Synopsis

You are to create a financial app using Java and JavaFX to build UI but you are not supposed to use any UI Builder. It must render well on all OS. The app shall have the capability of undertaking typical financial calculations such as savings, loans, and mortgages. The app shall have the ability to solve for **one** unknown from the given parameters noted in bold:

t – time in years (synonymous with number of payments)
r (%) – interest rate – for simple savings only[1]
P – present value
PMT – Payment
A – future value
PayPY – number of payments per year
CpY – number of compound payments per year
PmtAt – payment due at the beginning or end of each period (default is END)

Table 1 - Financial parameters (see examples below in the requirement notes)

[1] You only need to solve for the interest rate in problems where there are no monthly payments. For example lump sum investments.

[2] Look at the given link for more information of functional requirement https://www.calculator.net/finance-calculator.html

The app shall split typical financial problems up over typically four/five views:

1) Compound Interest savings (fixed sum investment with no further payments)

2) Savings – compound interest with regular contributions (this is savings where there might be sum invested with a subsequent further monthly contribution

3) Loans - compound interest with regular payments

4) Mortgage

In addition to this, the software **shall contain a help view** that will contain instructions and guidance to the user on how to use the software. You have complete freedom on how to implement this view and this can be done as a separate view or modal context views, for example, e.g. a pop- overview activated by a help button.

Requirements

R1 The software shall allow the user to estimate the **interest rate** based on other financial data given in Table 1 - Financial parameters.

R2 The software shall allow the user to estimate the **final value** based on other financial data given in Table 1 - Financial parameters.

R3 The software shall allow the user to estimate **present value** based on other financial data given in Table 1 - Financial parameters.

R4 The software shall allow the user to estimate **the payment** based on other financial data given in Table 1 - Financial parameters.

R5 The software shall allow the user to estimate the **number of payments** [2] based on other financial data given in Table 1 - Financial parameters.

R6 The software shall persistently save all user data[3]

R7 The software shall provide a help view

Notes:

Typical user scenarios and examples:

A user wishes to calculate the interest required to return a **future value** for a fixed initial investment sum over a known period of time. E.g. A user wishes to know if they invest £1000 for 5 years, what interest rate would be required to return a **future value** of £3000.

A user wishes to calculate the length of time in years (**or number of compound payments**[4]) required to return a future value for a known fixed initial investment given a known interest rate.
E.g. A user wished to know how long it would be before an investment of £1000 at an interest rate of 4% returned a **future value** of £3000.

[3] For the purpose of this application, all payments and compound interest are considered to be monthly. So, for example, 60 payments are equivalent to 5 years.

[4] If the app is backgrounded or closed by the user or runtime, all user data will be preserved persistently and the entry text fields repopulated with the last data used when the app is started or foregrounded.

[5] Compounds are monthly payments from the interest on savings or are the monthly interest charges on a loan.

A user wishes to calculate the initial investment (**present value**) that will return a known future value given a known interest rate and investment period (time).
E.g. A user wished to know how much to initially invest to get a **future value** of £3000 on savings account with an interest rate of 4% over a 5-year period.

E.g. A user wished to know how much to initially invest to get a **future value** of £5000 on savings account with an interest rate of 4% over a 5-year period where they make monthly payments into their savings account of £100.

