* **Ampelphasen**

1. Das grüne Licht ist Immer an und
2. Das Gelb licht ist an und der Timer ist an für 2 Sekonden
3. Das Rot licht ist an und der Timer ist an für 6 Sekonden
4. Die Rot und Gelb lichter sind an für 2 Sekonden

* **Schritt 0**
  + der Struktur (der plan) das Programm schreiben
  + Zeichnen von drei Farben als Bilder für die Ampel (Rot, Gelb, Grün) und ein Ampelmann (Rot, Grün)
  + Hintergrundbild Hinzufügen (Ampel und Straße)
  + Ordne die Bilder in der richtigen Reihenfolge an
  + Ein Timer und eine Taste (Stopptest) hinzufügen
  + Mit (if) die Bilderreihenfolge steuern
* **Schritt 1**
* Ein "Picturebox" hinzufügen, um die Bilder der Ampel aufzunehmen, und eine Schaltfläche, um die Phasen zu starten.
* erstellen ein "Enum" enthält die Phasen der Ampel „Grün = Go, Gelb = Attention, Rot = Stop, Rot und Gelb = Prepare“
* Phase "GO" ist immer An mit die Farbe Hellgrün und Dunkel Rot und Gelb, wann die Taste gedrückt wird der Timer starte wann der Timer tick, aktiviert die nächste Phase.
* **Schritt 2**
* eine neue Klasse „Lamp“ erstellen, die den Status des Lichts steuert, wenn es ein- oder ausgeschaltet ist, indem Sie die Bilder zwischen dunkler und heller ändern. Die Klasse Steuert die "Picture Boxes". Auf diese Weise ändert die Klasse „TrafficLight“ nicht die Farbe, sondern den Zustand der Lampe in Abhängigkeit von der Phase, in der sich die "Ampel" befindet.
* Die Klasse "Lampe" hat "Enum / On, Off", der "Enum" Kontrolliert den Zustand der Lampe. Mit der Option " Enum " kann die Klasse „TrafficLight“ überprüfen, in welchem ​​Status sich die Lampe befindet, und dann das Bild ändern.
* Die Klasse "Lampe" enthält eine Methode "SetState" mit der Variablen "requestState". Diese Methode kann vom Hauptprogramm aufgerufen werden, um dem Licht den Status "ON, OFF" zu geben. Dadurch kann das Image nach Bedarf geändert werden.
* **Schritt 3**
* neue Klasse "TrafficLight" hinzufügen Enthält die Struktur der „Ampel „die Schnittstelle und die Logik und erbt "User Control". Diese Klasse kann als "Tool" in der "Tool Box" verwendet werden, um sie zum Hauptformular hinzuzufügen. Dies können viele Elemente derselben Struktur enthalten, ohne dass eine vollständig andere Klasse geschrieben werden muss.
* das Formular "MainForm" Initialisiert nur die Klasse "TrafficLight".
* die Klasse "Lamp" die Eigenschaft "CurrentPhase" zuweisen, um zwischen den Farben der einzelnen "PictureBox" zu wechseln. Dies bedeutet, dass der Timer nur für den Phasenwechsel zuständig ist.
* Jetzt kann die Klasse "TrafficLight" den Zustand der Lampe in Abhängigkeit von der aktuellen Phase einstellen. Jede Lampe verfügt über eine einzelne Codezeile. Steuert die gewünschte Farbe, zu welcher Phase die Lampe gehört.
* **Schritt 4**
* Der "Timer\_Tick" ist nun für den Countdown verantwortlich, indem das "Label" jede Sekunde aktualisiert wird. und die Dauer des Countdowns aus dem Phasenwert in der "Enum". Der "Timer-Tick" ruft die Methode "GoToNextPhase ()" auf, wenn der Countdown "0" ist.
* Die Methode "GoToNextPhase ()" ist dafür verantwortlich die Phase jedes Mal zu ändern, wenn der Timer den Countdown beendet. Mit "Switch" ändert die Methode die Phase abhängig von der aktuellen Phase.
* Die Variable " remainingTime" erbt den Wert der aktuellen Phase und ist für den Countdown verantwortlich (indem jede Sekunde eine Zahl verringern).
* **Schritt 5**
* neue Klasse "TrafficPhase" erstellen.
* Die neue Klasse enthält die Eigenschaft "Type", die vom Enum "Phase Type" des Hauptformulars abgeleitet ist, um den Typ der Phase "Go, Attention, Stop, Prepare" festzulegen. Enthält auch einen Konstruktor, der für das Festlegen des Typs und der Zeitdauer für jede Phase verantwortlich ist, indem die Werte aus der Warteschlange übernommen werden. Und die Klasse enthält eine Methode, deren Hauptaufgabe es ist, jede Phase zu blockieren, bis die gewünschte Zeit abgelaufen ist.
* Die generische Klasse *Queue* als Warteschlange für diese Klasse verwenden, um die FiFo-Reihenfolge der Phasen zu steuern. Die generische Warteschlange kann zum Speichern von Objekten mit Typ und Wert verwendet werden. Und dann steuern, in welcher Reihenfolge die Objekte verwendet werden
* **Schritt 6**
* Die Klasse "TrafficPhase" behandelt jede Phase separat, indem sie den Typ und die Zeitdauer der Phase einstellt und den Timer startet und stoppt, abhängig vom Wert der Phase, wenn der Timer endet.
* Die Klasse "TrafficLight" ist dafür verantwortlich, die Phasen zur Warteschlange hinzufügen und die Reihenfolge zu steuern, in der sie ausgeführt und die Klasse "TrafficPhase" aufgerufen werden.
* **Schritt 7**
* Neues Label hinzufügen, um den Countdown anzuzeigen.
* Ändert das Zeitintervall auf eine Sekunde, ist der Timer jetzt nicht mehr für die Steuerung der Zeit der Phase verantwortlich.
* eine neue Klasse " RemainingTimeEventArgs" zuweisen, um den verbleibenden Zeitwert zu tragen, so dass er in "Ampel" verwendet werden kann, ohne die "remainingTime"variable aufzurufen.
* Der Timer tickt jede Sekunde und prüft, ob der Wert von " remainingTime " Null ist, wenn Null erreicht wird. Die Phase kann in die nächste Phase geändert werden. Jedes Mal, wenn eine Sekunde vergeht, ändert sich das Etikett auf den gleichen Wert der verbleibenden Zeit.
* **Schritt 8**
* neue Ampel anlegen "PedestrianLight" enthält rotes und grünes Licht. Und fügen Sie zwei Fußgängerampeln und eine Ampel hinzu, jetzt gibt es vier Ampelanlagen.
* neue Klasse "Phasenregler" anlegen Diese Klasse ist für den gesamten Ablauf der Ampel verantwortlich, sie steuert das Hinzufügen der Phasen zu einer Warteschlange und initialisiert die Sequenz. In diesem Fall weiß die Ampel nur, was die aktuelle Phase ist.

