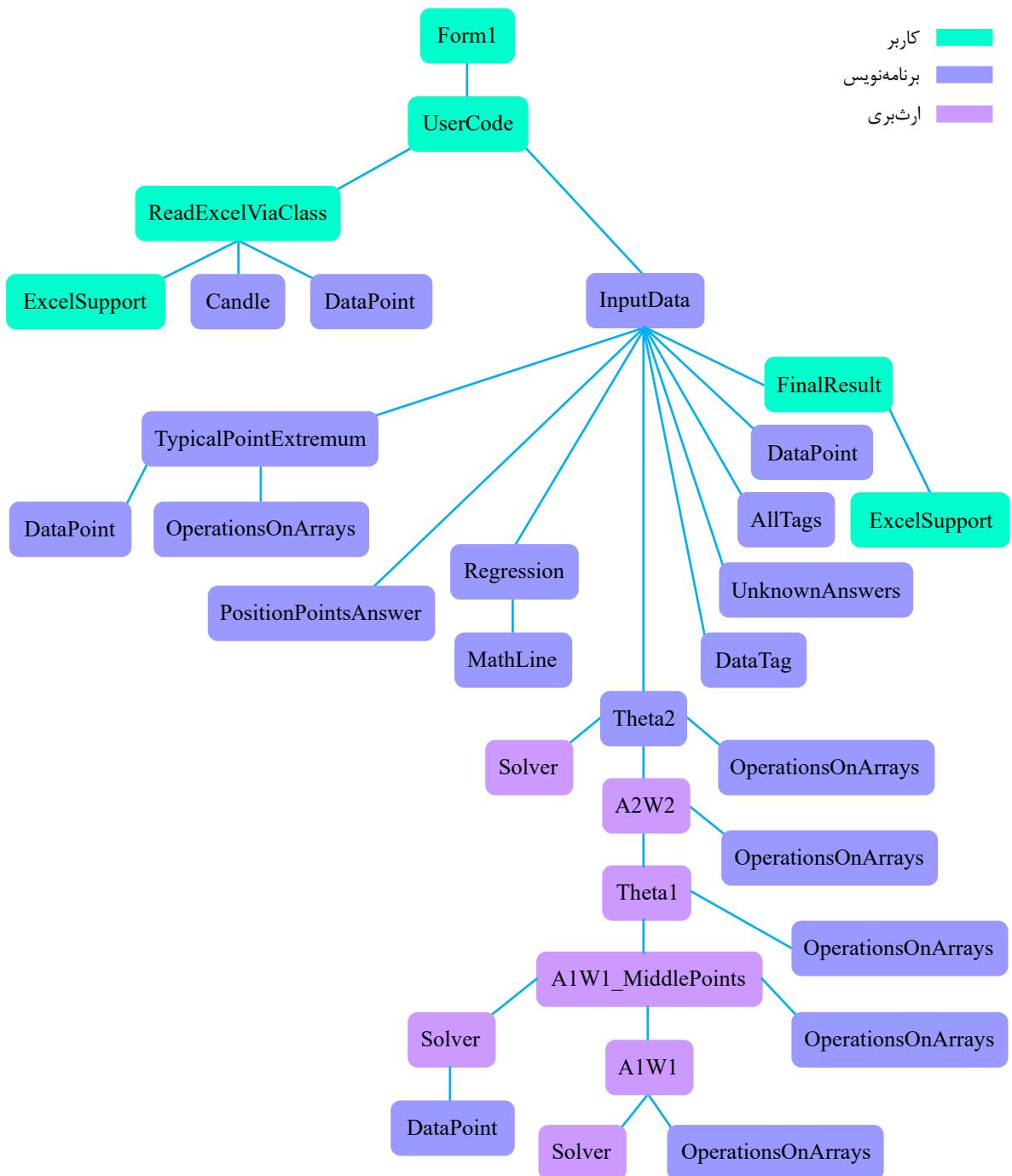


گراف برنامه داتنت به صورت زیر است:



توضیح هر کلاس و متدهای آن:

در این کلاس اطلاعات مورد نیاز از کاربر گرفته می‌شود و توسط کاربر این فرم ایجاد می‌شود. کاربر از متدهای **Receive_Result** برای بکارگیری کلاس **UserCode** استفاده می‌کند.

