GRASP Principles in ParkingApp

ATM: This superclass would be a Creator, since it creates static variables that its child classes act upon in the methods for which they are responsible. Its child classes manipulate data from the variables it creates.

CheckInATM: This could also easily fall under the Creator principle. Not only does the CheckInATM have a method to pull information from the FileInput ticketFile, but it also has the createTicket(int inTime) method through which it creates all of the Ticket instances used by itself as well as the other CheckOutATM (made possible by their superclass ATM). It may also fall under Information Expert, as both ATMs are responsible for the actions that don't fall under the Main driver class, especially regarding the manipulation of Tickets.

CheckOutATM: This would primarily be the Information Expert, as stated above. With the information residing in its parent ATM class, its methods act upon the Ticket instances to allow the program to proceed as desired.

FileInput and FileOutput: These classes are good examples of Low Coupling, because they need to have no knowledge of any other classes within this program. Their constructors and methods are utilized by others in their own methods, but these two classes have no need for the information outside classes might have.

Ticket: The Ticket class is a perfect example of Information Expert, not for the fact that it does something with that information, but because it knows the information necessary for others to act upon. It knows the specific attributes of each Ticket, and provides it to the ATM classes requesting the information.