Java Coding Question: Employee Management System

Problem Statement:

You are tasked with creating an Employee Management System for a company. The system should manage different types of employees, such as Full-Time and Part-Time employees. Each type of employee will have different attributes and behaviors, but there will also be some common features shared among all employees.

Requirements:

1. **Abstraction:**
   - Create an abstract class `Employee` that will serve as the base class for all employee types. This class should have the following abstract methods:
     - `calculateSalary()`: This method should calculate the salary of the employee. The implementation will differ based on the type of employee.
     - `getEmployeeDetails()`: This method should return a string containing the employee's details, including name, ID, and type of employee.
   - The `Employee` class should also have some common attributes, such as:
     - `String name`
     - `int employeeId`
   - Implement appropriate constructors, getters, and setters using encapsulation.
2. **Inheritance:**
   - Create two subclasses `FullTimeEmployee` and `PartTimeEmployee` that extend the `Employee` class.
   - The `FullTimeEmployee` class should have an additional attribute `double annualSalary`.
   - The `PartTimeEmployee` class should have an additional attribute `double hourlyRate` and `int hoursWorked`.
   - Override the `calculateSalary()` and `getEmployeeDetails()` methods in each subclass to provide specific implementations for each type of employee.
3. **Encapsulation:**
   - Ensure that all attributes in the `Employee` class and its subclasses are private and accessible only through public getter and setter methods.
4. **Polymorphism:**
   - In the main method of your program, create an array of `Employee` objects and populate it with different types of employees (both full-time and part-time).
   - Loop through the array and print the details of each employee using the `getEmployeeDetails()` method.
   - Calculate and display the total salary for all employees using the `calculateSalary()` method.

Example Output:

```java
Employee Details:
Name: John Doe, ID: 101, Type: Full-Time, Salary: 50000.0
Name: Jane Smith, ID: 102, Type: Part-Time, Salary: 2000.0

Total Salary: 52000.0
```

Bonus:

- Add a method `displayEmployeeType()` in the `Employee` class that prints the type of employee (Full-Time or Part-Time) using runtime polymorphism.