可以处理大整数值的类。用户应该能够输入一个非常大的数字（比如我们银河系或已知宇宙中的恒星数量），然后该类应该能够存储该数字，并允许该类的两个实例相互减法、加法、乘法或除法。其中一个操作的结果是large numbers类的另一个实例。

***提示：***

一种方法是把一个大整数看成一系列数字，你会考虑一系列字符串。然后，您可以在执行各种操作时分别处理这些数字。首先尝试使用减法或加法，因为它们应该是最容易实现的。

***增加的困难：***

支持其他一些运算符，例如模数，或者查看是否可以通过“ToString”方法对数字应用各种格式。

**标题：购物车说明：*说明：***

设计并实现一个购物车系统，允许用户输入各种产品，并将其展示给商店中的潜在买家。买家可以选择一个项目，把它放在他们的购物车，并通过结帐计算小计，税和最终购买总额。

***提示：***

这里我们有另一个产品作为对象的实例。购物车也将是一个保存这些产品对象的对象。要计算小计，您需要在购物车中循环，将总和相加（通过从product对象获取价格并将其乘以买方想要的数量），根据此总和计算税费，并将其添加到小计中以找到订单总额。根据您用于实施此项目的语言/媒介，您可能必须跟踪会话，以便在将商品放入购物车后，用户可以继续购物而不会丢失这些商品。然后，当他们准备购买时，购物车将包含所有产品。确保购物车具有移除物品和更改给定物品数量的选项。

***增加的困难：***

在会话之间（打开/关闭程序或访问网站之间）保存购物车的内容。附加捆绑，其中两个项目可以以折扣价格捆绑在一起。这里bundles可能是另一种对象类型，它也具有折扣属性，并包含捆绑项目的列表。

# 图形和图形用户界面

**标题：海龟图形说明：*说明：***

制作一个应用程序，指示屏幕上的海龟图标根据用户输入绘制各种形状。例如，如果用户发出“放下笔”命令，海龟将开始一行。如果用户随后发出命令“移动到1,1”，它将从当前位置到坐标1,1绘制该线。。。这就形成了一条线。然后，用户可以告诉海龟“举起笔”，在那里它将结束线。程序应该允许用户指导海龟如何画线，移动到不同的坐标，放下或举起笔。

***提示：***

如果您希望学习如何接收用户指令，解析它们，然后将它们转换为屏幕上的海龟所做的操作，那么这种类型的程序非常棒。因此，第一部分是为程序可以解析的命令提供语法。例如，可能命令是“DROP”，对象是“PEN”，在这种情况下，您可以告诉海龟放下笔。如果用户输入“DROP MARKER”，它会看到“DROP”并理解它，但不会理解“MARKER”，因此会发出错误。一旦你有了一个可以解析各种命令的函数，剩下的就是指导海龟做什么。建议您创建可以调用以控制海龟的各种函数。您甚至可以创建一个海龟类，并使用各种方法来控制它。

***增加的困难：***

让海龟用一个命令从当前位置画一颗星星。