| Circle | CirclePrivate |
| --- | --- |
| radius:float | -radius:float |
| Circle(radius=1)  getPerimeter() : float  getArea(): float  setRadius(): None  getRadius(): float  \_\_str\_\_: String(Radius, Perimeter, Area) | Circle(radius=1)  getPerimeter() : float  getArea(): float  setRadius(): none  getRadius(): float  \_\_str\_\_: String(Radius, Perimeter, Area) |

| Rectangle | RectanglePrivate |
| --- | --- |
| width:float  height:float | Student complete |
| Rectangle(width=1,height=2)  getWidth():float  getHeight():float  getArea() : float  getPerimeter(): float  \_\_str\_\_:String(Width, Height,Area, Perimeter) |  |

| Account |
| --- |
| -idd: int = (int)(time.time())  -name: string  -checking: float  -savings:float |
| Account(checking=0.0,savings=0.0)  getID() : int  getName(): string  getChecking(): float  getSavings(): float  checkingDeposit(): none  checkingWithdrawal(): none  savingsDeposit(): none  savingsWithdrawal(): none  \_\_str\_\_(): string (idd, name, checking, savings, total) |

| import datetime  now = datetime.datetime.now() |
| --- |
| Student |
| -name: string  -gradYear: int  -town: string  -classes: list()  -scores: list() |
| Student(name=”unknown”, gradYear= now.year, town=”unknown”, classes=list(), scores=list())  getName() : String  setName(): none  getGradYear(): int  setTown(): none  getTown(): string  printClasses(): none  addClass(): none  \_\_str\_\_(): String (name, gradYear, town, printClasses()) |

| import time |
| --- |
| StopWatch |
| -startTime: float  -endTime: float |
| StopWatch(startTime=0,endTime=0)  getStartTime() : float  getEndTime(): float  start(): none  stop(): none  returnTime():String(HH:MM:SS:MMMM)  getElapsedTime(): float |

| import turtle |
| --- |
| Line |
| -xStart: float  -yStart: float  -xEnd: float  -yEnd: float  -aline: turtle.Turtle() |
| Line(xStart, yStart, xEnd, yEnd)  drawLine() : none  getDeltaY(): float  getDeltaX(): float  getM(): float  getB(): float  \_\_str\_\_(): String (xStart, yStart, xEnd, yEnd) |







| import turtle |
| --- |
| LinearEquation |
| -line1: Line  -line2: Line  -vertex: turtle.Turtle() |
| LinearEquation(line1,line2)  isSolvable() : print  drawVertex(): float  getX(): float or print  getY(): float or print  getGraph(): none |