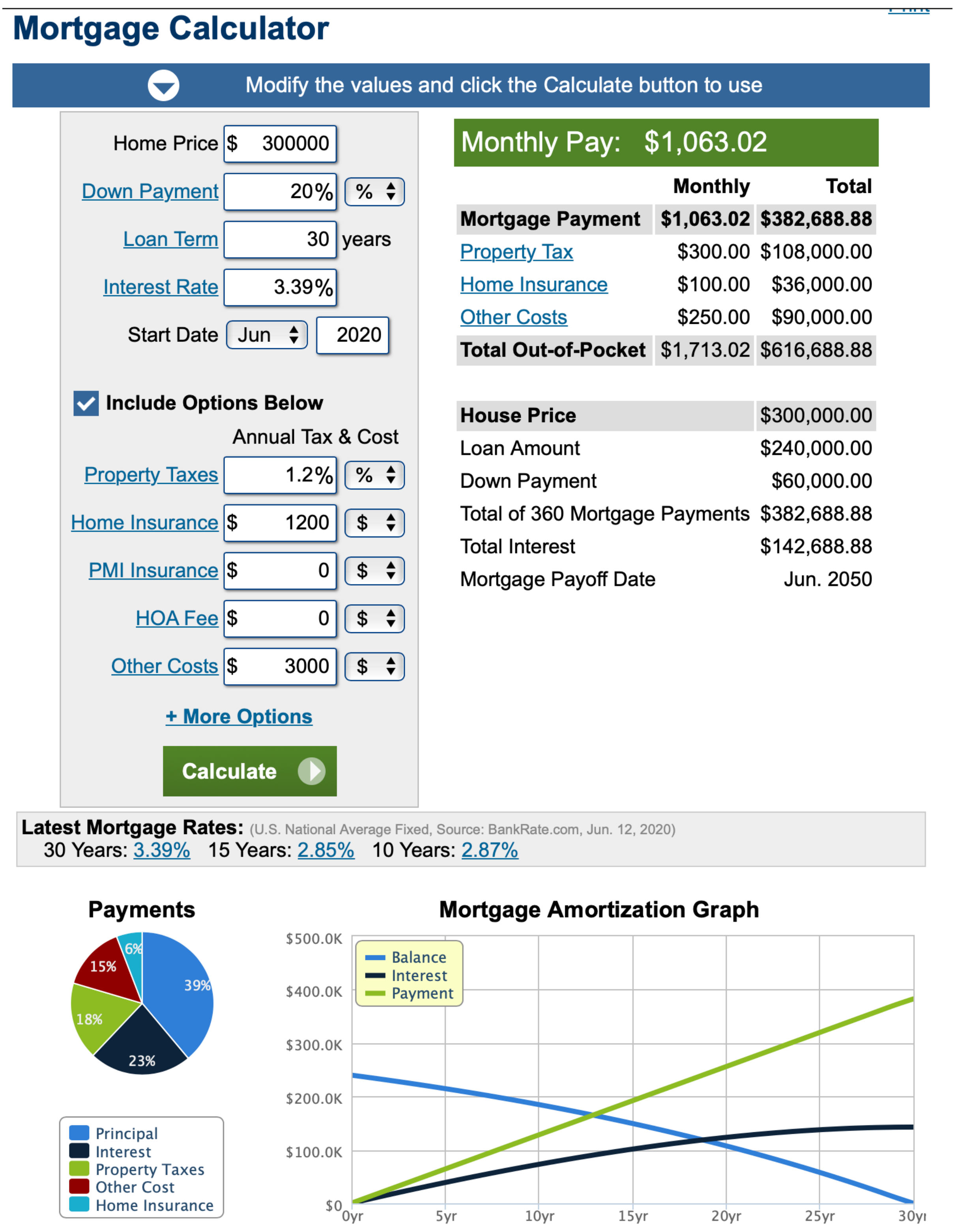
Note that loans are very similar to the above, for example, a user wished to borrow £3000, they can afford to pay £350 a month, how many monthly payments (remember this is the same as time) would they need to make if the best loan interest rate they could get was 7%?

View Requirements

Some possible designs for the app are shown below in figures. Note that you are free to design this app as you wish but it must meet all requirements and constraints. Note that the optimum design is to have views for: 1) Mortgage 2) Compound Savings (with and without regular contributions 3) Loans (note this will be similar to the mortgage but display the total number of payments as opposed to years)

[5] See the link for more information - https://www.calculator.net/finance-calculator.html

Mortgage



Mortgage Calculator

Modify the values and click the Calculate button to use

Home Price	$ 300000	
Down Payment	20%	%
Loan Term	30	years
Interest Rate	3.39%	
Start Date	Jun	2020

☑ Include Options Below

	Annual Tax & Cost	
Property Taxes	1.2%	%
Home Insurance	$ 1200	$
PMI Insurance	$ 0	$
HOA Fee	$ 0	$
Other Costs	$ 3000	$

+ More Options

Calculate

Monthly Pay: $1,063.02

	Monthly	Total
Mortgage Payment	**$1,063.02**	**$382,688.88**
Property Tax	$300.00	$108,000.00
Home Insurance	$100.00	$36,000.00
Other Costs	$250.00	$90,000.00
Total Out-of-Pocket	**$1,713.02**	**$616,688.88**