کلاسی است که توسط کاربر جهت دادن هر نوع اطلاعاتی به برنامهنویس طراحی می‌شود. از آنجایی که کاربر از هر نوع فایلی می‌تواند استفاده کند پس مجبور به تعریف یک سری کلاس‌های دیگر می‌شود از جمله کلاس **ExcelSupport**.

UserCode دو کلاس نام برده از طریق متدهای **ComputePoint** در کلاس **ExcelSupport** و **ReadExcelViaClass** بکار گرفته می‌شوند.

: این کلاس برای خواندن و نوشتن سلول به سلول یک اکسل مورد استفاده قرار می‌گیرد. از متدهای **Get_One_Cell** برای خواندن هر سلول اکسل استفاده می‌شود و از متدهای **Set_One_Cell** برای نوشتن در هر سلول اکسل استفاده می‌گردد.

: این کلاس برای ذخیره اطلاعات فایل اکسل خوانده شده به کمک کلاس **ExcelSupport** به صورت لیستی از دو کلاس **DataPoint** و **Candle** است. از متدهای **ComputeAllPoints2** در این کلاس برای ایجاد دو لیست از کلاس **DataPoint** و **Candle** استفاده می‌کنیم.

: این کلاس، موجودیت هر کندل را تعریف می‌کند که شامل اطلاعاتی مانند زمان پیوسته، قیمت باز شدن، بالاترین قیمت، پایین‌ترین قیمت، قیمت بسته شدن، حجم در هر کندل، اندیس (زمان گستته) است.

: این کلاس، موجودیت هر نقطه را با constructor-های متفاوتی، متناسب با نیاز برنامه، تعریف می‌کند. پس از اینکه لیستی از اطلاعات جهت ارائه به برنامه‌نویس آماده گردید، این اطلاعات با فراخوانی کلاس اصلی (از طریق ایجاد شی از آن)، به برنامه‌نویس تحويل داده می‌شود. برنامه اصلی، **InputData** است که برای اجرا این کلاس، ابتدا ورودی به کلاس داده می‌شود و سپس متدهای **Compute** برای اجرای برنامه از طریق یک شی از این کلاس اجرا می‌شود.

: این کلاس، اصلی‌ترین کلاسی است که توسط برنامه‌نویس به کاربر ارائه می‌شود. در این کلاس از کلاس‌ها و متدهای ذیل استفاده می‌شود: **Theta2**, **Regression**, **PositionPointsAnswer**, **TypicalPointExtremum**, **Tags**, **CONT_R**, **FinalResult**, **DataPoint**, **AllTags**, **UnknownAnswers**, **DataTag**, **Compute** و **StabilityCoefficient_DSP**.

کلاس‌های **DataTag** و **Theta2**, **Regression**, **TypicalPointExtremum** و کلاس‌هایی جهت انجام محاسبات هستند و کلاس‌های **DataPoint** و **AllTags**, **UnknownAnswers**, **PositionPointsAnswer** برای ذخیره محاسبات به صورت یک لیست جهت تحويل دادن به کاربر هستند.

: برای شمارش تعداد اعضای مخالف صفر یک لیستی از داده‌های **Int** است.

: در این متدهای هر 6 نقطه اکسترم متوالی، اطلاعات زیر کسب می‌شود:

- شماره نقاط اکسترم درستگاه اول که خط روند از آن دو نقطه رسم شده است.
- تعداد نقاط اکسترمی که در هر دستگاه در بالا یا پایین خط روند رسم شده، قرار گرفته است.
- خط روند رسم شده در هر دستگاه، صعودی است یا نزولی؟
- خط روند رسم شده در هر دستگاه، اگر نزولی است عدد 1 و اگر صعودی است عدد 2 را چاپ کن.
- اگر تمامی نقاط اکسترم در هر دستگاه در پایین خط روند رسم شده قرار بگیرد down خواهد بود و اگر تمامی نقاط اکسترم در بالای خط روند رسم شده قرار بگیرند up خواهد بود.
- اگر تمامی نقاط اکسترم در هر دستگاه در پایین خط روند رسم شده قرار بگیرد down خواهد بود و عدد 1 را چاپ کن. اگر تمامی نقاط اکسترم در بالای خط روند رسم شده قرار بگیرند up خواهد بود و عدد 2 را چاپ کن.
- نقاط اکسترم در هر دستگاه که خط روند از آن رسم شده است ماکریم هستند یا مینیمیم؟

