BaseUi
+ show(): void
# showHeader(): void
showContent(): void
showFooter(): void
+ buildBaseController(): BaseController
+ getTitle(): String

BaseController
+ getBalance(): BigDecimal
+ buildCheckingAccount(): CheckingAccount
+ getWeeklyExpense(): BigDecimal

IncomeRegisterUI
controller: RegisterIncomeController
listIncomeType: List<IncomeType>
+ buildBaseController(): BaseController
getTitle(): String
showContent(): void
readAmount(): BigDecimal
readDate(): Date
readDescription(): String
selectIncomeType():

RegisterIncomeController
incomeTypeRepository: IncomeType
buildCheckingAccount(): CheckingAccount
getIncomeTypes(): IncomeTypes

Income
- incomeType: IncomeType
- dateOccurred: Date
+ getIncomeType(): IncomeType
+ setDateOccured(Date): void
+ setIncomeType(IncomeType): void
+ getDateOccurred(): Date

IncomeTypeRepository
- list: List<IncomeType>
+ getIncomeTypes(): list

CheckingAccount
income: income
amount: BigDecimal
+ getBalance(): BigDecimal
+ getExpensesTotal(): BigDecimal
setIncomesTotal(): BigDecimal
+ add(Income): void
+ add(Expense): void

IncomeRepository
listIncome: List<Income>
+ save(income): void
+ getListIncome(): List<Income>

Note: "\: not study, i.e., not our scoop, i.e., dependencies.

imply / invocates

Controller: Register Income Type Repository
Model: IncomeType   Transaction (inherits by Income)
Persistence:
Presentation: Main menu   Register Income Type UI

MR1: assoc.
① implied by return types
  BC — CA
② Instances creation between A and B
  RIC → I

MR:2 An aggr. from the code is modeled as an assoc. in the model
  IR → I
Dis: CA → IR