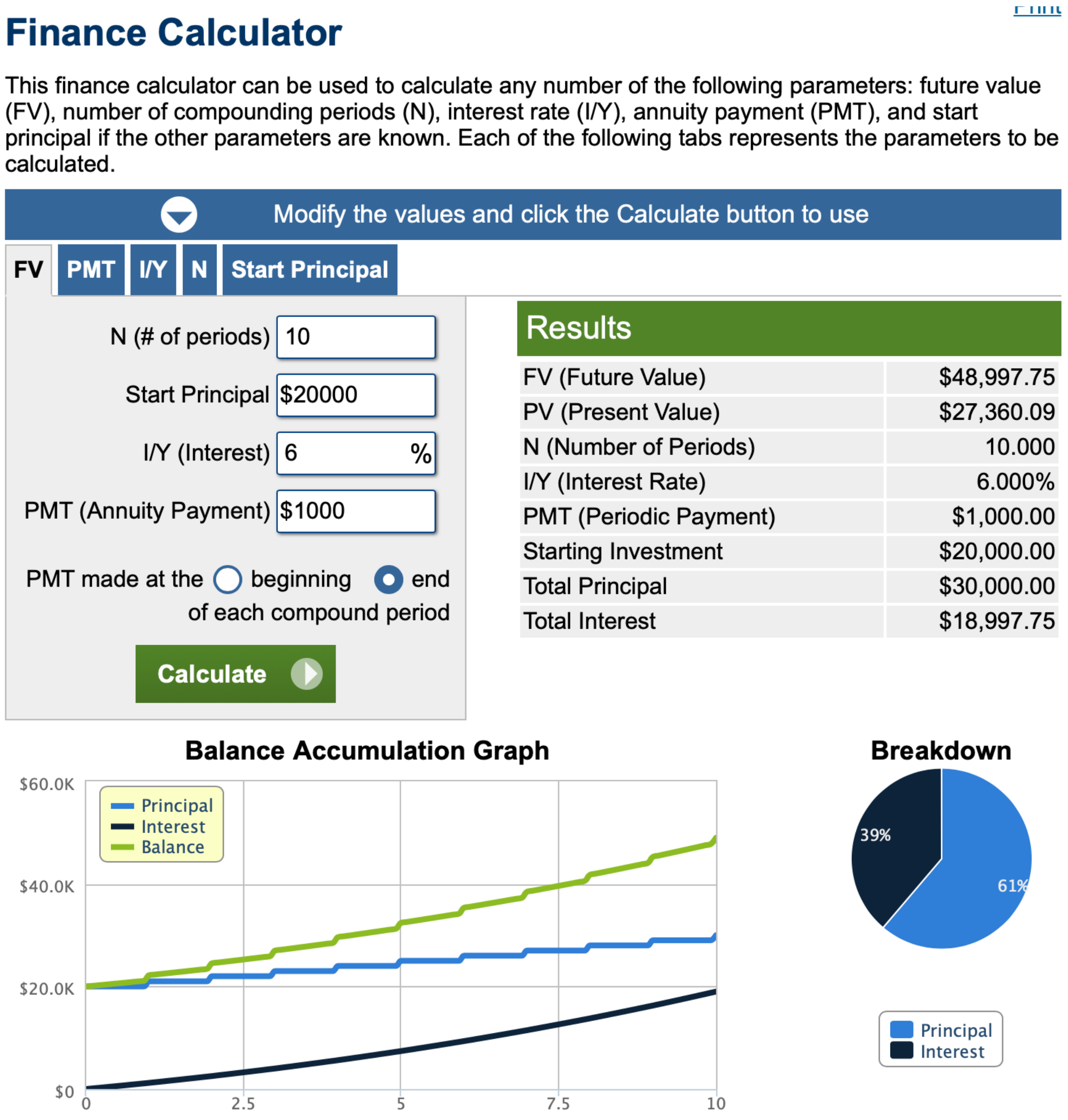
House Price	$300,000.00
Loan Amount	$240,000.00
Down Payment	$60,000.00
Total of 360 Mortgage Payments	$382,688.88
Total Interest	$142,688.88
Mortgage Payoff Date	Jun. 2050

Latest Mortgage Rates: (U.S. National Average Fixed, Source: BankRate.com, Jun. 12, 2020)
30 Years: 3.39% 15 Years: 2.85% 10 Years: 2.87%

Payments

Mortgage Amortization Graph

Savings



Finance Calculator

This finance calculator can be used to calculate any number of the following parameters: future value (FV), number of compounding periods (N), interest rate (I/Y), annuity payment (PMT), and start principal if the other parameters are known. Each of the following tabs represents the parameters to be calculated.

