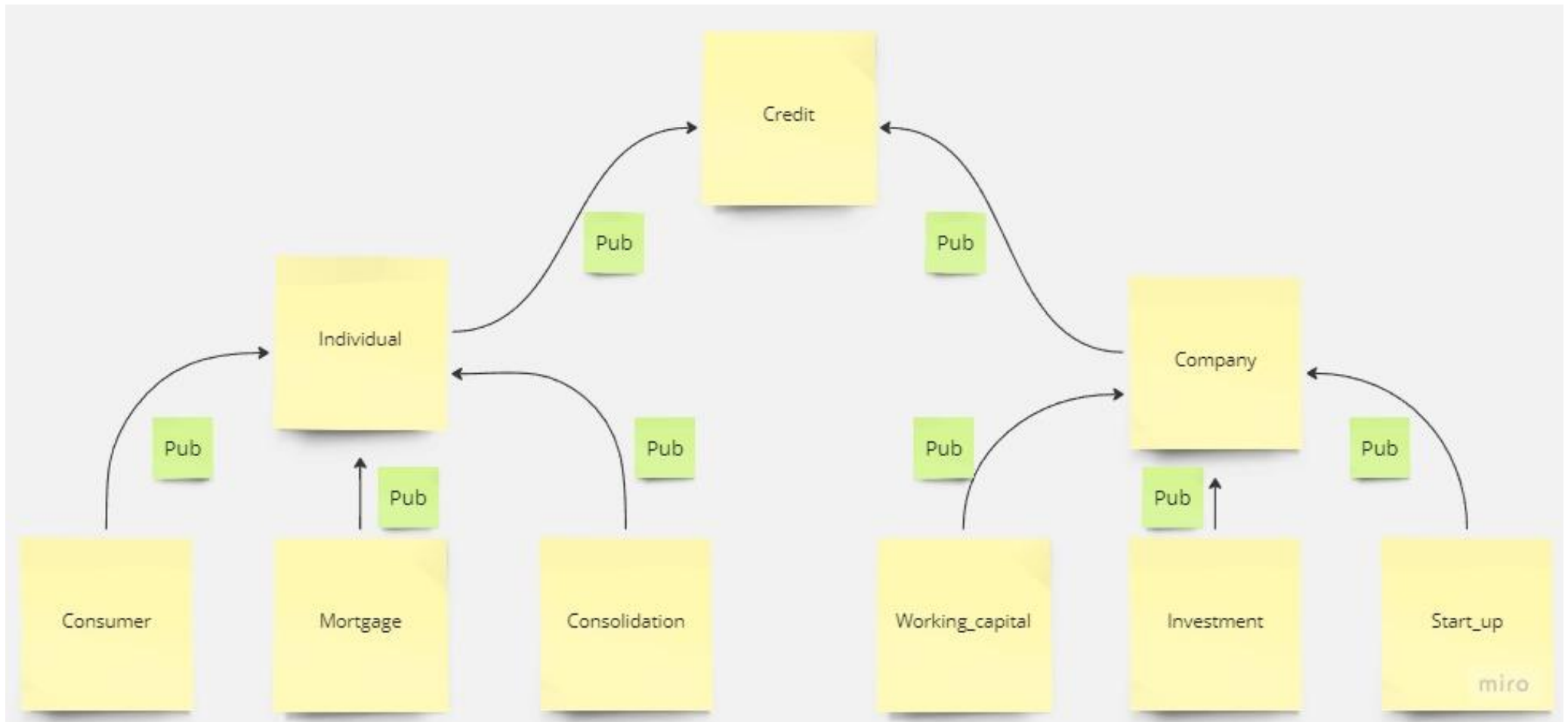


## Schemat hierarchii klas, struktury danych i najważniejsze metody

Strony są orientowane poziomo ze względu na zwiększenie czytelności hierarchii klas



Klasa „Credit” jest klasą abstrakcyjną z której dziedziczą klasa „Individual” (reprezentująca klientów indywidualnych) oraz klasa „Company” (będąca odpowiednikiem firm korzystających z oferty banku).

Dzięki temu podziałowi w klasie „Individual” mogę stworzyć ogólne metody, które będą adekwatne dla trzech klas pochodnych tej klasy.

Klasy te reprezentują trzy opcje kredytowe (Consumer, Mortgage, Consolidation)

Analogicznie do klasy „Individual” postąpię w klasie „Company” wyodrębniającej firmy starające się o kredyt.

Poniższe opisałem najważniejsze funkcje (są one wymienione w spodziewanej kolejności wywoływania):

*Wszystkie wirtualne metody zostaną umieszczone w klasie abstrakcyjnej „Credit”.*

*W trakcie przedstawienia danej metody zdarza się, że komunikuję jakie są następne kroki działania programu, jednak wszystkie funkcje będą wywoływane w funkcji „main”.*

Najważniejsze funkcje:

- **friend auto make\_decision()**: w tej funkcji klient podejmuje decyzje jaki kredyt chce dostać (dzięki tej funkcji program będzie wiedział na której klasie działać);

*Następnie klient zostanie poproszony o wypełnienie formularza z potrzebnymi danymi*

- **virtual void Credit::complete\_data() = 0**; metoda zadaje pytania i zapisuje odpowiedzi w strukturze „**needed\_data**”;

- **virtual int Credit::credit\_capacity() = 0**; metoda na podstawie udzielonych informacji oblicza zdolność kredytową klienta i zwraca przybliżoną wartość;

- **virtual bool check\_capacity()** ; jeżeli kwota o którą ubiega się klient nie jest większa od jego zdolności kredytowej to zwracana jest wartość „**true**”;

- **virtual void feedback()**: metoda udziela klientowi informacji zwrotnej o decyzji w sprawie kredytu i jeżeli jest ona negatywna to podaje aktualną zdolność kredytową osoby ubiegającej się o kredyt;
- **virtual int checkpoint()**: w przypadku (check\_capacity == false) metoda pyta klienta czy chce wypełnić swój formularz kolejny raz, jeśli klient wyrazi taką chęć to funkcja zwraca wartość „0”, dając tym możliwość ponownego wypełnienia formularza kredytowego. Metoda zwraca wartość „1”, gdy „check\_capacity==true”, wtedy zostaje wykonane przekierowanie do funkcji „final”. W przeciwnym wypadku funkcja zwraca „2” (skutkuje to przekierowaniem do funkcji exit);
- **virtual void final()**: metoda „final” informuje klienta o wysokościach rat oraz najważniejszych informacjach związanych z finalizacją kredytu, następnie prosi klienta o wpisanie „imienia i nazwiska”, które pieczętuje dokonanie kredytu, (wywołanie funkcji save\_info);
- **virtual void save\_info()**: metoda zapisuje odpowiedzi klienta do struktury accept\_credit,
- **virtual void exit()**: metoda pyta czy klient chce skorzystać po raz kolejny z pełnej oferty kredytów (opcjonalne przekierowanie do metody „make\_decision”, jeżeli klient wyrazi brak takiej chęci to zostanie zapytany czy chce on zakończyć działanie programu przez wpisanie komendy „!exit” (wpisanie tej komendy zamyka program)