- ($\text{abs}(\text{w_max}) - \text{abs}(\text{w_min})$) / ($\text{abs}(\text{w_max}) + \text{abs}(\text{w_min})$) * 200 ○
- اگر رابطه ($\text{abs}(\text{w_max}) - \text{abs}(\text{w_min})$) / ($\text{abs}(\text{w_max}) + \text{abs}(\text{w_min})$) * 200 برای هر دستگاه، بزرگتر و مساوی 50 بود عدد 1 را چاپ کن و در غیر این صورت عدد 0 را چاپ کن.

در این متدهای **DSP**، ضریب پایداری قیمت، ضریب پایداری زمانی و نسبت این دو ضریب پایداری، محاسبه می شود.

Compute: محاسبات اصلی کلاس **InputData** در این متدهای انجام می شود که کاربر باید با استفاده از نمونه شی کلاس **InputData**، این متدهای ذیل را جهت اجرای برنامه فراخوانی کند.

TypicalPointExtremum: این کلاس برای محاسبه نقاط اکسترمم در n تعداد داده است. در این کلاس، از کلاس ها و متدهای ذیل استفاده می شود: **EMA**, **CONT_2**, **CONT**, **OperationsOnArrays**, **DataPoint** و **Compute**. در کلاس اصلی **InputData**، یک نمونه شی از کلاس **TypicalPointExtremum** ساخته می شود و سپس متدهای **EMA** فراخوانی می شود.

CONT: این متدهای شمارش تعداد علامت های مثبت و منفی ستون های هیستوگرام مکدی در هر فاز مثبت و منفی است.

CONT_2: این متدهای شمارش تعداد ستون های هیستوگرام مکدی در آخرین فاز مثبت یا منفی که در انتهای داده اتفاق می افتد، است.

EMA: در این متدهای شمارش میانگین متحرک 12 و 26 روز و مکدی برای تمامی کندهای محاسبه می شوند. نقاط اکسترمم تعیین می شوند. اولین کندهای که معین می کند در داده های گذشته یک اکسترمم وجود دارد نیز تعیین می شود.

این کلاس برای عملیات روی آرایه و لیست ایجاد شده است و شامل متدهای زیر است:

- برای افزایش بکار می رود: **LineSpace** ○
- پیدا کردن اندیس یک عضو مشخصی از آرایه: **FindIndexArray** ○
- پیدا کردن اندیس یک عضو مشخصی از لیست: **FindIndexList** ○
- تبدیل یک لیستی از داده های **Int** به **String**: **ConvertIntListToString** ○
- تبدیل یک لیستی از داده های **Double** به **String**: **ConvertDoubleListToString** ○
- تبدیل یک لیستی از داده های **Int** به لیستی از **String**: **ConvertIntListToStringList** ○
- تبدیل یک لیستی از داده های **Double** به لیستی از **String**: **ConvertDoubleListToStringList** ○
- تبدیل یک آرایه ای از داده های **Double** به لیستی از **String**: **ConvertDoubleArrayListToStringList** ○
- تبدیل یک آرایه ای از داده های **Int** به لیستی از **String**: **ConvertIntArrayToStringList** ○

PositionPointsAnswer: موجودیت 7 نقطه اکسترمم متوالی که برای حل دو دستگاه متوالی استفاده می شود، تعریف می کند و این اطلاعات شامل موارد ذیل است: زمان پیوسته، زمان گسسته، قیمت در نقطه اکسترمم، زمان شکار که بیانگر وجود اکسترمم در اطلاعات قبلی است، فاصله تغییر فاز بعد تعیین اکسترمم اول با اولین زمان شکار جهت یافتن اکسترمم دوم، فاصله اکسترمم اول با اولین زمان شکار جهت یافتن اکسترمم دوم، زمان تغییر فاز بعد تعیین اکسترمم اول و جست و جو برای اکسترمم دوم.