N (# of periods): 10

Start Principal: $20000

I/Y (Interest): 6 %

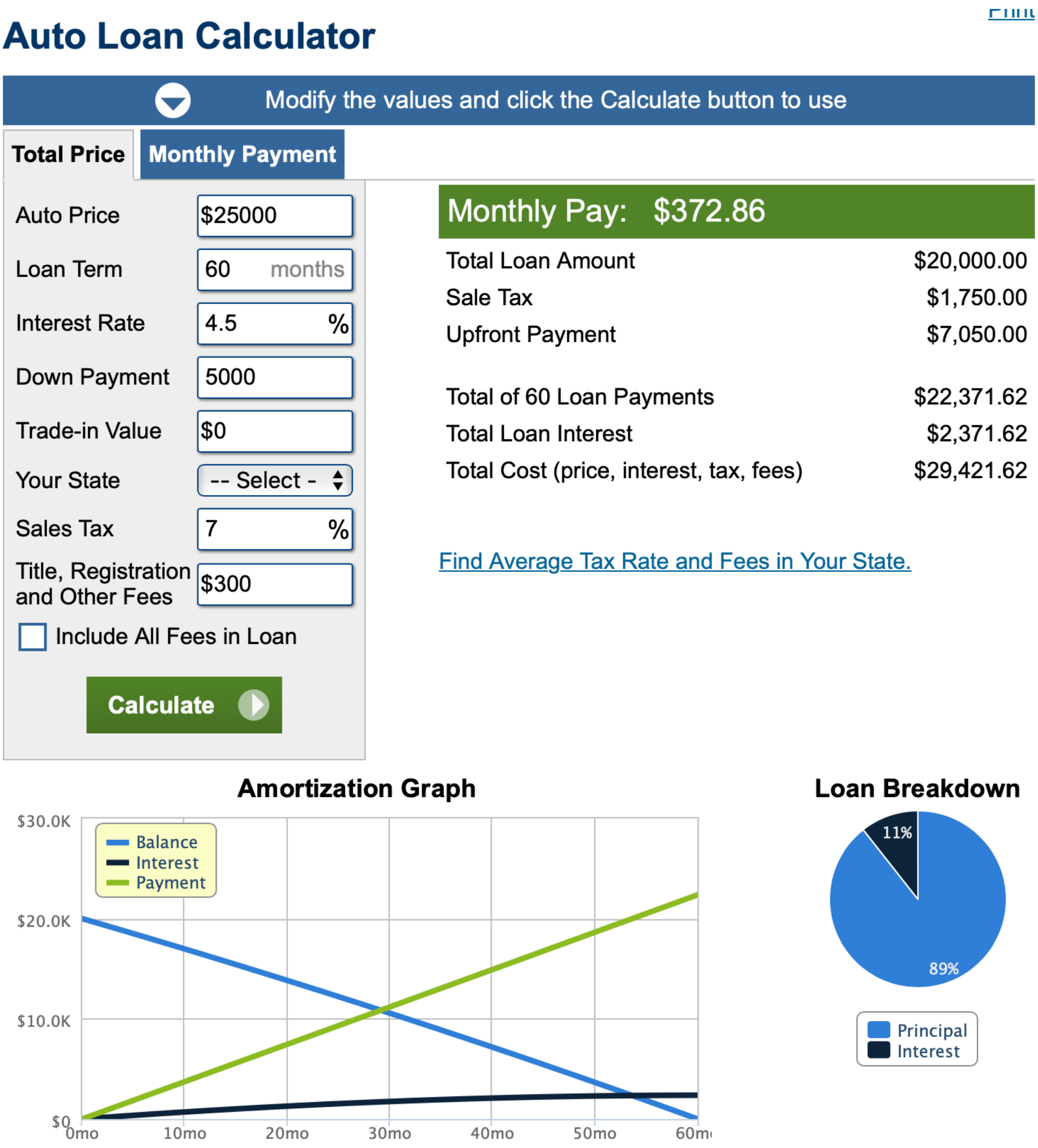
PMT (Annuity Payment): $1000

PMT made at the ○ beginning ● end of each compound period

Results

FV (Future Value)	$48,997.75
PV (Present Value)	$27,360.09
N (Number of Periods)	10.000
I/Y (Interest Rate)	6.000%
PMT (Periodic Payment)	$1,000.00
Starting Investment	$20,000.00
Total Principal	$30,000.00
Total Interest	$18,997.75

Loan



Print

Auto Loan Calculator

Modify the values and click the Calculate button to use

Total Price | Monthly Payment

Auto Price	$25000
Loan Term	60 months
Interest Rate	4.5 %
Down Payment	5000
Trade-in Value	$0
Your State	-- Select -
Sales Tax	7 %
Title, Registration and Other Fees	$300

□ Include All Fees in Loan

Calculate

Monthly Pay: $372.86

Total Loan Amount	$20,000.00
Sale Tax	$1,750.00
Upfront Payment	$7,050.00
Total of 60 Loan Payments	$22,371.62
Total Loan Interest	$2,371.62
Total Cost (price, interest, tax, fees)	$29,421.62

Find Average Tax Rate and Fees in Your State.