die Steuerung ist die einzige, die ein Signal von der Fußgängerampel empfängt, um die Sequenz zu starten, wenn die Schaltfläche angeklickt wurde.

die Ampel empfängt die aktuelle Phaseninformation über den Eventhandler "PhaseChanged", erfasst sie die Information jedes Mal, wenn sich die Phase geändert hat, und leitet sie an die Ampel weiter.

* **Schritt 9**
* zwei neue Ampeln hinzufügen und die Fußgängerampeln löschen. Die Ampeln als Haupt- und Neben Ampel Zuweisen und eine Schaltfläche zu den Nebenampeln hinzufügen. der Ampel eine Eigenschaft hinzufügen, damit die Sichtbarkeit der Schaltfläche steuern können.
* Der "Phase Controller" ist nun dafür verantwortlich, nur die Abfolge der Phasen zu starten.
* Der "Phase Controller" ist nun dafür verantwortlich, nur den Ablauf zu starten und den Phasenwechsel zu steuern.
* Das Hauptformular "Crossing" ist dafür verantwortlich, zwei separate Controller-Haupt- und -Unterkontroller zu erstellen und diese den Ampeln zuzuweisen. und für jede Ampel eine Phasenwarteschlange zu erstellen.

Diese Methode erleichtert es, die Reihenfolge und die Zeit der Lampen wie gewünscht zu steuern. Das Programm kann eine Sequenz von einem Controller starten und den anderen für die gewünschte Zeit gedrückt halten.

* **Schritt 10**
* In diesem Schritt ist der Haupt-Controller dafür verantwortlich, die Sequenz von sich selbst und dem Sub-Controller zu starten.
* indem der Hauptcontroller als übergeordnet zum Subcontroller gemacht wird, ist er nun in der Lage, Befehle an den Subcontroller zu übergeben, wann er gestartet und bewertet werden soll.
* "Phase Controller" ist verantwortlich für das Starten des Timers und das Löschen der Phasen
* **Schritt 11**
* die Klassen "Lampe, Ampel, Verkehrsphase" sind nicht veränderbar, so dass sie nicht in jedem Schritt kopiert werden müssen.

Wenn Sie ein neues Projekt "Ampel Common" erstellen und diese Klassen hinzufügen, kann es in den anderen Projekten direkt ohne eine neue Kopie verwendet werden, indem Sie das Projekt "Ampel Common" referenzieren.

* Hinzufügen einer neuen Eigenschaft in der „TrafficLight“ zur Kontrolle, ob sie für Autos oder für Fußgänger ist.
* Neue Klasse erstellen Enthält alle Aufzählungen.
* Hinzufügen einer weiteren Warteschlange in "PhaseController", um die Original-Phasenwarteschlange zu kopieren, so dass die Arbeitswarteschlange verwendet werden kann und das Original so bleibt, wie es ist, so dass es bei Bedarf nicht kopiert werden kann, ohne die Warteschlange neu zu erstellen.
* **Schritt 13**
* Hinzufügen von "TrackBar", um das Programm vorwärts oder rückwärts bewegen zu lassen.
* durch Ändern des Wertes der "TrackBar" wurde eine Methode "GoToTime" genannt, diese Methode überprüft die Zeit des TrackBars und vergleicht sie mit der Phasendauer, indem sie die Zeit des TrackBars von der Summe aller Phasen subtrahiert, dann die Warteschlange löscht und den Rest der benötigten Phasen wieder der Warteschlange hinzufügt.