این کلاس برای محاسبه ضرایب a_1 و a_0 در خط رگرسیون $r = a_1t + a_0$ است که محاسبه آن توسط متدهای **ComputeRegr** در این کلاس انجام می شود. در این کلاس از کلاس **MathLine** برای ذخیره اطلاعات استفاده می کنیم.

: این کلاس، موجودیت یک خط را بیان می‌کند که به دو صورت ضمئی و صریح، معادله خط تعریف می‌شود. در این کلاس موجودیت روند هر دستگاه مشخص می‌شود که متد **Tags** در کلاس **InputData** از نوع این کلاس است.

: این کلاس، موجودیت ضرایب مجھول معادله زیر را مشخص می‌کند که با اجرای برنامه، این ضرایب بدست می‌آیند و به کاربر تحویل داده می‌شوند:

$$y = a_1 \sin(w_1 t + \theta_1) + a_2 \sin(w_2 t + \theta_2).$$

ضرایب مجھول عبارتند از: $a_1, w_1, a_2, w_2, \theta_1$ و θ_2 .

: موجودیت تگ‌های مختلف را بیان می‌کند.

: این کلاس، نتایج حاصله از اجرای برنامه را به کمک کلاس **ExcelSupport** و ایجاد نمونه شی از این کلاس و فراخوانی متد **Set_One_Cell** از کلاس **ExcelSupport** در هر سلول اکسل نوشته و به کاربر تحویل می‌دهد.

: در این کلاس ضریب θ_2 محاسبه می‌شود. نکته‌ای که وجود دارد این است که محاسبات این ضرایب به هم وابسته هستند لذا به این دلیل بحث وراثت پیش می‌آید. ضرایب معادله فوق به ترتیب به صورت زیر بدست می‌آیند:

$$w_1 \text{ و } a_1 \quad (1)$$

$$\theta_1 \quad (2)$$

$$w_2 \text{ و } a_2 \quad (3)$$

$$\theta_2 \quad (4)$$

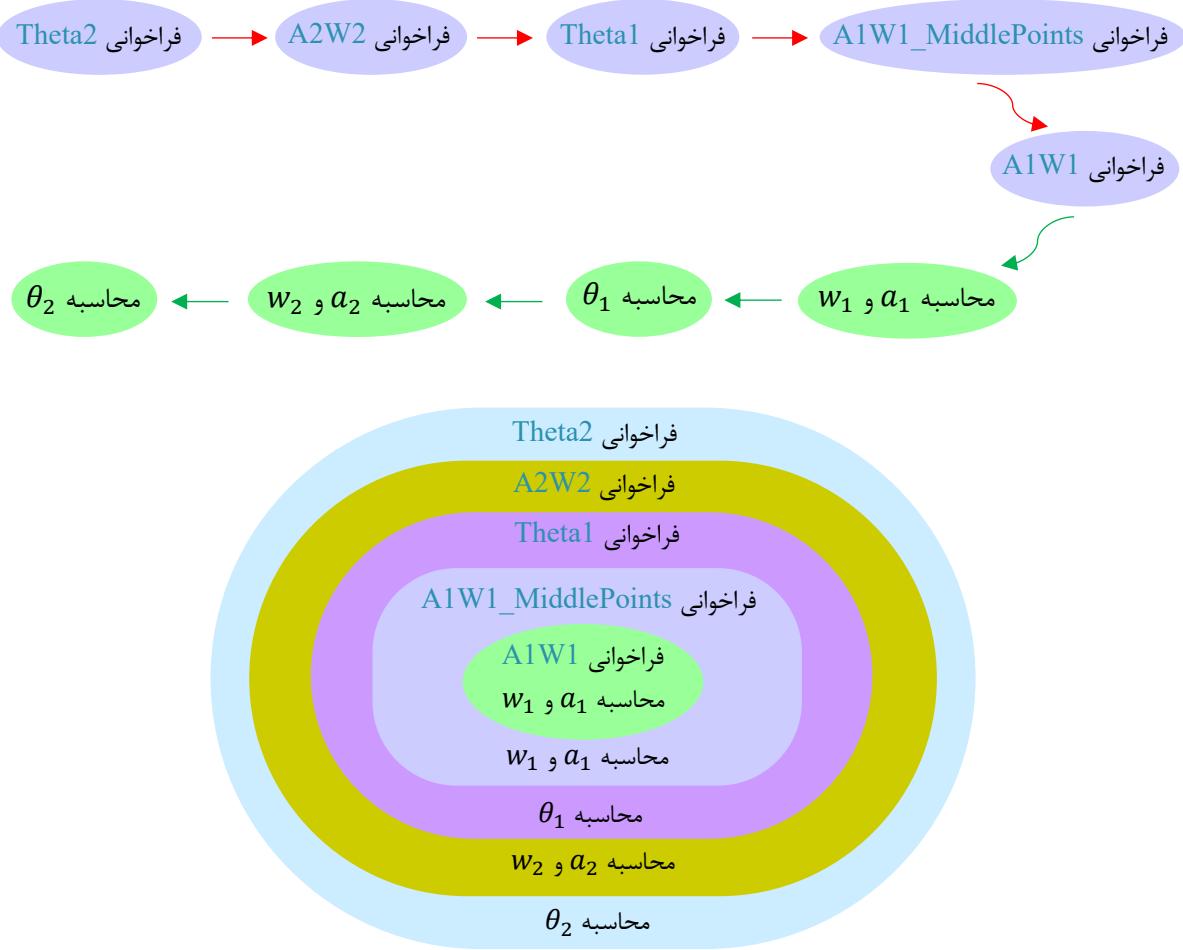
پس به کلاس‌هایی نیاز داریم که از هم ارثبری کنند. این ارثبری به صورت زیر است:

- 1) A1W1 : Solver
- 2) A1W1_MiddlePoints : A1W1
- 3) Theta1 : A1W1_MiddlePoints
- 4) A2W2 : Theta1
- 5) Theta2 : A2W2

ارثبری	کلاس پدر	کلاس فرزند
A1W1 : Solver	Solver	A1W1
A1W1_MiddlePoints : A1W1	A1W1	A1W1_MiddlePoints
Theta1 : A1W1_MiddlePoints	A1W1_MiddlePoints	Theta1
A2W2 : Theta1	Theta1	A2W2
Theta2 : A2W2	A2W2	Theta2

نکته: زمانی که بحث وراثت داریم، برای شی‌سازی، درونی‌ترین فرزند را صدای زنیم و برنامه وقتی اجرا می‌شود در ابتدا سراغ بیرونی‌ترین پدر می‌رود.

نحوه بکارگیری کلاس‌ها برای محاسبه ضرایب معادله فوق به صورت زیر است:



: این کلاس، یک کلاس پدر است و سایر کلاس‌های فرزند، ویژگی‌های این کلاس را ارث بری می‌کنند. در این کلاس، دو متده است `Solver` `ComputeRegressionLine` و `ComputeMiddlePoints` تعریف شده است.

: این متده برای محاسبه نقاط میانی هر 6 نقطه اکسترمم متوالی است. `ComputeMiddlePoints`

: این متده یک آرایه از نوع `double` خط رگرسیون با شیب a_1 و عرض از مبدأ a_0 ایجاد می‌کند (یعنی $r = a_1 t + a_0$)

: این کلاس، از کلاس `Solver` ارث بری می‌کند و خودش کلاس پدر برای کلاس `A1W1` است. چهار متده `CompA1W1`, `YF1`, `F1` و `F2` در این کلاس تعریف شده‌اند. متدهای `F1`, `F2` و `YF1`، تابع‌های مورد نیاز برای محاسبه a_1 و w_1 هستند. متده `CompA1W1` برای محاسبه a_1 و w_1 است. در این کلاس، از کلاس `OperationsOnArrays` برای افزایش‌بندی یک بازه و پیدا کردن اندیس یک عنصر مشخصی از لیست، استفاده می‌شود. چون متده `CompA1W1` توسط کلاس فرزند `A1W1` ارث بری می‌شود آن را به صورت `virtual` تعریف می‌کنیم.