Amortization Graph

Loan Breakdown

Help View

The app shall contain a help view with simple instructions and optional images to help guide the user.

Operation

To use the application the user first selects one of the tabbed views from an array of icon buttons (either tabs or a toolbar) that can be accessed in all views. The field in which the user wishes to estimate one of the parameters should be left blank.

Keyboard Requirements

The app shall have a custom keyboard and shall have keys 0-9, delete last entry key, and decimal point[6]. A negative key is only needed if a combined savings/loan **view is used as negative numbers represent money out from the saver/borrower.**

[6] Note you can use the system decimal system keyboard also though a custom keyboard is better suited.

Non-functional requirements

1. The app shall render correctly in OS
2. The strings must be consistently formatted and correctly spelled.
3. Currency must be correctly formatted to two decimal places e.g. £345.21.
4. All financial calculations shall be mathematically correct to two decimal places.
5. Code shall be commented to assist marking and to assist understandability
6. All user data shall be persistently stored in file or database preferably NoSQL database

Appendix A - Financial Formula8

Note there are many formulae available for these calculations are you are free to use any you wish.

Compound Interest – Simple savings with lump sum (no payments)

$$A = P\left(1 + \frac{r}{n}\right)^{nt}$$

Where:

A = the future value of the investment/loan, including interest
P = the principal investment amount (the initial deposit or loan amount – present value)
r = the annual interest rate (e.g. 3.2% is 0.032)
n = the number of times that interest is compounded per unit time (this is always monthly for the purpose of this coursework, i.e. **12** per year)
t = the time the money is invested or borrowed in **years**

Interest Rate

$$r = n\left[\left(\frac{A}{P}\right)^{\frac{1}{nt}} - 1\right]$$

Principle Amount

$$P = \frac{A}{\left(1 + \frac{r}{n}\right)^{nt}}$$

Formula for time

$$t = \frac{ln\left(\frac{A}{P}\right)}{n\left[ln\left(1 + \frac{r}{n}\right)\right]}$$

Note: that ln is the natural logarithm (this is **log()** in Swift)

Compound interest formula (with regular payment or contributions)

These equations assume frequency of contribution and compounding is the same (for example every month)

Compound interest formula (with regular contributions) for deposits made at the end of the period - (important note: any money given so going away from lender/saver e.g. a principle amount or payments is a negative number)

Note that the Total A = **[Compound interest for principal] + [Future value of a series]**

Compound interest for a principal amount

$$\boldsymbol{P\left(1+\frac{r}{n}\right)^{nt}}$$

Future value

$$A = PMT \times \left\{\frac{\left[\left(1+\frac{r}{n}\right)^{nt} - 1\right]}{\frac{r}{n}}\right\}$$

Payment

$$PMT = \frac{A}{\left\{\frac{\left[\left(1+\frac{r}{n}\right)^{nt} - 1\right]}{\frac{r}{n}}\right\}}$$

Time to achieve a certain future value A

The following formula will calculate the **time taken in years** to **reach an investment goal A loan.**

$$t = t = \ln\frac{(1+\frac{rA}{PMT})}{\ln(1+r)\times 12}$$

The following formula will calculate the **time taken in years** to **completely pay a loan** – can be useful to express this also in number of monthly payments to a user also (i.e. t x 12). P is the loan amount.

$$t = \ln\frac{(1-\frac{rP}{PMT})}{\ln(1+r)\times 12}$$

Mortgage Payments (or any loan) – future value is assumed to be equal to zero for mortgage and loan calculations

$$PMT = \frac{A\frac{r}{12}\left(1+\frac{r}{12}\right)^{12t}}{(1+\frac{r}{12})^{12t}-1}$$

Mortgage Length of time (t) in years (same as loan)

$$t = \ln\frac{(1-\frac{rA}{PMT})}{\ln(1+r)\times 12}$$