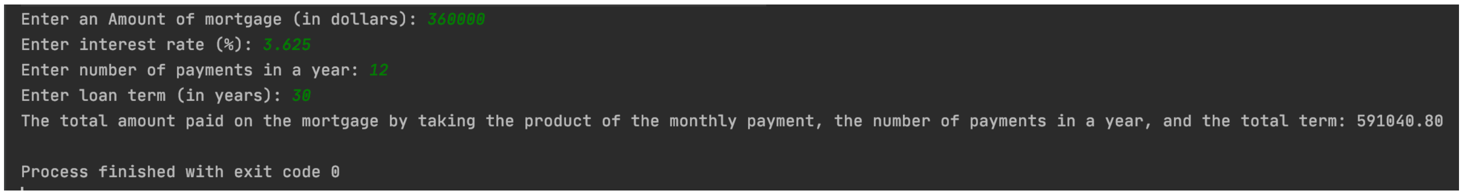
Chapter 6, #12 – Mortgage caclulator.

```
package Homework2.mortgage;
import java.util.Calendar;
public abstract class FinancialCalculator {
    protected Calendar calendar; // calendar
    protected int getDaysInMonth() {
        return calendar.getActualMaximum(Calendar.DAY_OF_MONTH);
    } // returns the number of days for the current month set on calendar
    protected int getDaysInYear() {
        return calendar.getActualMaximum(Calendar.DAY_OF_YEAR);
    } // returns the number of days in the current year set on calendar
    protected abstract void getUserInput();
    protected abstract void compute();
}

package Homework2.mortgage;
import java.math.BigDecimal;
import java.math.MathContext;
import java.math.RoundingMode;
import java.util.Scanner;
public class MortgageCalculator extends FinancialCalculator {
    protected BigDecimal A; //mortgage Amount
    protected BigDecimal r; //interest Rate
    protected int n; //number Of Payments in a Year;
    protected int T; //term Of Mortgage in Year;
    @Override
    protected void getUserInput() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter an Amount of mortgage (in dollars): ");
        A = new BigDecimal(scanner.next());
        System.out.print("Enter interest rate (%): ");
        r = new BigDecimal(scanner.next());
        System.out.print("Enter number of payments in a year: ");
        n = scanner.nextInt();
        System.out.print("Enter loan term (in years): ");
        T = scanner.nextInt();
    }
    public BigDecimal computeMonthlyPayment() {
        MathContext mc = MathContext.DECIMAL128;
        BigDecimal partialNumerator = r.divide(BigDecimal.valueOf(n * 100), mc); // Numerator partially
r.divide(n)
        BigDecimal numerator = A.multiply(partialNumerator); // Numerator whole A*r/n
        int exponentiation = n * T; // Calculation exponentiation only
        BigDecimal denominator =
BigDecimal.valueOf(1).subtract(BigDecimal.valueOf(1).divide(BigDecimal.valueOf(1).add(partialNumerator).pow(expon
entiation), mc)); // BigDecimal denominator 1-[1/(1+r/n)^nT]
        BigDecimal monthlyPayment = numerator.divide(denominator, mc).setScale(2, RoundingMode.HALF_UP); //
Monthly payment = A*r/n/1-[1/(1+r/n)^nT]
        return monthlyPayment;
    }
    @Override
    protected void compute() {
        BigDecimal monthlyPayment = computeMonthlyPayment(); // Monthly payments
        int numOfPayments = n * T; // Calculation number of payments
        BigDecimal totalAmount = monthlyPayment.multiply(BigDecimal.valueOf(numOfPayments)); // Total amount
        System.out.println("The total amount paid on the mortgage by taking the product of the monthly payment,
the number of payments in a year, and the total term: " + totalAmount);
    }
}

package Homework2.mortgage;
public class Calculate {
    public static void main(String[] args) {
        MortgageCalculator mortgageCalculator = new MortgageCalculator();
        mortgageCalculator.getUserInput();
        mortgageCalculator.compute();
    }
}
```



```
Enter an Amount of mortgage (in dollars): 360000
Enter interest rate (%): 3.625
Enter number of payments in a year: 12
Enter loan term (in years): 30
The total amount paid on the mortgage by taking the product of the monthly payment, the number of payments in a year, and the total term: 591040.80

Process finished with exit code 0
```