: این کلاس از کلاس `A1W1` ارث بری می‌کند و خودش کلاس پدر برای کلاس `A1W1_MiddlePoints` است. در این کلاس، دو متده `CompA1W1` و `ComputeA1W1` تعریف می‌شوند که متده `CompA1W1` از کلاس `A1W1` ارث برده است به این دلیل آن را به صورت `override` تعریف کرده‌ایم. از این دو متده از کلاس `A1W1_MiddlePoints`

متدهای `ComputeA1W1` باید در کلاس فرزند `Theta1` با استفاده از دستور `base` فراخوانی شود. متدهای `CompA1W1` برای یافتن λ بهینه، مدام متدهای `F3` و `F5` نقطه میانی بکار می‌گیرد و اگر a_1 و w_1 بدست آمده مناسب نبود سراغ متدهای `CompA1W1` از کلاس پدر `A1W1` برای `F6` نقطه می‌رود.

در این کلاس، از کلاس `OperationsOnArrays` برای افزایش بندی یک بازه و پیدا کردن اندیس یک عنصر مشخصی از لیست، استفاده می‌شود.

این کلاس از کلاس `A1W1_MiddlePoints` ارث بری می‌کند و خودش کلاس پدر برای کلاس `A2W2` است. در این کلاس، دو متدهای `F3` و `F5` تعریف می‌شوند. `F3`،تابع مورد نیاز برای محاسبه θ_1 است. متدهای `F4` و `F6` تابع های مورد نیاز برای محاسبه θ_2 و w_2 هستند. متدهای `F3` و `F5` در این کلاس تعریف شده‌اند. متدهای `F4` و `F6` تابع های مورد نیاز برای محاسبه θ_2 و w_2 است. متدهای `F3` و `F5` برای محاسبه a_2 و w_2 است. متدهای `F4` و `F6` برای یافتن ضریب مناسب جهت تعیین بازه برای w_2 ، بارها اجرا می‌شود تا این ضریب بهینه پیدا شود. متدهای `CompA2W2` باید در کلاس `CompA2W2` فرزند `Theta2` با استفاده از دستور `base` فراخوانی شود.

در این کلاس، از کلاس `OperationsOnArrays` برای افزایش بندی یک بازه و پیدا کردن اندیس یک عنصر مشخصی از لیست، استفاده می‌شود.

این کلاس از کلاس `Theta1` ارث بری می‌کند و خودش کلاس پدر برای کلاس `A2W2` است. چهار متدهای `F3`، `F4`، `F5` و `F6` در این کلاس تعریف شده‌اند. متدهای `F3` و `F5` تابع های مورد نیاز برای محاسبه θ_1 و w_1 هستند. متدهای `F4` و `F6` تابع های مورد نیاز برای محاسبه θ_2 و w_2 هستند. متدهای `F3` و `F5` برای محاسبه a_1 و w_1 است. متدهای `F4` و `F6` برای یافتن ضریب مناسب جهت تعیین بازه برای w_1 ، بارها اجرا می‌شود تا این ضریب بهینه پیدا شود. متدهای `CompA2W2` باید در کلاس `CompA2W2` فرزند `Theta2` با استفاده از دستور `base` فراخوانی شود.

در این کلاس، از کلاس `OperationsOnArrays` برای افزایش بندی یک بازه و پیدا کردن اندیس یک عنصر مشخصی از لیست، استفاده می‌شود.

این کلاس از کلاس `A2W2` ارث بری می‌کند. در این کلاس، دو متدهای `F3` و `F5` تعریف می‌شوند. `F3` تابع مورد نیاز برای محاسبه θ_2 است. متدهای `F3` و `F5` برای یافتن آن را به صورت `override` تعریف می‌کنیم.

در این کلاس، از کلاس `OperationsOnArrays` برای افزایش بندی یک بازه و پیدا کردن اندیس یک عنصر مشخصی از لیست، استفاده می‌شود.

الگوریتم برنامه به صورت زیر است:

1. تحويل اطلاعات از کاربر به برنامه نویس.
2. تعیین نقاط اکسترمم.
3. تعیین شیب و عرض از مبدأ خط رگرسیون.
4. محاسبه ضرایب معادله.
5. پیاده‌سازی تگ‌ها روی دستگاه معادلات.